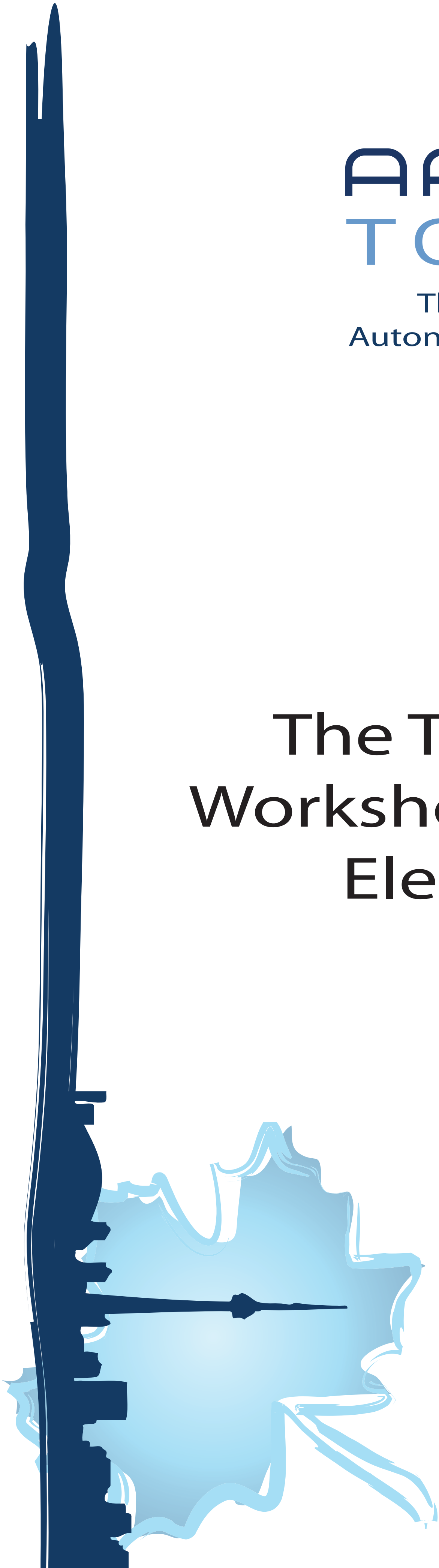


AAMAS 2010
TORONTO

The 9th International Conference on
Autonomous Agents and Multiagent Systems
May 10-14, 2010
Toronto, Canada

Workshop 7

The Twelfth International Workshop on Agent-Mediated Electronic Commerce

AMEC 2010

Editors:
Wiebe van der Hoek
Gal A. Kaminka
Yves Lespérance
Michael Luck
Sandip Sen

Twelfth International Workshop on

Agent-Mediated Electronic Commerce

AMEC 2010

Onn Shehory
IBM Haifa Research Lab
Haifa University, Israel
onn@il.ibm.com

Esther David
Department of Computer Science
Ashkelon Academic College, Israel
astrdod@acad.ash-college.ac.il

Sebastian Stein
School of Electronics and Computer Science
University of Southampton, UK
ss2@ecs.soton.ac.uk

Alex Rogers
School of Electronics and Computer Science
University of Southampton, UK
acr@ecs.soton.ac.uk

Programme Committee

John Collins, University of Minnesota, USA

Shaheen Fatima, Loughborough University, UK

Enrico Gerding, University of Southampton, UK

Minghua He, Aston University, UK

Sverker Janson, SICS, Sweden

Sven Koenig, University of Southern California, USA

Kate Larson, University of Waterloo, Canada

Tracy Mullen, Penn State University, USA

David Pardoe, University of Texas, USA

Simon Parsons, City University of New York, USA

Juan Antonio Rodriguez Aguilar, IIIA-CSIC, Spain

Jeffrey Rosenschein, The Hebrew University of Jerusalem, Israel

Alberto Sardinha, Carnegie Mellon University, USA

Perukrishnen Vytelingum, University of Southampton, USA

William Walsh, CombineNet, Inc., USA

Michael Wellman, University of Michigan, USA

Dongmo Zhang, University of Western Sydney, Australia

Table of Contents

FULL PRESENTATIONS

SHORT PRESENTATIONS

FULL PRESENTATIONS

Decentralised Supply Chain Formation: A Belief Propagation-Based Approach

Michael Winsper, Maria Chli

Computer Science, Aston University, Aston Triangle,
Birmingham B4 7ET, United Kingdom
{winsperm, m.chli}@aston.ac.uk

Abstract. Decentralised supply chain formation involves determining the set of producers within a network able to supply goods to one or more consumers at the lowest cost. This problem has been tackled in a number of ways, including auctions, negotiations, and argumentation-based approaches. In this paper we show how this problem can be cast as an optimisation of a Markov Random Field energy function. Optimising this class of energy functions is NP-hard but efficient approximations to the global minimum can be obtained using Loopy Belief Propagation (LBP). Here we detail a LBP-based approach to the supply chain formation problem, involving decentralised message-passing between participants. Our approach is evaluated against a well-known double-auction method and an optimal centralised technique, showing several improvements: it obtains better solutions for most networks that admit a competitive equilibrium[1] while also solving problems where no competitive equilibrium exists, for which the double-auction method frequently produces inefficient solutions.

Keywords: Supply chain formation, task allocation, belief propagation

1 Introduction

As the drive for efficiency and adaptability becomes an increasing focus in industry, together with rising levels of uncertainty about market conditions, the ability to quickly form effective, mutually beneficial trading partnerships becomes increasingly important. Although the concept of virtual enterprises – ad-hoc coalitions of businesses formed to pool resources and create synergies in order to respond to emergent business opportunities – may have yet to reach the level of popularity its proponents had hoped for, the principles of such arrangements, along with outsourcing, remain integral to the business processes of many organizations [1].

Traditional non-computational approaches to supply chain formation are inefficient processes, with time wasted on contract tendering and negotiations. Time constraints and human irrationality may lead to the establishment of inefficient supply chains, a problem that could be mitigated or avoided with the use of computational techniques.

[1] Bounded competitive equilibrium is an SAMP-SB-specific notion defined in [2]. The authors use this definition to allow for minor inefficiencies in their auction protocol and agent bidding strategies.

Agent-based computational approaches to supply chain formation model potential supply chain participants – businesses capable of forming a link in the yet-to-be-completed chain – as rational self-interested computational agents. These agents deliberate between themselves, typically either through centralised negotiations or decentralised auctions, about the subset of agents capable of forming the most efficient supply chain. At the conclusion of these deliberations, which, although depending on the number of potential participants, are typically much faster than real life negotiations, the supply chain is formed instantly.

While both negotiation and auction-based approaches to supply chain formation have seen significant research, the reliance on an assumption of centralisation (for many negotiation-based approaches), or the problem of inefficient agent bidding strategies leading to suboptimal allocations (for auction-based approaches) has meant that neither method has been truly established as the most efficient technique for the computational formation of supply chains.

In this paper, we propose a belief propagation-based approach to decentralised supply chain formation which is capable of producing efficient results over a range of network topologies. With our use of belief propagation, we are able to produce results comparable to that of a centralised approach whilst still working in a decentralised manner. The use of message passing also allows us to take full advantage of the graphical structure of our networks.

In section 2, we provide details of previous models of decentralized supply chain formation, and explain why belief propagation is a useful approach for supply chain formation when networks are represented graphically. In section 3, we provide details of our model, inspired by work previously conducted by Walsh and Wellman [2], and provide the details of the loopy belief propagation algorithm. Section 4 describes our experiments, while section 5 shows our results and compares them to the results obtained by [2] and the optimal centralised method. Section 6 provides some conclusions about our work, while section 7 identifies areas of related future research.

2 Background

Multi-agent systems enable us to model a number of properties characteristic of supply chains: uncertainty, decentralized decision making by self-interested agents and the process of self-organisation being just a few examples. It is no surprise, then, that application of the multi-agent paradigm to the area of supply chain formation and related areas has been an ongoing focus of multi-agent systems research for several years. Existing decentralised supply chain formation and related literature can be thought of as encompassing two broad areas: research which focuses on methods for decentralised negotiation as a means for determining allocations, and studies which model the supply chain as networks of auctions.

Davis and Smith's Contract Net protocol [3], which details a method for distributed problem solving based on task decomposition and negotiation, formed the basis for agent-based models of decentralised negotiation, an approach which lends itself well to the modelling of supply chains: each individual procurement and sale decision by each participant in the supply chain can be looked at as a multi-party negotiation, with factors such as negotiation protocols and bidding rules providing interesting avenues of investigation. Papers using the negotiation approach include Wang et al. [4], which uses argumentation-based negotiation for decision making in supply chain formation, and Jiang et al. [5], which applies an extension of the contract net to a petroleum supply chain.

The other main approach to supply chain formation involves modelling the supply chain as a network of auctions, with first and second-price sealed bid auctions, double auctions and combinatorial auctions among the most frequently-used methods.

Supply chain formation through auctions is a popular approach for a number of reasons: auctions are frequently used for real-world tendering and sales opportunities, many auctions possess a number of interesting game-theoretic properties, and auctions are often able to form satisficing solutions. Perhaps the most comprehensive series of studies on supply chain formation using auctions comes from Walsh, Wellman et al, who examine the efficiency of supply chains formed using simultaneous auctions [2] and combinatorial auctions [6]. In their simultaneous auctions paper, Walsh and Wellman define a market protocol with bidding restrictions, referred to as SAMP-SB – simultaneous ascending M+1st Price Auctions with Simple Bidding – capable of producing near-optimal allocations, over several network structures, although it frequently struggled on networks where competitive equilibria did not exist. They proposed a similar protocol with the provision for decommitment in order to remedy the inefficiencies caused by "dead ends" created by producers acquiring input goods and not finding a buyer for their output, but this was recognized as an imperfect solution, due to the possible problems created by rendering the results of auctions as non-binding. More recent work has seen the proposal of mixed multi-unit combinatorial auctions (MMUCAs) for supply chain formation [7], with the standard combinatorial model of bids being placed for bundles of goods replaced by negotiations over "transformations", essentially commitments by bidders to produce a set of output goods given a set of input goods.

We chose to use [2] as a basis for our model due to its being extensively and lucidly defined, with clearly specified graphical network structures and results providing a sound foundation for performance evaluation and comparison. As stated in [2], supply chain networks lend themselves well to a graphical representation, in the form of task dependency networks, due to the property of hierarchical subtask decomposition, a defining feature of supply chain formation. Where [2] use double auctions to determine allocations, however, we use loopy belief propagation. The graphical nature of the supply chain networks used by [2] provides for the use of graphical inference algorithms in determining the optimal participants within a supply chain network, allowing us to quickly find efficient allocations in a decentralized manner. Loopy belief propagation, an extension of Pearl's polytree algorithm [8] for inference in graphs containing cycles, is well-suited to this task, due to its ability to converge to correct solutions on acyclic and single-cycle graphs [9] whilst still providing good approximations for more complex graphs [9].

Although loopy belief propagation has not heretofore been applied to supply chain formation, it has seen success in related areas such as sensor networks [10], [11]. Loopy belief propagation works using decentralised message passing, allowing our agents to share beliefs about the optimal structure of the network without revealing any more private cost information than they would in an open auction, whilst being able to produce results comparable to those of centralised approaches.

3 Model

We model our supply chain networks as bipartite directed acyclic graphs. There are two types of node: individual producers and consumers, which are represented by rectangles in our network diagrams, and goods represented with circles. Directed edges indicate potential flows of goods. Edges signify the ability for supply chain participants to produce or consume goods. An edge leading from a producer to a good indicates that the producer is capable of producing the good, while an edge leading from a good to a producer or consumer means that that producer or consumer is able to consume the good. Consumers, as their name suggests, cannot produce goods.

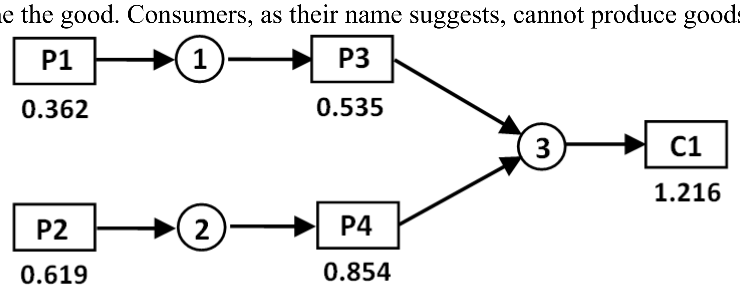


Fig. 1. A sample supply chain network, from [2]. Producers (P1,P2,P3,P4) and consumers (C1) are represented by rectangles, with goods represented by circles. Edges between vertices show possible flows of goods. Numbers below producers indicate production costs, while numbers below consumers indicate consumption values.

This approach, identical to that used by Walsh and Wellman [2], allows us to clearly state network structures while retaining fidelity to the structure of real-world supply chains. For example, in Figure 1, we see that producer P1 is able to produce good 1 at a cost of 0.36, which producer P3 needs to consume in order to produce good 3, at a cost of 0.53 plus the cost of acquiring good 1, for consumer C1. Similarly, producer P2 is able to produce good 2, for possible consumption by producer P4, which is also able to supply consumer C1 with good 3. If both producers P3 and P4 are able to acquire their single *input good*, C1 must make a choice about which producer to purchase from. Ideally it will choose the producer able to supply the good at the lowest accumulated cost, in this example P3, leaving C1 with a final positive consumption value of $(1.216 - (0.362 + 0.535) = 0.319)$.

Goods represent a single unit of a commodity which is non-divisible, and equivalent in all aspects other than price; for reasons of simplicity and clarity, we do not attempt to model aspects such quality, quantity or delivery constraints.

3.1 Agents

Our supply chain networks are made up of multiple interlinked producers aiming to supply a good or goods to one or more consumers.

Producers Producers are capable of producing a single unit of a single type of output good, and to do so are required to have obtained a single unit of each of the goods in their set of input goods, which may be zero, one, or many. Producers which do not require any inputs to produce their output good are known as no-input producers, and form the initial echelon of the supply chain. In the case of a producer requiring multiple inputs, we refer to the goods as complementary – a producer is unable to produce its output good if it is only able to acquire a subset of its required input goods. In producing their output good, producers incur a *production cost*. This is a constant that depends on the particular producer. It represents the expense incurred by producing a certain product, e.g. the cost of running a factory.

Consumers Consumers require a single unit of a single good from their set of consumable goods. In each network, each consumer is assigned a static *consumption value* V_c: this is the personal valuation the consumer holds for obtaining one of its consumable goods.

3.2 States

Due to the fixed structure of the networks, for each agent there exist a finite number of purchases and sales (if the agent is a producer) in which the agent is viable. We encode each of these tuples of exchange relationships as *states*, with each state defining a list of suppliers and a buyer if the agent is a producer, and a single supplier for consumers. For example, a possible state for producer P3 in Figure 1 is "Buy from P1 and sell to C1". The number of states an agent possesses increases with the number of producers able to supply its input good(s), and the number of producers or consumers able to consume its output good. As well as a list of active states, we also allow for the *inactive state*, where the agent does not acquire or produce any goods.

3.3 Cost Function

We allow for two distinct types of cost, denoted as $f_v(x_v)$, the *unary cost* for agent v of being in state x_v, and $g_{uv}(x_u, x_v)$, the *pairwise cost* of connected agents u and v being in states x_u and x_v. Our method minimises the function given below:

$$\varepsilon(x_1, \dots, x_N) = \sum_{v \epsilon V} f_v(x_v) + \sum_{(u,v) \epsilon E} g_{uv}(x_u, x_v) \ . \quad (1)$$

Where $\varepsilon(x_1, \dots, x_N)$ is the set of agents, $f_v(x_v)$ is the unary cost of agent v being in state x_v, and $g_{u,v}(x_u, x_v)$ is the pairwise cost of linked agents u and v, being labeled with states x_u and v_v. With all else equal, the lower the cost function, the more efficient the allocation. We use the efficiency of an allocation as a measure of the quality of the solution found.

Unary Cost Each agent associates each of its states with a cost. For all agents, the cost of being in the inactive state is zero. For producers, all active states incur a positive cost, equal to their production cost. Consumers assign a negative cost $0 - V_c$ to all states in which they acquire a good, where V_c represents the consumer's *consumption value*, the value they assign to the acquisition of their consumable good.

Pairwise Cost In order to calculate the pairwise cost for two states, we must first assess their compatibility. Two states are compatible if both are inactive states, *or* the lists of sellers and buyers align such that neither is trying to buy from or sell to the same agent, *or* the states describe a situation where agent u wants to sell to agent v, v's list of sellers includes u, and neither is inactive, and vice versa. If the states are compatible, we set the pairwise cost to zero; if not, the pairwise cost is infinite.

Simple Network Costs To provide an example of our system of costs in practice, we now show the vectors of unary costs and matrices of pairwise costs in our simple network, as shown in Figure 1. The simple network is made up of a set of four producers and a single consumer, as well as three potential goods for production. The possible states of our agents are:

P1: $\{t1, t2\}$. $t1$ = "Inactive". $t2$ = "Sell to P3".
P2: $\{u1, u2\}$. $u1$ = "Inactive". $u2$ = "Sell to P4".
P3: $\{v1, v2\}$. $v1$ = "Inactive". $v2$ = "Buy from P1 and sell to C1".
P4: $\{w1, w2\}$. $w1$ = "Inactive". $w2$ = "Buy from P2 and sell to C1".
C1: $\{x1, x2, x3\}$. $x1$ = "Inactive". $x2$ = "Buy from P3". $x3$ = "Buy from P4".

Producer P1 does not require any inputs, and is only capable of selling to one agent - producer P3 - meaning its sole active state is $t2$, representing the state of not buying any inputs, and selling to P3. Consumer C1 has two valid active states: buying from P3 and selling to no-one - $x2$ - and buying from P4 and selling to no-one, $x3$.

With our list of states complete, we now show the unary costs of the states. Inactive states incur a unary cost of 0, while active states depend upon the type of agent in question. For producers, the unary cost is equal to the production cost of the producer in question. Consumers incur a unary cost of $0 - V_C$, where V_C is the consumption value of the consumer in question. Thus, our unary costs are as follows:

P1: $f_{P1}(t1) = 0$. $f_{P1}(t2) = 0.362$.
P2: $f_{P2}(u1) = 0$. $f_{P2}(u2) = 0.619$.
P3: $f_{P3}(v1) = 0$. $f_{P3}(v2) = 0.535$.
P4: $f_{P4}(w1) = 0$. $f_{P4}(w2) = 0.854$.
C1: $f_{C1}(x1) = 0$. $f_{C1}(x2) = -1.216$. $f_{C1}(x3) = -1.216$.

Finally, we show the pairwise costs associated with P3 in our simple network:

$g_{P1P3}(t1, v1) = 0$. $g_{P1P3}(t1, v2) = \infty$, $g_{P1P3}(t2, v1) = \infty$. $g_{P1P3}(t2, v2) = 0$.
$g_{P3P1}(v1, t1) = 0$. $g_{P3P1}(v1, t2) = \infty$, $g_{P3P1}(v2, t1) = \infty$. $g_{P3P1}(v2, t2) = 0$,
$g_{P3C1}(v1, x1) = 0$. $g_{P3C1}(v1, x2) = \infty$, $g_{P3C1}(v1, x3) = 0$. $g_{P3C1}(v2, x1) = \infty$,
$g_{P3C1}(v2, x2) = 0$. $g_{P3C1}(v2, x3) = \infty$.
$g_{C1P3}(x1, v1) = 0$. $g_{C1P3}(x1, v2) = \infty$, $g_{C1P3}(x2, v1) = \infty$. $g_{C1P3}(x2, v2) = 0$,
$g_{C1P3}(x3, v1) = 0$. $g_{C1P3}(x3, v2) = \infty$.

The next section introduces the details of loopy belief propagation, the technique we employ to minimise our cost function. There are, of course, other possible approaches: a brute force approach, for example, would be able to deliver consistent optimal results given enough time,; various DCOP and DPOP search algorithms would also be capable of producing optimal solutions. We propose LBP, as an as-yet unused approach to this problem, for its ability to produce efficient results whilst operating in a decentralised manner.

3.4 Loopy Belief Propagation

The process of belief propagation [12] begins by first initialising the beliefs of the set of agents about each of their states to zero. The agents then pass a message – containing a vector of belief values – to each of their neighbours in the network. Once all agents have passed a message to each of their neighbours, each agent updates its beliefs based upon the content of the messages it received. This process of message passing and belief aggregation continues until the beliefs of our agents become stable, at which point we determine the final state of each agent and perform the allocation.

Belief Aggregation For each of agent u's possible states, we use equation 2 to calculate u's belief in that state. At initialisation, each agent holds a belief of zero about each of its states.

$$bel_u(x_u) = f_u(x_u) + \sum_{w \in N_u} m_{w \to u}(x_u) \, . \qquad \textbf{(2)}$$

$bel_u(x_u)$ corresponds to agent u's belief in its state x_u. This belief is made up of two parts: first is the unary cost $f_u(x_u)$ to u incurred by being in state x_u. This is added to the sum of the beliefs about state x_u contained within the messages $m_{w \to u}(x_u)$ received from u's set of neighbours $w \in N_u$.

Messages At each step, each agent in the network passes a message to each of its neighbours, consisting of a vector of values representing the sender's beliefs about each of the recipient's states. This involves sender u comparing the compatibility of each individual state x_u from its own set of states with each individual state x_v from recipient v's set of states, taking into account u's belief about its own state x_u, as well as the belief value about state x_u contained within the message passed from v to u in the previous step. Messages can therefore be interpreted as encoding both a compatibility component (through the pairwise cost) and a cost component (through the encoding of cost data in one's current beliefs, if the states are compatible).

$$\widetilde{m}_{u \to v}(x_v) = min_{x_u}\left(bel_u(x_u) - m_{v \to u}(x_u) + g_{uv}(x_u, x_v)\right) . \qquad \textbf{(3)}$$

Equation 3 show the process of calculating a message to be passed from agent u to agent v. $bel_u(x_u)$ corresponds to agent u's belief in its own state x_u. We subtract from this the belief passed from v to u about state x_u in the previous round of messages, represented as $m_{v \to u}(x_u)$. Finally, we add the pairwise cost incurred by agents u and v being in states x_u and x_v. We repeat this process for each of agent u's possible states, comparing them in turn to agent v's state x_v. Once the set of possible costs for state x_v dependent on u's set of states have been determined, we take the minimum of these values and add it to the vector of beliefs to be passed from agent u

to agent v. This process is repeated for each of v's possible states, resulting in a final vector of values to be passed from u to v. Before we perform allocation, we determine the "final state" of each agent – the state, at convergence, which the agent believes holds the lowest cost.

3.5 Allocation

Once the final states of each of the agents have been determined, we can perform the process of allocation. For each of the agents in the network, we remove edges leading to other agents which are not listed in their final state if there are no other producers/consumers of that good; in the case of agents being in the inactive state, we remove all of their edges. We then iterate through the agents once more, this time checking to see if, given the results of the previous stage of allocation, each producer was able to acquire all the goods in its set of input goods. If a producer is determined to have acquired an incomplete set, we remove the edge leading to their output good.

Allocation Value We determine the value of our allocations by the equation given below, where C is the set of consumers to acquire a good, V_c is the consumption value obtained by each of those consumers, P is the set of producers in the allocation who produce a good, and PC_p is the production cost of each producer p. This is equivalent to equation (1).

$$\sum_{c \in C} V_c - \sum_{p \in P} PC_p \ . \qquad \textbf{(4)}$$

4 Experiments

4.1 Network Structures

We test our belief propagation method over a variety of network structures, taken from [2]. For the purposes of evaluation, upon initialisation of each of the networks, the production cost of each producer is set to a decimal value drawn from the interval $U(0,1)$. These values are re-computed and changed after each run. Consumption values, taken from [2], are fixed at the values given underneath each consumer (C1, C2 and so on) in each of the following figures, over every run. We implemented our system using a combination of Java and MATLAB.

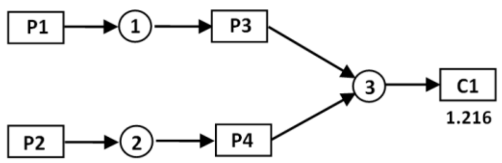


Fig. 2. Simple network, from [2]. Numbers below consumers indicate consumption values, which remain static for each network over each run. Production costs vary between runs and are drawn randomly from the interval $U(0,1)$.

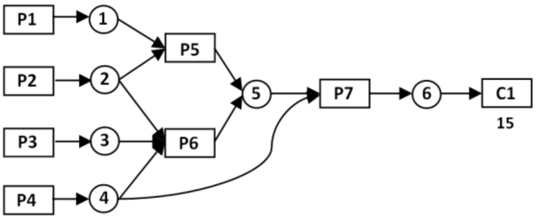


Fig. 3. Greedy-Bad network, from [2].

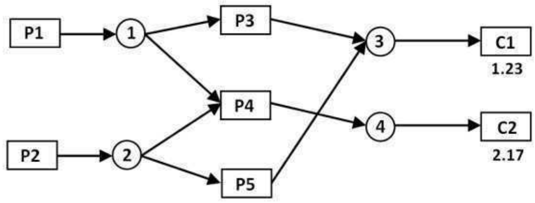


Fig. 4. Two-Cons network, from [2].

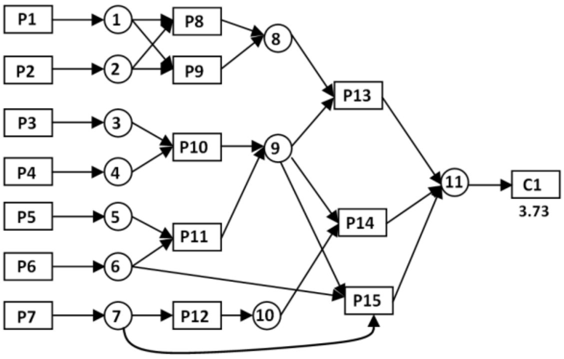


Fig. 5. Unbalanced network, from [2].

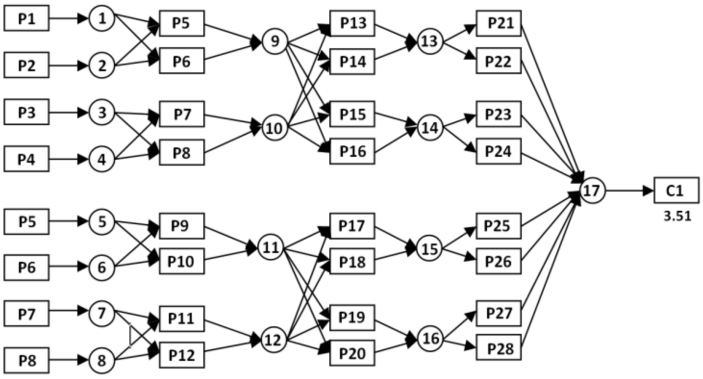


Fig. 6. Bigger network, from [2].

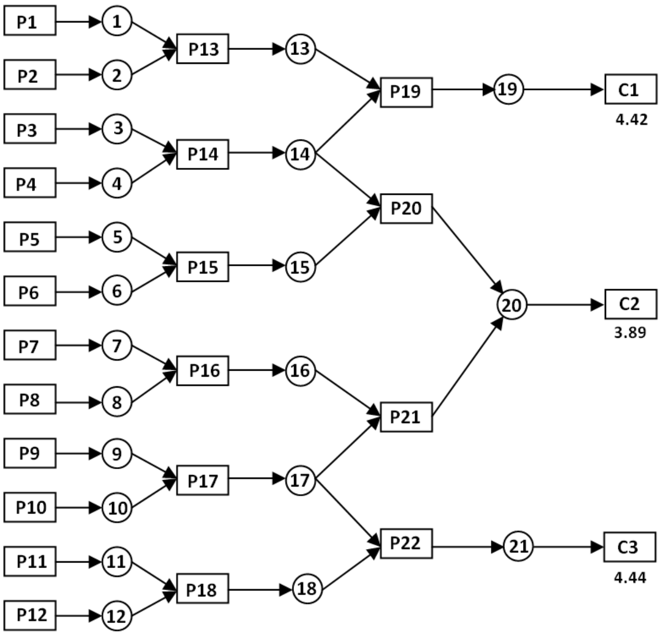


Fig. 7. Many-Cons network, from [2].

4.2 Performance Evaluation

To evaluate the performance of our method, we perform belief propagation on each network until a convergence is reached, using the final state of each agent as the basis for our allocations. If no convergence is reached, i.e. we see a continuing oscillation in the transmitted messages, we regard the result as zero-value allocation, indicating that no solution was found. We compare the value of our allocations to the optimal efficient value, calculated using a centralised technique, and to the results of the auction protocols given in [2]: SAMP-SB, and SAMP-SB-D. SAMP-SB-D is a modification of SAMP-SB which allows inactive producers to decommit from contracts for inputs for which they would pay a positive price, a situation referred to as a “dead end”. Decommitment allows for the avoidance of this potential source of inefficiency, though at the cost of rendering contracts non-binding. As in [2], we gather 100 results for each network, discarding runs in which the optimally efficient value is non-positive.

4.3 Efficiency

We first divide our results into efficiency classes: negative, zero, suboptimal and optimal. Recall equation 4, which allows us to determine the value of an allocation. The efficient allocation within a network, given a set of producer costs, is the one which maximises this value. We use the efficient allocation as a benchmark for the results we obtain using our loopy belief propagation method. We determine the efficient allocation for each run by using local search, finding each possible solution in the network and determining the most efficient. We classify our results using the belief propagation method as follows:

Negative: A negative efficiency result is an allocation in which the production costs of active producers exceeds the value that the consumer(s) obtain from acquiring their consumable good. This is caused by dead ends: the presence of inactive producers who acquire one or more input goods but do not produce an output, either due to no buyer being found for their potential output good, or due to the producer acquiring an incomplete set of input goods. SAMP-SB-D avoids the problem of dead ends – and thus negative efficiency – by allowing producers in such situations to decommit from contracts for their inputs.
Zero: A zero-value allocation is one in which our algorithm does not reach convergence, with all producers assigned to an inactive state, and as such is equivalent to a zero efficiency allocation produced by SAMP-SB or SAMP-SB-D. Zero valued allocations are more desirable than negative-valued allocations, but less desirable than suboptimal or optimal allocations.
Suboptimal: Suboptimal allocations are allocations in which a non-optimal solution was found. This can be caused by the presence of dead ends, or by finding an allocation without dead ends when an allocation which would have produced a higher value existed.
Optimal: An optimal allocation means that our algorithm was able find the allocation which produced the maximum efficiency available, meaning we achieved the same value as the centralised benchmark, determined by local search. There are no dead ends in optimal allocations.

5 Results

In keeping with our desire for as fair a comparison between the methods as possible, the efficiency classes of the results produced by SAMP-SB and SAMP-SB-D are near identical to those for our belief propagation method. For SAMP-SB and SAMP-SB-D, a zero result means that no solution was found, with no dead ends. This is equivalent to our zero result in which no convergence is reached. The definitions of negative, suboptimal and optimal allocations given in [2] are identical to ours. The ability for inactive producers to decommit from contracts under the SAMP-SB-D protocol means that there is no negative efficiency category for SAMP-SB-D.

We see from Table 1 that our belief propagation approach is able to match SAMP-SB's performance for network Simple while significantly outperforming it over all other networks tested. Due to the absence of producer surplus in our model, we make no attempt to distinguish between the existence of competitive equilibrium (CE) or otherwise in our results. Our producers aim to break even, as in [2], but are not limited in their attempts to do so by bidding strategy inefficiencies. Despite this, even if we compare our results with the best case for SAMP-SB, using only those results in which competitive equilibria exist, we are still able to show a significant advantage in the proportions of our runs showing optimal efficiency, with marked reductions in negative, zero and suboptimal solutions in almost all cases.

Table 1. Distribution of efficiency classes from Belief Propagation, SAMP-SB and SAMP-SB-D. Classes are Negative, Zero, Suboptimal and Optimal.

Network	Belief Propagation % of instances				SAMP-SB % of instances				SAMP-SB-D % of instances		
	Neg	**Zero**	**Sub**	**Opt**	**Neg**	**Zero**	**Sub**	**Opt**	**Zero**	**Sub**	**Opt**
Simple	0.0	0.0	0.0	100.0	0.0	0.3	0.0	99.7	0.3	0.0	99.7
Unbalanced	8.0	1.0	0.0	91.0							
CE					5.0	1.0	7.0	87.0	1.0	1.0	98.0
No CE					100.0	0.0	0.0	0.0	100.0	0.0	0.0
Two-Cons	0.0	0.0	0.0	100.0							
CE					11.0	0.0	6.0	83.0	0.0	3.0	97.0
No CE					18.0	0.0	78.0	4.0	1.0	95.0	4.0
Bigger	0.0	0.0	0.0	100.0	0.0	0.0	4.0	96.0	0.0	0.0	100.0
Many-Cons	0.0	0.0	0.0	100.0	27.0	0.0	56.0	17.0	0.0	2.0	98.0
Greedy-bad	0.0	7.0	0.0	93.0							
CE					4.0	0.0	21.0	75.0	1.0	0.0	99.0
No CE					100.0	0.0	0.0	0.0	100.0	0.0	0.0

Our results are also comparable to those produced by SAMP-SB-D, with similar efficiency class proportions between the two methods if only the results where competitive equilibria exist for SAMP-SB-D are compared. In this case, SAMP-SB-D generates optimal allocations slightly more frequently than belief propagation for networks Unbalanced and Greedy-Bad, though like SAMP-SB it struggles when competitive equilibria are not present. The performance of our method is aided by the fact that, when viewed as undirected graphs, networks Simple, Two-Cons and Many-Cons are all acyclic – recall that belief propagation is guaranteed to converge to the correct solution under such conditions. The performance of belief propagation on the other networks, however, shows that such favourable conditions are not a prerequisite for the production of good results.

Table 2 shows the average efficiency achieved by belief propagation, SAMP-SB and SAMP-SB-D as a fraction of the efficient value. Negative values indicate that runs over a network recorded negative average efficiency. For example, a result showing -1.0 average efficiency means the method achieved, on average, -100% of the maximum available allocation value. Once again, belief propagation essentially equals or significantly outperforms SAMP-SB for every network, capturing, with the exception of the unbalanced network, a higher proportion of the efficient value than SAMP-SB is able to, performing at 82.7% of the optimal or better for each network tested. As with the previous set of experiments, if our results are compared to only those where competitive equilibria are present for SAMP-SB-D, then we see that SAMP-SB-D is able to capture slightly more of the average efficiency than belief propagation for the Unbalanced and Greedy-bad networks, with essentially equal results for the other networks. Unlike SAMP-SB-D however, we are able to produce strong results, in most cases, regardless of the cost structure of the networks, as long as there exists a solution with a positive value.

Table 2. Fraction of average efficiency obtained by Belief Propagation, SAMP-SB and SAMP-SB-D in each network. Negative values indicate that the method achieved less than 0% of the available allocation value on average.

Network	Belief Propagation	SAMP-SB	SAMP-SB-D
Simple	1.000	0.997	0.997
Unbalanced	0.827		
CE		0.867	0.990
No CE		-20.080	0.000
Two Cons	1.000		
CE		0.733	0.986
No CE		0.268	0.686
Bigger	1.000	1.000	1.000
Many-Cons	1.000	0.120	0.996
Greedy-bad	0.854		
CE		-5.320	0.990
No CE		-18.230	0.000

6 Conclusions

In this paper, we present a new method for decentralized supply chain formation, using work by [2] as both a foundation for the structure of our networks, and as a comparison to our results. Our belief propagation method, involving decentralized message passing to propagate beliefs held by our agents, is able to perform at worst equally and, more frequently, significantly better in creating efficient allocations than an established approach utilizing ascending auctions [2], whilst making no assumptions of centralisation. Over every network structure tested, we were able to show that, even in the best case for the auction-based approach, our method is able to match or outperform the results it obtains, producing consistently optimal or near-optimal average efficiency results.

We believe that our method provides an interesting avenue for future research by merit of its ability to produce more efficient allocations than an established auction protocol in a comparable scenario, whilst operating in a decentralised manner. By allowing our agents to share limited information about their capabilities and production costs (equivalent to sale price, which means participants are revealing no more information about their private preferences than in an open auction) we are able to produce highly efficient allocations over a range of network topologies.

7 Future Work

In using [2] as a basis for our work, we traded potentially complex extensions in favour of an expressive graphical representation of supply chain networks and a clear basis for fair comparison. Potential extensions, therefore, might involve expanding the properties of goods, to take into account factors such as quality, quantity, delivery dates and default penalties. Producers could be improved by implementing properties to model production capacity and the possibility of strategic behaviour – at present all agents are truth-telling – while consumers might be imbued with richer preferences over the goods available, using principles taken from consumer behavior theory. A temporal aspect could be introduced, with trading relationships forming and dissolving over time, based upon scarcity of goods and inherent preferences, with trust and reputation interesting issues to be taken into account.

8 References

1. Nachira F., Dini P., Nicolai A., Le Louarn M., and Rivera Lèon L.: Digital Business Ecosystems. European Commission, Luxembourg (2007)
2. Walsh W.E. and Wellman M.P.: Decentralized Supply Chain Formation: A Market Protocol and Competitive Equilibrium Analysis. JAIR 19, 513-567 (2003)
3. Davis, R. and Smith, R.G.: Negotiation as a metaphor for distributed problem solving. Artificial Intelligence 20 (1), 63-109 (1983)
4. Wang M., Wang H., Vogel D., Kumar K., and Chiu D.K.W.: Agent-based negotiation and decision making for dynamic supply chain formation. Eng. Apps. of A.I. 22 (7), Distributed Control of Production Systems. 1046-1055 (2009).
5. Jiang T., Foley R., Yao X., and Tianfield H.: An extended contract net mechanism for dynamic supply chain formation and its application in China petroleum supply chain management. Multiagent and Grid Systems 2, 183-207 (2006)
6. Walsh W.E., Wellman M.P., and Ygge F.: Combinatorial auctions for supply chain formation. Second ACM Conf. on Electronic Commerce, 260-269 (2000).
7. Cerquides J., Endriss U., Giovannucci A. and Rodríguez-Aguilar J.A. Bidding Languages and Winner Determination for Mixed Multi-Unit Combinatorial Auctions. In Proc. of IJCAI (2007).
8. Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, (1988).
9. Murphy P., Weiss Y. and Jordan M.I. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In Proc. of UAI, 467-475 (1997).
10. Crick C., and Pfeffer A.: Loopy belief propagation as a basis for communication in sensor networks. Uncertainty in Artificial Intelligence (2003).
11. Farinelli, A., Rogers, A., Petcu, A., and Jennings, N. R.: Decentralised coordination of low-power embedded devices using the max-sum algorithm. In Proc. of the 7th international Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2 (2008).
12. Frey, B. J. and MacKay D. J. C.: A revolution: Belief propagation in graphs with cycles. In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), Advances in Neural Information Processing Systems, Volume 10. MIT Press (1998)

Search Costs as a Means for Improving Market Performance

David Sarne and Yonatan Aumann

Computer Science Department
Bar-Ilan University
Ramat Gan, 52900 Israel

Abstract. In this paper we study the benefits of search costs in distributed multi-agent systems (MAS). These costs, often associated with obtaining, processing and evaluating information relating to other agents in the environment, can be either monetary or manifested in some tax on the agent's resources. Traditionally, such costs are considered as market inefficiency, and, as such, aimed to be reduced to the minimum. Here we show, in contrast, that in many MAS settings the introduction of search costs can actually improve market performance. This is demonstrated in three different settings. First we consider one-sided and two-sided (equilibrium-driven) search applications. In both settings we show that, while search costs may decrease the individual agents' outcomes, the overall *market throughput* may actually improve with the introduction of such costs. Next, we demonstrate a setting where, somewhat paradoxically, the introduction of search costs improves both the overall market throughput and the utility of each and every individual agent. We stress that we assume that the proceeds from the search costs are *wasted*, with no one directly benefiting from them. The importance of the results is for the design of MAS systems, where in many cases one should consider deliberately increasing (potentially artificially) the search friction to some desired level in order to improve the system's performance.

1 Introduction

In many cases, market friction and seemingly "inefficiency -based" mechanisms can be used for establishing desired market behaviors. For example, minimum wage has been shown as a mechanism that can, in some cases, improve employment rates [16] and Vickrey Clarke Groves (VCG) ensures assignment of items in a socially optimal manner and truth telling by requiring significant transfer of payments from agents to the center [8]. Similarly, particular taxes (known as marginal cost pricing) can eliminate inefficiencies of equilibria as demonstrated in transportation economics [5].

In this paper, we learn the positive role that search costs, incurred as part of a repeated search process, can play in enhancing performance of markets. In many multi-agent systems (MAS), agents may incur a cost when engaged in obtaining, processing or reasoning about information related to their environment and other agents that can be found in it [14, 9]. These costs can be either in the form of explicit monetary payments or resources that need to be consumed in order to carry out these activities (e.g., computational, communication). In economics and operations research such costs are often referred to as "search costs" or "environment friction" as they represent the inefficiency and lack of transparency of the environment the agents are operating in [3, 25]. Typical examples include distributed matching applications (where the interaction with other agents is costly) [1], shopbots (where price and product information must be actively gathered) [14], and reputation systems (where agents may need to pay a fee for querying the system) [13].

Traditionally, search costs are regarded as market *inefficiency*, and associated with reduced market performance. Indeed, in the presence of search costs, a rational player would not aim to find the optimal element in the environment, but rather settle for the "good enough", beyond which the marginal benefit of continuing the search is less than the search cost. Thus, search costs promote sub-optimal results (or so it would seem). As such, the traditional wisdom is that when designing a MAS environment, search costs should be avoided or reduced as much as possible. Taking eCommerce as an example, most researchers see a great benefit in the ability of electronic marketplaces to lower the buyers' cost to obtain information (e.g. about the price and product features) from multiple sellers, as well as the sellers' reduced costs to communicate their information [4]. The lowered buyer's search cost is associated in this case with an increased economic efficiency and enable new markets to emerge [4]. Similarly, many systems have been introduced in which central mechanisms or mediators are used in order to supply the agents full immediate information concerning market opportunities, eliminating the need to engage in costly search [10, 2].

In this paper we show that, not withstanding the above, search costs — "friction" — can also be beneficial, and may *improve* market performance, in some cases, when applied appropriately. We show this on two levels. First, we note that while each player's individual goal is to maximize its own utility, and as such may suffer from search costs, the market designer should consider the overall welfare produced by the system as a whole. This, we argue, is not the average utility of the individual players, but rather the total *utility throughput*, i.e. the aggregate utility *per time unit*. Note that in search settings the players' individual utilities and the utility throughput are not necessarily directly correlated, as players can stay in the system for longer or shorter time periods. With this understanding in mind, we show two examples where introducing search costs increases the market throughput, and hence also the social welfare. Specifically, we consider classical one and two-sided search settings, and show that in both, search costs can improve market throughput. It is important to stress that throughout the paper we assume that the proceeds from the search costs are discarded and do not benefit anyone in the system.

Next, we show that even when considering the benefit of each individual player, there are cases where search costs can increase the expected utility of *each and every* player in the system. This seemingly paradoxical phenomena is exhibited in a setting of two-sided search with multiple search rounds, each round consisting of several parallel searches. We show that if the cost for each round is given, then it may be beneficial to artificially increase (up to a certain level) the cost of each individual parallel search within the round, and that all players simultaneously benefit from this. The exact details are provided in Section 4.

The common rational to all these examples is that in some cases it may be beneficial (either to the system designer or to all) to reduce the amount of searches performed by the players, but when dealing with self-interested agents this cannot be directly dictated. Search costs provide a means to incite players to perform less searches. What is interesting in these examples is that even when the proceeds of the costs are wasted, the benefit of the reduced search outweighs the loss.

The analysis given differs from classical examples of seemingly "inefficiency -based" mechanisms as it inherently derives from the modeling of the search strategy as a repeated process, and applies to the various models investigated as part of classical search theory [21, 15, 26, 6, 11, 18, 19]. The main contribution of this paper is thus in establishing the notion of market throughput within the context of search-based MAS applications and demonstrating that market friction in the form of search cost can often play a positive role market-wise, despite the reduction in individual performance experienced by the agents. In some cases, search costs can actually improve both individual and market performance. Therefore, when designing a new MAS, the system designer should carefully consider the option of deliberately generating some inefficiency in the system in the form of search costs.

2 One-Sided Search

Consider an environment with N homogeneous servers, and an infinite incoming flow of homogeneous agents requesting service from these servers.[1] In practice, the servers may represent medical specialists and agents - the patients, or servers may represent online retailers and the agents - shopbots. Each agent can send a query to any of the servers, where the utility of reply x received by agent j from a server i is randomly drawn from a distribution characterized by a probability distribution function (p.d.f.) $f(x)$, and cumulative distribution function (c.d.f.) $F(x)$, defined over the interval $(\underline{x}, \overline{x})$. For example, patients may seek for a second opinion from medical specialists, where the quality of the diagnosis received varies, and shopbots may query different retailers for price information. For simplicity we assume that all servers and all agents are homogeneous, and thus share the functions $f(x)$ and $F(x)$, and that an agent's utility from the returned value x equals x. We assume that the time it takes to execute a single query on any of the servers is fixed and WLOG will be considered a time unit. Based on the revealed utility of x, the agent can decide to send an identical query to an additional server, again obtaining a utility drawn from the same distribution function. This process continues until the agent decides that there is no point in sending any further queries, or until all N servers have been queried. In both cases, the resulting utility is the maximum among the utilities obtained from the queries sent. We assume that queries sent by agents that have already received service have priority over those of newly arriving agents.

Agents are assumed to be self-interested, and thus aiming to maximize their expected utility. The system designer, on the other hand, should be interested in the aggregate utility. Since the system is assumed to continue working indefinitely, the aggregate is not well defined, and what we are really interested in is the average *throughput*, i.e., the average aggregate utility *per time unit*. Formally, denote by $A(\underline{t}, \overline{t})$ the set of agents that have completed their querying process within the time interval $(\underline{t}, \overline{t})$, and by $U(A_i)$ the utility obtained by agent $A_i \in A(\underline{t}, \overline{t})$. The average throughput of the system during the interval $(\underline{t}, \overline{t})$, denoted $T(\underline{t}, \overline{t})$, is defined as:

$$T(\underline{t}, \overline{t}) = \frac{\sum_{A_i \in A(\underline{t}, \overline{t})} U(A_i)}{\overline{t} - \underline{t}} \tag{1}$$

[1] Since the purpose of this paper is to investigate the effect of search costs over throughput of MAS, we deliberately assume that the demand for the servers is infinite. Without this assumption, the analysis necessitates including queueing theory aspects, which add many complexities without adding to the understanding of the core phenomena illustrated.

For a system working indefinitely the throughput is defined as:

$$T_\infty = \lim_{\bar{t}\to\infty} \frac{\sum_{A_i \in A(0,\bar{t})} U(A_i)}{\bar{t}} \tag{2}$$

Since the agents are homogeneous and the setting is a symmetric game, they all use the same optimal querying strategy. Therefore, the following holds for large $(\underline{t}, \bar{t})$ intervals:

$$\sum_{A_i \in A(\underline{t},\bar{t})} U(A_i) = \frac{E[U(A_i)] \cdot N \cdot (\bar{t} - \underline{t})}{E[q]}$$

where $E[q]$ is the expected number of queries sent by each agent (identical to all agents as the agents are homogeneous). Therefore:

$$T(\underline{t}, \bar{t}) = \frac{N \cdot E[U(A_i)]}{E[q]} \tag{3}$$

Note that the right-hand side of Equation (3) is independent of $(\underline{t}, \bar{t})$. Hence, the same value also holds for the limit T_∞.

Suppose that each agent queries exactly $k \leq N$ servers, i.e., $E[q] = k$. The expected utility of each agent in this case is the expected maximum of a sample of size k taken from the distribution $f(x)$, which equals $\int_{y=\underline{x}}^{\overline{x}} y f_k(y) dy$, where $f_k(x)$ is the probability distribution function of the maximum of a sample of size k drawn from $f(x)$.[2] Therefore:

$$T_\infty = \frac{N \int_{y=\underline{x}}^{\overline{x}} y f_k(y) dy}{k} \tag{4}$$

Obviously, when there is no cost associated with querying a server, agents will query all servers (as the expected utility $\int_{y=\underline{x}}^{\overline{x}} y f_k(y) dy$, increases with k) . In this case, substituting $k = N$ in Equation 4, the average throughput reduces to the expected utility of a single agent, i.e., $T_\infty = \int_{y=\underline{x}}^{\overline{x}} y f_N(y) dy$. While this strategy maximizes individual agents' utility, it is certainly not so in terms of overall throughput. Given the limited number of servers and infinite demand, having each agent take advantage of all servers is clearly sub-optimal.

Therefore, for the sake of optimizing throughput, it would be best for a market designer to limit the number of queries sent by each agent (the throughput is inversely related to $E[q]$ - see by Equation (4)). While this is possible in some systems, in many systems, where agents are self-interested, the system designer cannot directly dictate such behavior. Introducing search costs is an implicit mechanism by which a system designer may incite agents to execute less queries. On the other hand, search costs themselves also reduce the throughput, as their proceeds are assumed to be wasted, and must thus be deducted from the overall utility. We show that nonetheless a certain level of search costs may still *increase* the overall throughput. The details follow.

In the presence of search cost, agents consider the tradeoff between the marginal utility of each additional query and its cost. This problem, of finding the agents' optimal querying strategy in the presence of a search cost, can be mapped to a variant of the

[2] The value of $f_k(x)$ is the derivative of the c.d.f. of the maximum of a sample of size k, $F_k(x)$. Since $F_k(x) = (F(x))^k$, we obtain: $f_k(x) = k(F(x))^{k-1} f(x) dx$.

known "Pandora's problem" [26].[3] Accordingly, in the optimal search strategy, each agent searches sequentially using a reservation value z, querying (randomly) additional servers as long as the maximum utility obtained so far is smaller than z or until all servers have been queried. Given a fixed search cost c (incurred when sending a query to a server) and assuming the cost is additive and expressed on the same scale as utilities, the reservation value z that maximizes the agent's expected utility can be extracted from:

$$c = \int_{y=z}^{\overline{x}} (y-z)f(y)dy \tag{5}$$

The expected number of queries each agent sends according to the above strategy is given by:

$$E[q] = \sum_{i=1}^{N} iP_q(i) \tag{6}$$

where $P_q(i)$ is the probability that exactly i queries will be sent ($P_q(i) = ((F(z))^{i-1}(1-F(z))$ for $i < N$ and $P_q(N) = (F(z))^{N-1}$ for $i = N$).

Hence, the expected utility of each agent using the optimal search strategy when incurring a search cost, denoted $V_{overall}$, is:

$$V_{overall} = E[x/x>z]\Big(\sum_{i=1}^{N-1} P_q(i) + P_q(N)(1-F(z))\Big) + P_q(N)F(z)E[Max(x_1,...,x_N/x_i<z)] - cE[q] \tag{7}$$

where $E[x/x > z]$ is the expected utility of a single query if above z ($E[x/x > z] = \int_{y=z}^{\overline{x}} yf(y)/(1-F(z))$) and:[4]

$$E[Max(x_1,...,x_N/x_i<z)] = \int_{\underline{x}}^{z} yf_N(y/y_i<z\forall i)dy = \int_{\underline{x}}^{z} \frac{yNf(y)}{F(z)}\Big(\frac{F(y)}{F(z)}\Big)^{N-1}dy \tag{8}$$

The expected throughput is thus $\frac{N(V_{overall}-cE[q])}{E[q]}$.

Figure 1 depicts the expected system throughput as a function of the search cost c, when using a uniform distribution function - $f(x) = 1, F(x) = x, 0 \le x \le 1$ - and ten servers ($N = 10$). The optimal market setting is obtained with a search cost of $c = 0.12$, in which case the market throughput is 2.5. Comparatively, if there is no search cost ($c = 0$) then each agent samples all servers, i.e., $E[q] = 10$. The expected throughput in this case is 0.91 (the maximum of a sample of size 10 drawn from a uniform distribution), significantly worse then when using $c = 0.12$. On the other hand, if market designer could *dictate* the number of queries per agent, then a throughput of 5 could have been obtained, by forcing each agent to query a single server (resulting with individual utility of 0.5, which is smaller than when using $c = 0.12$).

Figure 2 depicts the optimal search cost to be used and the resulting average throughput per server as a function of the the number of servers available, N, when using a uniform distribution function ($f(x) = 1, F(x) = x, 0 \le x \le 1$). The middle curve represents the marginal improvement to the overall throughput due to the addition of

[3] Alternatively, the problem can be mapped to a one-sided sequential search problem with a finite decision horizon and full recall [15, 21] and solved using backward induction, though with a greater complexity.

[4] Using $F_N(x/x<z) = (F(x/x<z))^N$, $f_N(x/x<z) = Nf(x/x<z)(F(x/x<z))^{N-1}dx$.

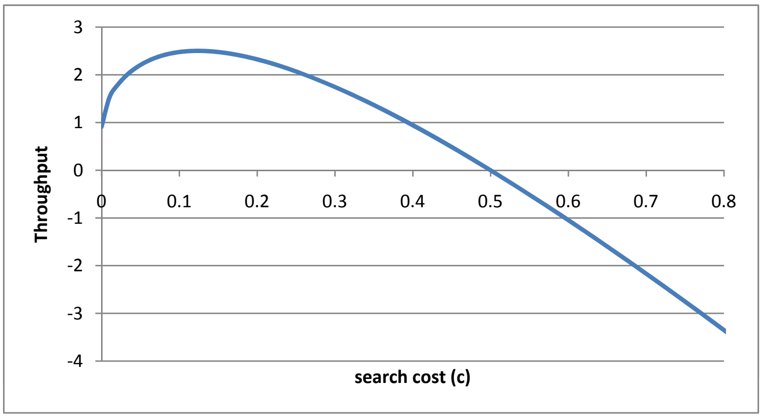


Fig. 1. Throughput as a function of search cost

each server (e.g., the transition from one server to two servers will be accompanied with a 0.2 addition to the overall system throughput). As can be observed from the figure, while adding more servers increases the overall throughput, it has no consistent marginal contribution to the overall throughput. For example, the transition from two to three servers is accompanied with the least marginal improvement in the system's throughput.

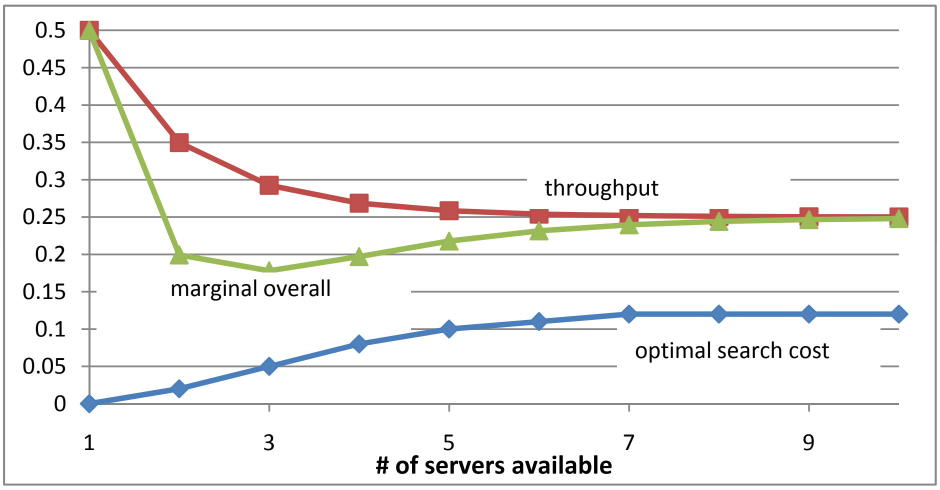


Fig. 2. Throughput per server, marginal improvement to throughput and optimal search cost as a function of the number of servers in the one-sided search model

As can be seen from Figures 1 and 2, the use of search cost in this model can improve the average throughput significantly, though the optimal magnitude of search cost to be used should be determined by analyzing the agents' resulting search strategies.

3 Two-Side Search

We next show that the benefits of search costs to market throughput are not limited only to settings where the agents are competing for limited resources, as in the previous section. In this section we demonstrate that search cost can also improve throughput in distributed matching environments, where the agents' value is generated from partnering with other agents, and where the search strategies are affected (in part) by the strategies of the other agents in the market. The model we use is a standard two-sided distributed search model, in which self-interested agents search for appropriate partners to form mutually acceptable pairwise partnerships [7]. The model postulates an environment populated with an infinite number of self-interested fully rational homogeneous agents[5]. Any agent A_i can form a partnership with any other agent A_j in the environment. A partnership between A_i and A_j results in utility $U(A_j \hookrightarrow A_i)$ for agent A_i and $U(A_i \hookrightarrow A_j)$ for agent A_j, where both $U(A_j \hookrightarrow A_i)$ and $U(A_i \hookrightarrow A_j)$ are drawn from a distribution characterized by a p.d.f. $f(x)$ and c.d.f. $F(x)$. The agents are assumed to know the utility distribution function $f(x)$. However, in the absence of central information source agents cannot tell a-priori what utility can be gained by a partnership with any specific agent. Therefore, the only way by which an agent A_i can learn the value it can obtain from partnering with a specific other agent A_j is by directly interacting with agent A_j. Since each agent in two-sided search models has no prior information concerning any of the other agents in its environment, it initiates interactions (i.e., search) with other agents randomly. The two-sided search model assumes that the agents are satisfied with having a single partner. Hence, once a partnership is formed the two partnering agents terminate their search process and leave the environment.

We define a search *round* as the interval in which the agent interacts with another agent and learns the utility it can obtain by partnership with it. Based on the learned values, the agents decide whether to commit or reject the partnership. If both agents mutually commit to the partnership, then the partnership is formed and both agents gain the corresponding utilities. If an agent does not form a partnership in a given round, it continues to the next search round and interacts with another agent in a similar manner.

We now define the market throughput in such a setting. Since the number of agents is infinite, it is meaningless to consider the total aggregate utility. Rather, we define the *market throughput* as the average expected utility per-agent, per time unit. If there would be no search costs, the agents' equilibrium strategy is to commit to partnerships only when the utility offered by that partnership is the maximum possible (i.e., $\overline{U}$). In such case, for any non-atomic p.d.f. the probability of actually attaining this maximum is zero, and partnerships will never be formed. Hence, in this case the expected throughput is zero. Adding a search cost to the model reduces each agent's expected utility, but can improve the overall throughput. We assume utilities and costs are additive and that the agents are trying to maximize their overall utility, defined as the utility from the partnership formed minus the aggregated search costs along the search process. Suppose that the agent's cost of interacting with another agent is c.

[5] The infinite number of agents assumption is common in two-sided search models (see [7, 23, 22]). In many domains (e.g., eCommerce) this derives from the high entrance and leave rates. In this case, the probability of running into the same agent twice is negligible.

The agents' optimal strategy in such model is reservation-based [22, 17, 26, 19].[6] The reservation value is used as a threshold for accepting/rejecting potential partnerships. The a reservation value is equal to the expected utility to be obtained from resuming the search; the agent will always prefer committing to an opportunity greater than the expected utility of resuming the search and will always prefer to resume the search otherwise.

Since the agents are not limited by a decision horizon, and their search process does not imply any new information about the market structure (e.g., about the utility distribution of future partnership opportunities), their strategy is stationary - an agent will not accept an opportunity it has rejected beforehand and will use the same reservation value along its search.

We now derive the general formula for the optimal reservation value. The expected utility of an agent when using a reservation value x, assuming all other agents are using reservation value x', denoted $V(x, x')$, is given by:

$$V(x,x') = -c + (1 - F(x')) \int_{y=x}^{\overline{x}} yf(y)dy + (1 - (1 - F(x'))(1 - F(x)))V(x,x') \quad (9)$$

Here, $(1 - F(x'))$ is the probability that the agent is found adequate by the other agent, in which case the partnership will form only if the value obtained from the partnership is greater than x. Otherwise, if the utility obtained from partnering with the other agent is below x or the other agent obtains a utility lesser than x' (i.e., with probability $(1-(1-F(x'))(1-F(x)))$), the search is resumed and the expected cost is $V(x, x')$. Using some simple mathematical manipulations, Equation 9 can be expressed as:

$$V(x,x') = \frac{-c + (1 - F(x')) \int_{y=x}^{\overline{x}} yf(y)dy}{(1 - F(x'))(1 - F(x))} \quad (10)$$

Differentiating the last equation according to x and setting it to zero, we obtain (using integration by parts) that the optimal reservation value to be used when all other agents are using reservation value x' can be derived from:

$$c = (1 - F(x')) \int_{y=x}^{\overline{x}} (1 - F(y))dy \quad (11)$$

The equilibrium reservation value, x^*, is obtained by setting $x' = x = x^*$ in Equation 11. The equilibrium expected utility of each agent is thus given by:

$$V(x^*) = \frac{-c + (1 - F(x^*)) \int_{y=x}^{\overline{x}} yf(y)dy}{(1 - F(x^*))^2} \quad (12)$$

Once search cost is introduced, the expected time it takes until a partnership is formed becomes finite, and its value is: $\frac{1}{(1-F(x^*))^2}$. The throughput is thus $\frac{V(x^*)}{(1-F(x^*))^2}$.

Figure 3 depicts the expected average throughput and individual agents' utility in the two-sided search model as a function of the search cost c, when using a uniform distribution function ($f(x) = 1, F(x) = x, 0 \leq x \leq 1$). As can be observed from the

[6] Notice the reservation value used here is different from a reservation price concept (that is usually used as buyers' private evaluation). While the reservation price represents an agent's valuation of its own utility from a given opportunity, the reservation value is a threshold defined over the objective (or common) value of the opportunity.

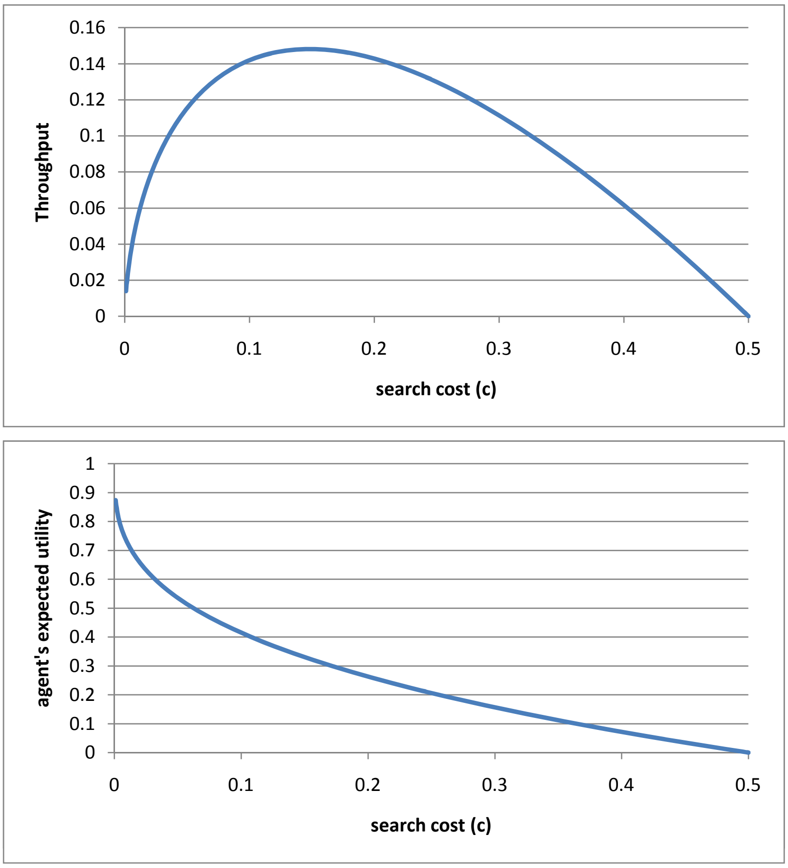


Fig. 3. Throughput and agents' utility as a function of search cost in two-sided search

example, the introduction of search cost up to some extent (0.5 in this case) improves system throughput. The optimal market setting is the one where the search cost is $c = 0.148$, in which case the expected throughput is 0.148. The agents' expected utility in this case drops to 0.333 and their expected search length (i.e., the number of search rounds) is in this case 2.25. Comparatively, when there is no search cost ($c = 0$) the expected throughput is zero, but each agent's (theoretical) expected utility is 1. This is yet another example for a case where an increase in the search cost can improve overall throughput however with the price of harming individual utilities. If the agents were fully cooperative and obey the market designer's instructions, a throughput of 0.5 could be attained, by having each agent search for a single search stage and then commit to whatever partnership it is offered. However, this behavior is not individually rational.

4 Search Costs Benefiting Individual Performance

In the two previous sections we exhibited settings where the market throughput is improved by introducing search costs. This is of interest to a market designer who's goal is to maximize the overall social welfare. However, in both previous examples, the search costs did reduce the agents' individual utilities. In this section we give a setting where, somewhat paradoxically, the introduction of search costs improves both the overall throughput and each and every player's individual expected utility. We exhibit this in the model introduced in [22], which is an extension of the standard two-sided search model. We now briefly review this model.

As in the standard two sided search, the model considers an environment populated by an infinite number of agents, each seeking a single partner, and utilities drawn from a distribution function (as in the previous section). The difference in the [22] model is in the search process. The model postulates a two-leveled search process, as follows. The search is conducted in discrete *rounds*. Within each round, each agent can choose to meet *in parallel* any number of other agents, and learn the utility associated with partnering with any of them. Given this information, each agent chooses if and with whom to partner. If two agents both choose to pair with one another then they obtain the said utilities and leave the system. Otherwise, they continue to the next search round. It is assumed that each agent make its decision independently of the decisions of all other agents, including potential partners. Furthermore, the agent can choose to pair with at most one of the agents it meets in a search round, and due to the synchronous nature of the mechanism has to reject all the rest of the agents met in that search round. There are two potential costs associated with each round: α - a fixed per-round cost, and β - an additional cost associated with each parallel probe in the round. Thus, if an agent chooses to meet with N potential partners then its total cost for the round will be $\alpha + N\beta$. The values of α and β are assumed to be the same for all agents.

Since the agents are not limited by a decision horizon and can control the intensity of their search, and the interaction with other agents does not imply any new information about the market structure, their strategy is stationary - an agent will not accept an opportunity it has rejected beforehand. Thus agents will use a reservation value strategy.

For analysis purposes, we'll use several notations. A strategy of sampling N other agents in each search round, and acting according to a reservation value x_N will be denoted (N, x_N). The expected utility of an agent when using strategy (N, x_N), will be denoted $V(N, x_N)$. Clearly, since all utilities for all agents are randomly selected from an identical distribution, there is no sense for any agent to consider partnering with anyone but the maximum of any given round. Thus, if an agent chooses to meet with N other agents in a given round, then the possible utility it can obtain during this round is distributed as the maximum of N random variables from the distribution with p.d.f. $f(x)$. As in Section 2 we use $f_N(x)$ and $F_N(x)$ to denote the p.d.f. and c.d.f. of the maximum distribution, respectively.

We start by formulating the expected utility for the agent when using a strategy (N, x_N), given that the strategy (k, x_k) is being used by all other agents in the environment. The expected future utility $V(N, x_N)$ is:

$$V(N, x_N) = -\alpha - \beta N + \frac{1}{k}(1 - F_k(x_k)) \int_{y=x_N}^{\overline{x}} y f_N(y) dy + (1 - \frac{1}{k}(1 - F_k(x_k))(1 - F_N(x_N)))V(N, x_N) \tag{13}$$

Here, $\frac{1}{k}(1 - F_k(x_k))$ is the probability that the utility obtained by the agent associated with the best value in the N-agents sample of agent A from partnering with agent A is greater than the reservation value x_k that is used by that agent and the maximum

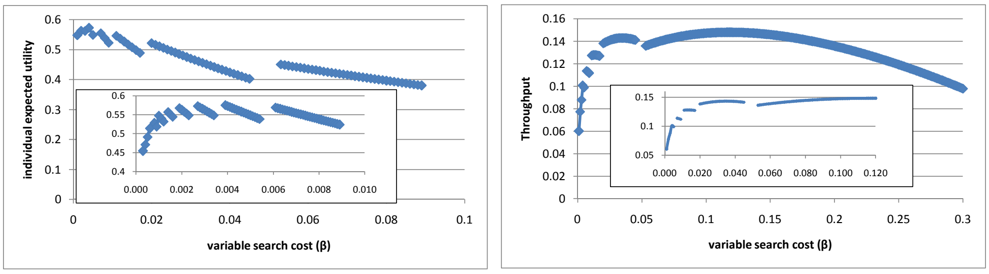


Fig. 4. (a) The expected utility of each agent as a function of the search cost used; (b) The throughput as a function of the search cost used.

in its k-agents sample. Similarly, the probability $(1 - F_k(x_k))(1 - F_N(x_N)))V(N, x_N)$ applies to any scenario other than the one where both agents choose to pair with each other, in which case the agent resumes its search with an expected utility $V(N, x_N)$. Equation 13 can also be formulated as:

$$V(N, x_N) = \frac{(1 - F_k(x_k)) \int_{y=x_N}^{\overline{x}} y f_N(y) dy - \alpha k - \beta N k}{(1 - F_k(x_k))(1 - F_N(x_N))} \tag{14}$$

Deriving Equation 14 according to x_N and applying several mathematical manipulations, we obtain that the agent's optimal reservation value x_N, satisfies:

$$\alpha k + \beta N k = (1 - F_k(x_k)) \int_{y=x_N}^{\overline{x}} (1 - F_N(y)) dy \tag{15}$$

Since all agents are homogeneous, the equilibrium strategy is the pair (N, x_N) that satisfies Equation 15 when substituting $k = N$ and $x_k = x_N$. While for each N value there is potentially a corresponding x_N value that satisfies the latter, the equilibrium strategy is the one yielding the maximum expected utility according to Equation 14 (for more details of the analysis of the model the reader is referred to [22]).

As in the former section, since the number of agents is infinite, the *market throughput* is defined as the average expected utility per-agent, per time unit. Figure 4 depicts the expected individual utility of each agent and the throughput as a function of β - the cost for each parallel search within a round, when using a uniform distribution function $(f(x) = 1, F(x) = x, 0 \leq x \leq 1)$ and $\alpha = 0.03$. The small graphs inside each of the graphs are enlargements of the originals over the more interesting parts of the graphs. As can be observed from the graphs, with α fixed, each and every individual agent actually benefits from the introduction of search costs into the market. The optimal individual utility of 0.575 is obtained for $\beta = 0.004$, in which case each agent meets with 5 possible partners in each round and the throughput is 0.1. The maximum throughput (0.148) is obtained when using a search cost of 0.118, yielding a utility of 0.333 to the agents. It is notable that the change in system's throughput as well as in individual expected utility is not consistent over large portions of the interval. The market designer should thus be careful when considering any change (either an increase or a decrease in search cost) in the setting represented by this example, as there is no way to predict the usefulness of such a change. In order to understand the implications of a suggested deviation from one value to another over the horizontal axis a direct calculation is required.

5 Related Work

Search is an inherent process in MAS, in particular when there is no central source that can supply full immediate reliable information on the environment and the state of the other agents that can be found. The introduction of search costs into MAS models leads to a more realistic description of MAS environments. In particular, search cost is highly recognized in eCommerce environments where agents need to invest/consume some of their resources in order to obtain information concerning the good or the transaction offered by other prospective agents [3, 9]. The overall agreement is that despite the significant reduction in search costs in MAS, due to recent advances in communication technologies, these cannot be ignored completely [3, 14].

Optimal search strategies for settings where individuals aim to locate an opportunity that will maximize the expected utility, while minimizing the associated search costs have been widely studied ([17, 15], and references therein). Within the framework of search theory, three main clusters of search models can be found. These are (a) the fixed sample size model; (b) the sequential search model; and (c) the variable sample size model. In the fixed sample size model, the searcher executes a single search round in which it obtains a large set of opportunities simultaneously [24] and chooses the one associated with the highest utility. In the sequential search strategy [21, 15], which for the general finite decision horizon case is also known as "Pandora's Problem" [26], the searcher obtains a single opportunity at a time, allowing multiple search stages. Several attempts were made to adopt the fixed sample size search [14] and the sequential search [9] models in agent-based electronic trading environments associated with search costs. In these cases the main focus was on establishing the appropriate characteristics of the environment and search strategy rather than the computational aspects of extracting it. Last, the variable sample size search method [6, 11, 18, 19] suggests a combined approach in which several opportunities are obtained during each search period.

In an effort to understand the effect of dual search activities in such models, the "two-sided" search research followed. This notion was explored within the equilibrium search framework [1, 7, 23]. While the literature in the area of one-sided and two-sided is rich and thorough, its focus is individual performance and search cost (often modeled as the discounting of gains) is considered as a non-favorable factor.

Finally, it is notable that the role of friction in distributed environments has been studied in several contexts before. For example, many authors rationalize the price dispersion (i.e., variation in prices across sellers of the same item, holding fixed the item's characteristics) observed in both offline and online markets by the cost for consumers to gather information about prices [24, 20, 12]. Others, have shown that an increase in the minimum wage, which is often considered by economics to be market inefficiency, can have positive employment effects [16]. In the auctions domain, the Vickrey Clarke Groves mechanism ensures several desired bidding characteristics by requiring significant transfer of payments from agents to the center [8]. Nevertheless, none of the above work has considered the positive effect search costs may have on system throughput.

6 Discussion and Conclusions

The main focus of the paper is in exhibiting that search costs are not necessarily and universally harmful to the system's performance. The illustrations in Sections 2-4 exhibit the fact that search costs can also play a positive role in improving market performance. Sections 2 and 3 exhibit the benefits of search costs for improving throughput.

The last example demonstrates that the improvement in system's throughput does not necessarily have to be at the expense of individual utilities. These results are non-intuitive, as traditionally search costs are considered to be market friction and as such their reduction is intuitively favorable. The examples given in the paper suggest that market designers should not take this latter claim as a general truth.

Notwithstanding, it is notable that the introduction of the measure of market *throughput* as a key measure for evaluating the performance and effectiveness of MAS systems is by itself an important contribution to the research of MAS. Unlike individual or collective agents' utilities, the throughput focuses in the value generated by the system as a whole over time. As such, we believe that this measure should be central for MAS designers, and understanding its behavior is crucial for the success of future mechanisms. As evident from Sections 2-4, the throughput is not necessarily correlated with individual utilities as these lack the aspect of time.

The paper relies on three established models from the "search theory" research area to support its main claims. Justification and legitimacy considerations for the applicability of these models to day-to-day settings were widely discussed in the literature we referred to throughout the paper.

It is notable that search costs can have many forms, and there are various methods for the market designer to control them. For example, search costs can be introduced as a payment an agent needs to pay in order to meet other agents or obtain a service, additional communication and computational overhead that result from the interaction protocol and even a payment per time unit for operating in the system. In this paper we adopted a pessimistic approach that assumes that the proceeds from any search costs are wasted and do not benefit anyone. In many cases, However, the proceeds of these costs can also be somehow redistributed back to the agents (e.g., equally split the proceeds among all agents when leaving the system - leaving their searching strategy unaffected). This could further improve individual utilities and the system's throughput.

Generally, as can be seen from the analysis given, the introduction of search costs should be carefully considered and their optimal magnitude should be calculated taking into consideration the resulting changes in the agents' strategies and equilibrium considerations whenever applicable. When search costs are already an inherent part of the system, there is no general answer for whether or not a decrease in these costs will improve system performance. In some settings, an increase rather than a decrease can actually contribute to improving throughput. In other cases, a decrease in search costs can contribute to improving system throughput, however decreasing these costs beyond to a certain point can result with the opposite effect. The analysis methodology given in this paper can facilitate the calculation of the right search cost to which the market designer should strive.

Acknowledgment

This work was supported in part by the Israeli Science Foundation grant no. 1401/09 and BSF grant 2008-404.

References

1. A. Atakan. Assortative matching with explicit search costs. *Econometrica*, 74(3):667–680, 2006.

2. M. Attane, I. Garcia, J. Martinez, and J. Papi. Tecapsy trans european car-pooling and parking systems. In C. A. Brebbia, editor, *The Sustainable City II: Urban Regeneration and Sustainability*. Spring Verlag, 2002.
3. Y. Bakos. Reducing buyer search costs: Implications for electronic marketplaces. *Mgmt. Science*, 42:1676–92, 1997.
4. Y. Bakos. The emerging role of electronic marketplaces on the internet. *Commun. ACM*, 41(8):35–42, 1998.
5. W. C. Beckmann M., McGuire C.B. *Studies in the economics of transportation*. Yale University Press, 1959.
6. J. Benhabib and C. Bull. Job search: The choice of intensity. *Journal of Political Economy*, 91:747–764, 1983.
7. K. Burdett and R. Wright. Two-sided search with nontransferable utility. *Review of Economic Dynamics*, 1:220–245, 1998.
8. R. Cavallo. Optimal decision-making with minimal waste: strategyproof redistribution of vcg payments. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 882–889, New York, NY, USA, 2006. ACM.
9. S. Choi and J. Liu. Optimal time-constrained trading strategies for autonomous agents. In *Proc. of MAMA'2000*, 2000.
10. K. Decker, K. P. Sycara, and M. Williamson. Middle-agents for the internet. In *IJCAI*, pages 578–583, 1997.
11. S. Gal, M. Landsberger, and B. Levykson. A compound strategy for search in the labor market. *International Economic Review*, 22(3):597–608, 1981.
12. M. C. W. Janssen and J. L. Moraga-Gonzalez. Strategic pricing, consumer search and the number of firms. *Review of Economic Studies*, 71(4):1089–1118, October 2004.
13. A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, March 2007.
14. J. Kephart and A. Greenwald. Shopbot economics. *JAAMAS*, 5(3):255–287, 2002.
15. S. Lippman and J. McCall. The economics of job search: A survey. *Economic Inquiry*, 14:155–189, 1976.
16. A. M. Masters. Wage posting in two-sided search and the minimum wage. *International Economic Review*, 40(4):809–826, 1999.
17. J. McMillan and M. Rothschild. Search. In R. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, pages 905–927. 1994.
18. P. Morgan. Search and optimal sample size. *Review of Economic Studies*, 50(4):659–675, 1983.
19. P. Morgan and R. Manning. Optimal search. *Econometrica*, 53(4):923–944, 1985.
20. M. R. Preston. Multiproduct equilibrium price dispersion. *Journal of Economic Theory*, 67(1):83–105, October 1995.
21. M. Rothschild. Searching for the lowest price when the distribution of prices is unknown. *Journal of Political Economy*, 82(4):689–711, 1974.
22. D. Sarne and S. Kraus. Agents strategies for the dual parallel search in partnership formation applications. In *Proc. of AMEC2004, LNCS 3435*, pages 158 – 172, 2004.
23. R. Shimer and L. Smith. Assortative matching and search. *Econometrica*, 68(2):343–370, 2000.
24. G. Stigler. The economics of information. *Journal of Political Economy*, 69(3):213–225, 1961.
25. J. Stiglitz. Imperfect information in the product market. In R. Schmalensee and R. Willig, editors, *Handbook of Industrial Organization*, volume 1, chapter 13, pages 769–847. Elsevier, 1 edition, 1989.
26. M. L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3):641–54, May 1979.

Flexibly Priced Options: A New Mechanism for Sequential Auctions with Complementary Goods

Valentin Robu, Ioannis A. Vetsikas, Enrico H. Gerding, and Nicholas R. Jennings

University of Southampton
School of Electronics and Computer Science
Highfield Campus, Southampton, UK
{vr2, iv, eg, nrj}@ecs.soton.ac.uk

Abstract. In this paper we present a novel option pricing mechanism for reducing the exposure problem encountered by bidders with complementary valuations when participating in sequential, second-price auction markets. Existing option pricing models have two main drawbacks: they either apply a fixed exercise price, which may deter bidders with low valuations, thereby decreasing allocative efficiency, or options are offered for free, in which case bidders are less likely to exercise them, thereby reducing seller revenues. Our novel mechanism with *flexibly priced options* addresses these problems by calculating the exercise price as well as the option price based on the bids in an auction. For this novel setting we derive the optimal strategies for a bidding agent with complementary preferences. Furthermore, to compare our approach to existing ones, we derive, for the first time, the bidding strategies for a fixed price mechanism, in which exercise prices for options are fixed by the seller. Finally, we use these strategies to empirically evaluate the proposed option mechanism and compare it to existing ones, both in terms of the seller revenue and the social welfare. We show that our new mechanism achieves higher market efficiency, while still ensuring higher revenues for the seller than direct sale auctions (without options).

1 Introduction

Auctions are an efficient method for allocating resources or tasks between self-interested agents and, as a result, have been an important area of research in the multi-agent community. In recent years, research has focused on complex auctions where agents have combinatorial preferences and are interested in purchasing bundles of resources. Most of the solutions designed to address this problem involve one-shot, combinatorial auctions, where all parties declare their preferences to a center, which then computes the optimal allocation and corresponding payments [2]. Although such auctions have many desirable properties, such as incentive compatibility or efficiency of the allocation, in practice many settings are inherently decentralized and sequential. Often, the resources to be allocated are offered by different sellers, sometimes in different markets, or in auctions with different closing times. In such settings, however, a buyer of a bundle is faced with the so-called *exposure* problem when it has purchased a number of items from the bundle, but is unable to obtain the remaining items. In order to address this important problem, this paper proposes a novel option pricing mechanism, and shows its superiority over existing approaches.

In detail, sequential auctions form an important part of many application settings in which agents could be realistically applied, and such settings cannot usually be mapped to one-shot, combinatorial mechanisms. Examples include inter-related items sold on eBay by different sellers in auctions with different closing times [6], or decentralised transportation logistics, in which part-capacity loads are offered throughout a day by different shippers [10]. In these examples, a buyer is often faced with complementary preferences, i.e., where bundles of resources have more value than the sum of individual resources through synergies between these goods. In sequential auctions, this can result in the *exposure problem*. This occurs whenever an agent is faced with placing a bid for an item which is higher than its immediate marginal value, in the expectation of obtaining extra value through a synergy with another item sold later. However, if she fails to get the other item for a profitable price, she risks making a loss. In this paper, we refer to such a bidder as a *synergy bidder*. Due to the exposure problem, synergy bidders will shade their bids when entering a sequential auction market, bidding less than their bundle valuation for the items they desire [1, 3]. This reduces not only their own expected profits, but also seller revenues and the allocative efficiency of the market.

Although the exposure problem is well known, it has mostly been studied from the perspective of designing efficient bidding strategies that would help agents act in such market settings (e.g. [1, 3, 9, 11]). In this paper, we consider a different approach, that preserves the sequential nature of the allocation problem, and we propose a novel mechanism which involves auctioning *options* for the goods, instead of the goods themselves. An option is essentially a contract between the buyer and the seller of a good, where (1) the writer or seller of the option has the *obligation* to sell the good for the *exercise price*, but not the right, and (2) the buyer of the option has the *right* to buy the good for a pre-agreed *exercise price*, but not the obligation. Since the buyer gains the right to choose in the future whether or not she wants to buy the good, she pays for this right through an *option price* which she has to pay in advance, regardless of whether she chooses to exercise the option or not.

Options can help a synergy buyer reduce the exposure problem she faces since, even though she has to pay the option price, if she fails to complete her desired bundle, she does not have to pay the exercise price as well and thereby is able to limit her loss. Part of the uncertainty of not winning subsequent auctions is transferred to the seller, who may now miss out on the exercise price if the buyer fails to acquire the desired bundle. At the same time, the seller can also benefit indirectly from the participation in the market by additional synergy buyers, who would otherwise have stayed away, due to the risk of exposure to a potential loss.

1.1 Related Work

Options have a long history of research in finance (see [4] for an overview). However, the underlying assumption for all financial option pricing models is their dependence on an underlying asset, which has a known, public value that moves independently of the actions of individual agents This type of assumption does not hold for the online, sequential auctions setting we consider, as each individual synergy buyer has its own, private value for the goods/bundles on offer.

When applying options to the problem of reducing the exposure problem in sequential auctions, previous literature considers two main types of pricing mechanisms. One approach, proposed by Juda & Parkes [5, 6]) is to offer options for free (i.e., without an option price or an advance payment), and then let the exercise price be determined by the submitted bids in the market. However, this approach enables self-interested agents to hoard those options, even if they are highly unlikely to exercise them, thus considerably reducing both the allocative efficiency of the market and seller revenue. The second main approach (proposed by Mous et. al. [8]) is to have a fixed exercise price set by the seller, and then have the market determine the option price through an open auction. In this case, however, this preset exercise price can be perceived as a reserve value, since no bidder with a valuation below that price has an incentive to participate. This negatively effects the market efficiency, and may also affect the seller's profits by excluding some bidders from the market.

1.2 Contributions

To address the shortcomings of existing option models, in this paper we introduce a novel pricing mechanism in which the exercise price, as well as the option price are determined by the open market, and we compare this model to existing options models. In more detail, we extend the state-of-the-art in the following ways:

- To compare our new approach with existing ones, we derive, for the first time, the optimal bidding strategy for a synergy bidder in an options model with fixed exercise price (Section 3).
- We introduce a novel option pricing mechanism where the exercise price, as well as the options price are determined by the bids in an auction, and we show that both direct auctions (without options), and offering free options to the bidders appear in this model as particular subcases. Furthermore, we derive the optimal bidding strategy for the synergy bidder in this new model (Section 4).
- We empirically compare our new pricing model to existing option models from the literature, as well as to using direct auctions. We show that our flexible options approach achieves much better market allocation efficiency (measured in terms of the social welfare of all participating agents) than the state of the art fixed price options model. Furthermore, we show that sellers do not stand to lose any revenue by using this option model, by comparison to auctioning their items directly, without using options (Section 5).

The remainder of the paper is structured as follows. Section 2 formally defines our sequential auction setting, while Section 3 presents the bidding solution for the option model using fixed exercise price options. Section 4 introduces our new, flexible options approach, together with the optimal bidding strategies for this case. Section 5 empirically compares the two approaches, and Section 6 concludes.

2 The Problem Setting

In this section we formally describe the auction setting and introduce the notation used. We consider a setting with m second-price, sealed-bid auctions, each selling an option to buy a single item. We choose these auctions because bidders without synergies have a simple, dominant bidding strategy and, furthermore, they are strategically equivalent to the widely-used English auction. We assume that there exists a single *synergy* bidder who is interested in purchasing all of the items and receives a value of v if it succeeds, and 0 otherwise. Furthermore, every auction $j \in \{1, \dots, m\}$ has N_j *local* bidders. These bidders only participate in their own auction, and are only interested in acquiring a single item. The values of this item for local bidders in auction j are i.i.d. drawn from a cumulative distribution function F_j. Finally, we assume that all bidders are expected utility maximisers.

Given this setting, we are interested in finding the *Bayes-Nash* equilibrium strategies for all of the bidders for different option pricing mechanisms.[1] However, even with options, due to the second-price auction, the local bidders have a dominant bidding strategy. Therefore, the main problem is finding the optimal strategy for the synergy bidder and this is largely decision-theoretic in nature.

We furthermore note that, although we focus largely on a single participating synergy bidder when presenting the strategies and results, this analysis can be easily extended to multiple synergy bidders. This is because synergy bidders are assumed to only be interested in either winning all of the auctions or none at all, and therefore, after the first auction all but one synergy bidder (with the highest valuation) will leave the market. Therefore, having multiple synergy bidders only affects the bid distribution in the first auction. We address this setting in more detail in Section 4.1.

3 Optimal Bidding Strategies for Fixed Exercise Price Options

To compare our new approach with existing option pricing mechanisms, we first derive the optimal bidding strategies for a synergy bidder in a fixed exercise price setting, where the exercise price for the options to be acquired is set by the seller before the start of the auctions.[2] While the different exercise prices for the auctions are fixed in advance, the option prices are determined by the second-highest bid in the auction. In the following, we let $\overrightarrow{K}$ denote the vector of fixed exercise prices, where K_j is the exercise price of the j^{th} auction, i.e. the price that the winner will have to pay in order to purchase the item in question. Note that, if $K_j = 0$, this is equivalent to a *direct sale* auction, i.e., without any options. Furthermore, note that local bidders have a dominant strategy to bid their value minus the exercise price if this is positive, and zero otherwise. We denote by $b_1^* \dots b_m^*$ denote the optimal bids of the synergy bidder in the m auctions, and by $p_1 \dots p_m$ the prices paid in these auctions. The expiry time for the options is set after the auctions for all m items close. The following theorem then specifies the optimal bidding strategy of the synergy bidder:

[1] The Bayes-Nash equilibrium is the standard solution concept used in game theory to analyze games with imperfect information, such as auctions.

[2] This option protocol is similar to the one proposed by Mous et al. [8], but that work relies on using a heuristic bidding strategies, and they do not derive analytical expressions for the equilibrium strategies w.r.t. the local bidders.

Theorem 1. *Consider the setting from Section 2, with a pre-specified exercise price vector $\overrightarrow{K}$. If $v \leq \sum_{j=1}^{m} K_j$, then $b_r^* = 0, r \in \{1, \ldots, m\}$ constitutes a Bayes-Nash equilibrium for the synergy bidder. Otherwise, the equilibrium is given by:*

$$b_r^* = \begin{cases} v - \sum_{j=1}^{m} K_j, & \text{if } r = m \\ \int_{K_{r+1}}^{b_{r+1}^* + K_{r+1}} H(\omega) d\omega, & \text{if } 1 \leq r < m \end{cases} \quad (1)$$

where $H_j(x) = (F_j(x))^{N_j}$.

Proof. The synergy bidder cares about the value of the highest bid among the N_j local bidders which participate in the j^{th} auction. The latter would place bids which would be truthful (see [7] for 2^{nd} price auctions), if it were not for the existence of the exercise price K_j of each auction; given that K_j is an additional cost for the winner, the actual bid placed by a non-synergy bidder is equal to his valuation minus K_j. Given that each valuation is drawn from a distribution with c.d.f. $F_j(x)$, the corresponding bid is drawn from $F_j(x + K_j)$, and therefore the c.d.f. of the highest bid among local bidders is the highest order statistic, $(F_j(x + K_j))^{N_j}$.

We now compute the bidding strategy of the synergy bidder using backward induction, starting from the last (m^{th}) auction. If the synergy bidder has not won all the auctions up to the last one, then it will bid $b_m^* = 0$, as it needs to obtain all items in order to make a profit. On the other hand, assuming that the synergy bidder has won the first round, it will make a profit equal to $v - \sum_{j=1}^{m} K_j - p_m$ if it wins the last item at a price equal to p_m. This price p_m is equal to the highest opponent bid, which is drawn from $(F_m(x + K_m))^{N_m}$ when the synergy bidder is the winner. Let $H_j(x) = (F_j(x))^{N_j}$. Then the expected utility of the synergy bidder when bidding b_m is:

$$EP_m(v, b_m, \overrightarrow{K}) = \left(v - \sum_{j=1}^{m} K_j \right) H_m(K_m) + \int_0^{b_m} \left(v - \sum_{j=1}^{m} K_j - \omega \right) H_m^{'}(\omega + K_m) d\omega \quad (2)$$

The bid which maximizes this utility is found by setting:

$$\frac{dEP_m(v, b_m, \overrightarrow{K})}{db_m} = 0 \Leftrightarrow v - \sum_{j=1}^{m} K_j - b_m = 0,$$

which gives Equation 1, for the case of $r = m$. We can furthermore compute the optimal expected utility of the synergy bidder in this round by using Equation 2.

Now, we can compute the bid b_r placed in auction round r, assuming that the bid and expected utility for the next ($r + 1$) round has been computed. The synergy bidder will make expected profit equal to $EP_{r+1}^*(v, \overrightarrow{K}) - p_r$ if it wins the r^{th} item at a price equal to p_r and it has won all auctions up to that point, where $EP_j^*(v, \overrightarrow{K})$ denotes the expected profit of the synergy bidder by bidding optimally from round j onwards. The price p_r is equal to the highest opponent bid, which is again drawn from $(F_r(x+K_r))^{N_r}$

when the synergy bidder is the winner. The expected utility of the synergy bidder from bidding b_r is thus:

$$EP_r(v, b_r, \overrightarrow{K}) = EP^*_{r+1}(v, \overrightarrow{K})H_r(K_r) + \int_0^{b_r} \left(EP^*_{r+1}(v, \overrightarrow{K}) - \omega\right) H'_r(\omega + K_r)d\omega \tag{3}$$

The bid which maximizes this utility is found by setting:

$$\frac{dEP_r(v, b_r, \overrightarrow{K})}{db_r} = 0 \Leftrightarrow EP^*_{r+1}(v, \overrightarrow{K}) - b_r = 0. \tag{4}$$

Now, we need to compute the optimal expected profit EP^*_r. The expected utility of the synergy bidder when bidding b_r is given by Equation 3. Replacing the solution $b^*_r = EP^*_{r+1}(v, \overrightarrow{K})$ from Equation 4, this gives the optimal utility:

$$EP^*_r(v, \overrightarrow{K}) = b^*_r H_r(K_r) + \int_0^{b^*_r} (b^*_r - \omega)H'_r(\omega + K_r)d\omega \tag{5}$$

We can then substitute the subscripts r in Equation 5 by $r + 1$, and then since $EP^*_{r+1}(v, \overrightarrow{K}) = b^*_r$ we get:

$$b^*_r = b^*_{r+1}H_{r+1}(K_{r+1}) + \int_0^{b^*_{r+1}} (b^*_{r+1} - \omega)H'_{r+1}(\omega + K_{r+1})d\omega$$

Which, after integration by parts and substitution gives Equation 1:

$$b^*_r = \int_{K_{r+1}}^{b^*_{r+1}+K_{r+1}} H(\omega)d\omega$$

$\square$

4 Optimal Bidding Strategies for Flexibly Priced Options

In the fixed exercise price options model from the previous section, the existence of the exercise prices created a secondary effect similar to having a reserve price in the auction.[3] This is because any bidder with a private valuation lower than K_j will not participate in the auction and the same will happen if the synergy bidder has a valuation lower than the sum of the exercise prices.Although this reduces the exposure problem of the synergy bidder, at the same time it may significantly reduce the market efficiency, and also negatively effects seller revenue if this value is set too high.

In order to remove this effect, in this section we introduce a novel model with flexibly priced options, i.e., that have a flexible exercise price. In more detail, in this model, we set a *maximum* exercise price K^H_j for the auction, but the actual exercise price K_j depends on the bids placed by the bidders so as to eliminate the reserve price effect. Specifically:

[3] Note, however, that there is a subtle but important difference between having a reserve price and selling options with a fixed exercise price; whereas the two auctions are equivalent from the perspective of a local bidder, the same is clearly not true for a synergy bidder.

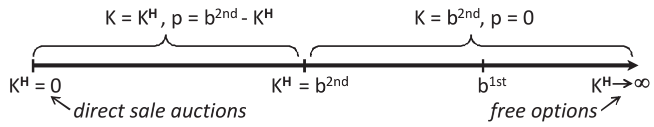


Fig. 1. The relationship between the maximum exercise price parameter, K^H, and the second-highest bid, b^{2nd}, and its effect in determining the option price, p, and actual exercise price, K, for the mechanism with flexibly priced options.

Definition 1. ***Flexible Exercise Price Options Mechanism*** *Let each seller in a sequence of m second-price auctions select a parameter K_j^H, which is the maximum exercise price she is willing to offer for the item sold in auction j. Let b_j^{2nd} denote the second highest bid placed in this auction by the participating bidders. Then, the actual exercise price of the auction is given by:*

$$K_j = \min\{K_j^H, b_j^{2nd}\}$$

Furthermore, the price paid by the winning bidder in order to purchase the option is set to:

$$p_j = b_j^{2nd} - K_j$$

Figure 1 illustrates the features of this mechanism and how it compares to some existing approaches. As shown, depending on the values of K_j^H and b^{2nd}, one of two situations can occur. Either $K_j^H < b^{2nd}$, in which case the actual exercise price is set to K_j^H, and the winner pays $b^{2nd} - K_j^H$. Otherwise, if $K_j^H \geq b^{2nd}$, then the actual exercise price is set to the second highest bid and the option is given to the winning bidder for free. In both cases, however, the total payment of the winner (if she decides to exercise the purchased option) will be equal the second highest bid. Crucially, this means that, unlike the option mechanism with fixed exercise price, from a local bidder's perspective, this auction is identical to a regular second-price auction, and there are no secondary effects on these bidders. Therefore, this options model only affects bidders with synergies.

Moreover, note from Figure 1 that this approach is a generalization of two other auction mechanisms. If the seller sets $K_j^H = 0$, then the auction becomes identical to a direct sales auction (without options). Furthermore, if K_j^H is set at a sufficiently high value (i.e. as $K_j^H \to \infty$), then the exercise price is always equal to the second highest bid, and the option is always purchased for free.

We now proceed with deriving the optimal bidding strategy for a synergy bidder.

Theorem 2. *Consider the setting from Section 2, using auctions with flexibly priced options with pre-specified maximal exercise prices K_r^H, for $r \in \{1..m\}$. The following bids b_r^* constitute a Bayes-Nash equilibrium for the synergy bidder:*

$$b_r^* = v - \sum_{i=1}^{r-1} K_i \text{ if } v \leq \sum_{i=1}^{r-1} K_i + K_r^H \tag{6}$$

$$b_r^* = K_r^H + EP_{r+1}^*(v, \overrightarrow{K}_r^H) \textit{ if } v > \sum_{i=1}^{r-1} K_i + K_r^H \tag{7}$$

$$b_m^* = v - \sum_{i=1}^{m-1} K_i \tag{8}$$

with the expected profit EP_r^* when bidding b_r^* being:

$$EP_r^*(v, \overrightarrow{K}_r^H) = \int_0^{v-\sum_{i=1}^{r-1} K_i} EP_{r+1}^*(v, \overrightarrow{K}_r^H(\omega))H_r^{'}(\omega)d\omega \tag{9}$$

if $v \leq \sum_{i=1}^{r-1} K_i + K_r^H$, otherwise it is:

$$EP_r^*(v, \overrightarrow{K}_r^H) = \int_0^{K_r^H} EP_{r+1}^*(v, \overrightarrow{K}_r^H(\omega))H_r^{'}(\omega)d\omega + \tag{10}$$

$$\int_{K_r^H}^{K_r^H + EP_{r+1}^*\left(v, \overrightarrow{K}_r^H\right)} \left(EP_{r+1}^*(v, \overrightarrow{K}_r^H) - \omega + K_r^H\right) H_r^{'}(\omega)d\omega$$

for $r \in \{1, \ldots, m-1\}$, and

$$EP_m^*(v, \overrightarrow{K}_m^H) = \int_0^{v-\sum_{i=1}^{m-1} K_i} \left(v - \sum_{i=1}^{m-1} K_i - \omega\right) H_m^{'}(\omega)d\omega \tag{11}$$

where $H_r(x) = (F_r(x))^{N_r}$ and K_r are the exercise prices for the options that have been purchased in the previous auctions. The vector $\overrightarrow{K}_r^H$ is defined as:
$\overrightarrow{K}_r^H = \{K_1, \ldots, K_{r-1}, K_r^H, \ldots, K_{m-1}^H\}$, for all rounds $r = 1, \ldots, m$.
Furthermore $\overrightarrow{K}_r^H(x)$ is defined as the vector $\overrightarrow{K}_r^H$ where the element K_r^H is replaced by value x, thus $\overrightarrow{K}_r^H(x) = \{K_1, \ldots, K_{r-1}, x, K_{r+1}^H, \ldots, K_{m-1}^H\}$.

Proof. The synergy bidder cares about the value of the highest bid among the N_r non-synergy bidders which participate in the r^{th} auction. The latter place truthful bids (as we discussed). Given that each of their valuations is drawn from a distribution with cdf $F_r(x)$, therefore the cdf of the highest bid among the local, one-item bidders is: $H_r(x) = (F_r(x))^{N_r}$, as it is the maximum of N_r random variables drawn from $F_r(x)$.

To compute the bidding strategy of the synergy bidder, we start from the last (m^{th}) auction. If the synergy bidder has not won all the previous auctions, then it will bid $b_m = 0$, as it needs to obtain all items in order to make a profit. If it has won all these auctions and the exercise prices for the options purchased are K_r, then it will make a profit equal to $v - \sum_{i=1}^{m-1} K_i - \omega$ if it wins the last item when the highest opponents bid is equal to ω.[4] Note that whether $\omega > K_m^H$ or the other way round, in both cases

[4] Note that it should be $v \geq \sum_{i=1}^{m-1} K_i$ when the synergy bidder has won all the previous auctions and the exercise prices are K_i. This follows from the fact that the bidder will not bid more than $v - \sum_{i=1}^{j-1} K_i$ in any auction j and the exercise price cannot be higher than his bid if he has won.

the winning synergy bidder makes a total payment equal to ω, where ω is drawn from $H_m(x)$ when the synergy bidder is the winner. We compute the expected utility of the synergy bidder when he bids b_m as:

$$EP_m(v, b_m, \overrightarrow{K}^H_m) = \int_0^{b_m} \left(v - \sum_{i=1}^{m-1} K_i - \omega \right) H^{'}_m(\omega) d\omega$$

The bid which maximizes this utility is found by setting:

$$\frac{dEP_m(v, b_m, \overrightarrow{K}^H_m)}{db_m} = 0 \Leftrightarrow v - \sum_{i=1}^{m-1} K_i - b^*_m = 0,$$

which gives Equation 8.

The expected profit of synergy bidder when bidding b^*_m is therefore: $EP^*_m(v, \overrightarrow{K}^H_m) = EP_m(v, b^*_m, \overrightarrow{K}^H_m)$, which gives Equation 11, when substituting b_m from Equation 8.

Assume that we have computed the bids b^*_j and expected profit EP^*_j for $\forall j > r$ and that these are given by the equations of this theorem. To complete the proof, we will now show how to compute the bid and the expected utility for the r^{th} auction (round). Depending on whether $b_r \leq K^H_r$ or not, we need to distinguish two different cases:

Case 1: $b_r \leq K^H_r$. In this case, $\forall \omega \leq b_r \Rightarrow \omega \leq K^H_r$. Therefore if the synergy bidder wins with a bid of b_r, then the second bid in the auction (which is the highest opponent bid) must be smaller than K^H_r and therefore the exercise price is equal to this bid and it is given for free. Hence, the expected profit of the synergy bidder is:

$$EP_r(v, b_r, \overrightarrow{K}^H_r) = \int_0^{b_r} EP^*_{r+1}(v, \overrightarrow{K}^H_r(\omega)) H^{'}_r(\omega) d\omega$$

To find the bid b^*_r that maximizes this expected utility we use Lagrange multipliers. The inequality is rewritten as $b_r - K^H_r + \delta^2 = 0$, and the Lagrange equation for this problem becomes:

$$\Lambda(b_r, \lambda, \delta) = -\int_0^{b_r} EP^*_{r+1}(v, \overrightarrow{K}^H_r(\omega)) H^{'}_r(\omega) d\omega + \lambda(b_r - K^H_r + \delta^2)$$

The possible variables which maximize this function are found by setting the partial derivatives for dependent variables b_r, λ, δ to 0:

$$\frac{\vartheta \Lambda(b_r, \lambda, \delta)}{\vartheta b_r} = 0 \Leftrightarrow EP^*_{r+1}(v, \overrightarrow{K}^H_r(b_r)) H^{'}_r(b_r) = \lambda$$
$$\frac{\vartheta \Lambda(b_r, \lambda, \delta)}{\vartheta \lambda} = 0 \Leftrightarrow b_r - K^H_r + \delta^2 = 0$$
$$\frac{\vartheta \Lambda(b_r, \lambda, \delta)}{\vartheta \delta} = 0 \Leftrightarrow \lambda\delta = 0$$

The last equation can mean that either: (i) $\delta = 0$, thus $b^*_r = K^H_r$, or (ii) $\lambda = 0$, and by substituting into the first equation, we get $EP^*_{r+1}(v, \overrightarrow{K}^H_r(b_r)) H^{'}_r(b_r) = 0$, thus $EP^*_{r+1}(v, \overrightarrow{K}^H_r(b_r)) = 0$. Here, the expected optimal utility EP^*_{r+1} is given either by

Equation 9 or 10, depending on whether the valuation discounted by the exercise prices up to that point $v - \sum_{i=1}^{i=r} K_i$ is less or greater (respectively) than K_{r+1}^H. The second case cannot occur if $EP_{r+1}(v, \overrightarrow{K}_r^H(b_r)) = 0$, because the first integral of Equation 10 is greater than 0, unless $b_r^* = 0$, which would yield an expected utility of 0 and thus cannot be the optimal bid (in general). On the other hand, if the optimal expected utility EP_{r+1}^* is given by Equation 9, then $EP_{r+1}^*(v, \overrightarrow{K}_r^H(b_r)) = 0$ exactly when the upper bound of the integral is 0, i.e. $v - \sum_{i=1}^{r-1} K_i - b_r = 0$, which gives Equation 6.

Note that of the two possible maxima $b_r^* = v - \sum_{i=1}^{r-1} K_i$ and $b_r^* = K_r^H$, given by this analysis, the first one yields a higher revenue, as $EP_{r+1}^*(v, \overrightarrow{K}_r^H(\omega))H_r^{'}(\omega) < 0$, $\forall \omega > v - \sum_{i=1}^{r-1} K_i$ in this case. This means that the optimal bid is $b_r^* = v - \sum_{i=1}^{r-1} K_i$. By adding the case condition $b_r^* \le K_r^H \Leftrightarrow v - \sum_{i=1}^{r-1} K_i \le K_r^H$, we obtain the bound in the theorem.

Case 2: $b_r > K_r^H$. In this case, if the synergy bidder wins with a bid of b_r, then the second bid in the auction could be smaller than K_r^H and therefore the exercise price is equal to this bid and it is given for free, like in the previous case. However, it could also be higher than K_r^H, and thus the exercise price is equal to K_r^H, whereas the payment for getting the options would be equal to the second bid minus K_r^H. Hence, the expected profit of the synergy bidder is:

$$EP_r(v, b_r, \overrightarrow{K}_r^H) = \int_0^{K_r^H} EP_{r+1}^*(v, \overrightarrow{K}_r^H(\omega))H_r^{'}(\omega)d\omega$$
$$+ \int_{K_r^H}^{b_r} (EP_{r+1}^*(v, \overrightarrow{K}_r^H) - \omega + K_r^H)H_r^{'}(\omega)d\omega$$

The optimal bid b_r^* that maximizes this expected utility is derived in the same way as in Case 1, by applying Lagrange multipliers to the above equation. We skip this part of the proof due to space constraints. □

4.1 Multiple Synergy Bidders

We finish the theoretical analysis by showing that in all the theorems that we presented the bidding strategies would remain unchanged, even if multiple (n) synergy bidders participate. We prove this for the flexible auction model from Theorem 2, but the same argument applies to the fixed option model in Theorem 1.

Proposition 1. *A setting with n synergy bidders with a valuation v_i for synergy bidder i when it obtains m items, and 0 otherwise, is strategically equivalent (in the case of a sequence of m auctions with flexibly priced options) to a setting where only a single synergy bidder participates and his valuation is equal to $\max_i\{v_i\}$.*

Proof. In the model considered in this paper, each synergy bidder would need to win all items in order to make a profit (i.e. there are no substitutable items available). Hence, it follows that only the synergy bidder (if any) who won the first auction would participate in the remaining ones, and the bidding strategies and the expected profit in the remaining rounds would remain unchanged. Now, in the first auction, we notice from

Equations 6 and 7 that the optimal bids of any synergy bidder in any auction (round) are not affected by the bids placed by its opponents in that particular auction, but only in the remaining ones. Therefore, the bid in the first auction is affected only by the opponent bidding in the remaining auctions, which remains unchanged as there is only one synergy bidder participating in all remaining auctions. Finally, from Equations 6 and 7 it follows that the bids placed (assuming that all the parameters K_i^L and K_i^H remain unchanged) increase as the valuation v increases, therefore the synergy bidder with the highest valuation $v = \max_i\{v_i\}$ is the only one who might win the first auction. From this, it follows that this scenario is thus strategically equivalent to one where this bidder with the highest valuation is the only synergy bidder participating.

5 Empirical Analysis

In this section, we evaluate and compare experimentally the fixed and flexible exercise price option models discussed above, as well as the direct auctions case which, as discussed above, appears as a special case in both models when $K = K_H = 0$.

5.1 Experimental Setup

The settings used for our experiments are as follows. In each run, we simulate a market consisting of $m = 3$ sequential auctions. Each auction involves $N = 5$ local bidders, and one synergy bidder. The valuations of the local bidders are i.i.d. drawn from normal distributions $\mathcal{N}(\mu = 2, \sigma = 4)$. This high variance makes the distribution almost flat (i.e. close to uniform). This uncertain valuation setting makes options more desirable.

The valuation for the synergy bidder, v_{syn}, is drawn the normal distribution $\mathcal{N}(\mu_{syn} = 20, \sigma = 2)$. We choose this setting as it demonstrates the effect of the exposure problem; if the value of the synergy bidder is set too high, then the it would win all of the auctions, even in the case of direct sale. On the other hand, if the value is set too low, then the exposure problem disappears since local bidders will win all of the auctions. Here the value of the synergy bidder is in between these extremes and is representative of a setting in which the exposure problem plays an important role.

For this setting, we compare the allocative efficiency as well as the seller revenue of the fixed and flexible exercise priced options mechanism, and for different values of the auction parameters (the exercise prices K_i in the first model, and maximum exercise prices K_i^H in the second). Formally, the allocative efficiency is calculated as follows. Let v_i^k denote the valuation of local bidder i (where $i \in \{1..N\}$) in the k^{th} auction (where $k \in \{1..m\}$) and let v_{syn} be the valuation of the synergy bidder. Furthermore, let $x_i^k, x_{syn} \in \{0,1\}$ denote the actual allocation of the options in a certain run of the simulation. That is, $x_i^1 = 1$ means that local bidder i acquired the option in the $1st$ auction, and $x_{syn} = 1$ means that the synergy bidder won *all* the auctions. Given this, the allocative efficiency, η, of the entire market in a given run is defined as:

$$\eta = \frac{\sum_{i=1}^{n}\sum_{k=1}^{m} x_i^k v_i^k + x_{syn} v_{syn}}{\max\left(v_{syn}, \sum_{k=1}^{m} \max_{i\in\{1,\dots,N\}}(v_i^k)\right)} \quad (12)$$

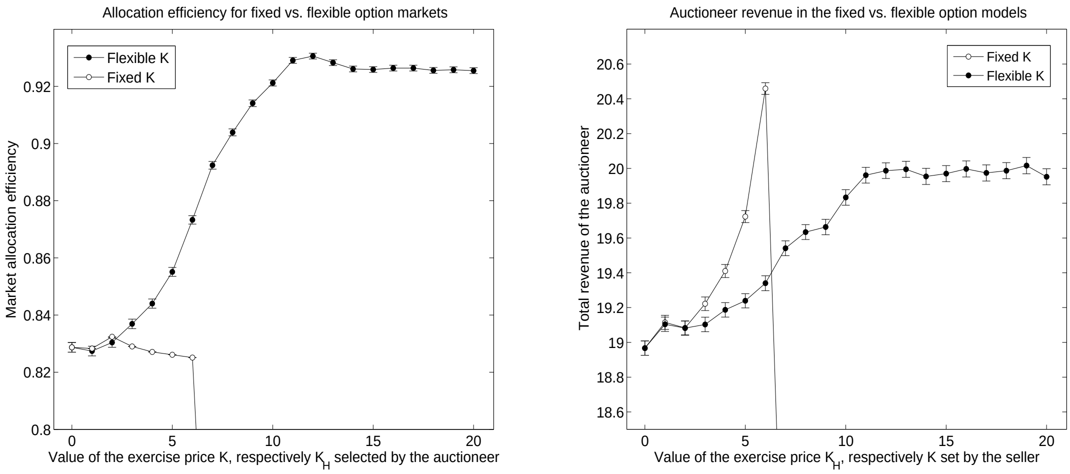


Fig. 2. Allocative efficiency (left) and seller revenue (right) using the options mechanism with fixed and flexible exercise price, and with identical parameters K and K^H respectively in all auctions. Result are averaged over 3500 runs. The error bars indicate the standard error.

By calculating the efficiency of the market in this way, we implicitly assume that local bidders will always exercise their options, and that the synergy bidder will exercise its option if and only if it wins all auctions. We can safely make this assumption because we consider optimal bidding strategies, and a rational bidder will never place a bid such that the combined exercise and option price will exceed the (marginal) value of the item. Therefore, it is optimal for a bidder who has acquired options for all of its desired items to exercise them. Thus an *in*efficient outcome occurs in two situations. Either the local bidders have won the items, but the value of the synergy bidder exceeds the sum of the values of the local bidders; or, the synergy bidder has won some auctions but not all, and will therefore not exercise its option(s).

5.2 Discussion of Numerical Results

To reduce the number of parameters, we consider a setting in which the parameters, K_i and K_i^H for the two options mechanism respectively, are set to the same value in all of the 3 sequential auctions, denoted by K and K^H respectively in the following. This models a setting in which the seller has to pre-specify the protocol for selling her item without any knowledge about a synergy bidder's endowment state[5] (i.e., how many options it has won so far).

To this end, Figure 2 compares the allocation efficiency of the market (left), and the seller revenues (right) for the two option mechanisms. Note that the direct auctions case appears, in both option models, as a particular subcase, for $K = K_H = 0$.

[5] This is especially relevant in a large open environment where synergy bidders can enter and exit the market dynamically.

Figure 2 shows that, for the *fixed exercise price* option model, both the efficiency and the seller revenue start to decrease sharply when K becomes larger than around 6.2. This is because, at this point, the synergy bidder is likely to leave the market due to the reservation price effect of this mechanism. Specifically, this occurs when the sum of the exercise prices of the auctions exceeds the valuation of the synergy bidder, in which case the bidder no longer has an incentive to participate. This also holds for many of the local bidders. This is precisely the outcome that we would like to avoid using our flexibly priced option mechanism. Note that, using our new mechanism, having a very high value for K^H has a very different effect, and the options become effectively free. This is because, when K_H is very high, it will almost certainly exceed the second-highest bid. If this happens, the exercise price becomes equal to this bid, and the option price becomes zero (see also Section 4).

As is shown in Figure 2 (left), since the flexibly priced option mechanism removes the reserve price effect, this mechanism outperforms a fixed exercise price in terms of efficiency. Furthermore, both mechanisms outperform direct auctions (i.e. without options, which is when $K = 0$ or $K^H = 0$) for an appropriately set parameter. The results also show that having free options is suboptimal in terms of efficiency. This is because there is a small chance that the synergy bidder will win the one or more auctions in the sequence, but loose the second or the third, and hence not exercise her options. If this occurs, some goods remain unallocated. This gives rise to inefficiency since the goods could have been allocated to a local bidder instead.

While the flexibly priced options outperforms other mechanisms in terms of efficiency, the same cannot be said for seller revenue. In this context, Figure 2 (right) shows that a seller can achieve significantly higher revenues by using a fixed exercise price. This is not entirely surprising, since the fixed exercise price has a secondary effect which is similar to setting a reserve price and standard auction theory shows that, even in a single second-price auction, the seller can increase its revenues by using reserve prices [7]. Nevertheless, we find that the flexibly priced option mechanism achieves a higher revenue than regular, direct sale auctions. Furthermore, it is important to point out that, if the aim is to maximise revenue, the flexibly priced option mechanism can be used in combination with a reserve price, and our mechanism enables the separation of the two effects: reducing the exposure problem of synergy bidders and increasing seller revenue. We intend to investigate models in which both of these effects are jointly captured, but through separate parameters, in future work.

6 Conclusions

The exposure problem faced by bidders with valuation synergies in sequential auctions is a difficult, but an important one, with considerable implications for both theory and practice, for a wide range of multi-agent systems. Due to the risk of not acquiring all the desired items in future auctions, bidders with valuation synergies often shade their bids, or do not participate in such markets, which considerably reduces both allocation efficiency and auctioneer revenue. Options have been identified before [5, 6, 8] as a promising solution to address this problem, but existing mechanisms in the literature either prescribe free options, or options in which the seller fixes a minimum exercise price (which can potentially deter many bidders from entering the market).

To this end, in this paper, we propose a novel option mechanism, in which the exercise price is set flexibly, as a minimum between the second highest bid and a seller-prescribed maximum level, while the option price is determined by the open market. We derive the optimal bidding policies of the synergy bidder in this new model and show that this mechanism can significantly increase the social welfare of the resulting allocations, while at the same time outperform sequential auction with direct sales in terms of seller revenues.

While the mechanism proposed in this paper makes a significant step forward in addressing this problem, several aspects are still left open to future work. One of these is combining the two options models by allowing sellers to fix a minimum exercise price for their options, as well as a maximum. Such a mechanism would, on the one hand, be able to reduce the exposure problem for synergy bidders, and therefore result in a corresponding increase in market efficiency, but would also allow the seller to extract more profits, as they do in a fixed-price options model. Another important area where further work is needed, is the derivation of optimal bidding strategies in more general market settings, such as those involving substitutable items as well as complementary ones.

References

1. C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resources with complementarities. *Proc. of IJCAI'99*, pp. 527–523, Morgan Kaufmann, 1999.
2. P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial auctions*. MIT Press, 2006.
3. A. Greenwald and J. Boyan. Bidding under uncertainty: theory and experiments. In *UAI '04: Proc. of the 20th Conference on Uncertainty in AI*, pages 209–216, 2004.
4. J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, fifth edition, 2003.
5. A. I. Juda and D. C. Parkes. An options-based method to solve the composability problem in sequential auctions. *Agent-Mediated El. Comm. VI*, pp. 44–58. Springer LNCS 3435, 2006.
6. A. I. Juda and D. C. Parkes. An Options-Based Solution to the Sequential Auction Problem. In *Artificial Intelligence*, vol. 173(7-8), pages 876–899, 2009.
7. V. Krishna. *Auction theory*. Academic Press, 2002.
8. L. Mous, V. Robu, and J. A. La Poutré. Using priced options to solve the exposure problem in sequential auctions. In *Agent-Mediated El. Commerce (AMEC'08)*, Springer LNBIP, 2009.
9. A. Osepayshvili, M. Wellman, D. Reeves, J. MacKie-Mason. Self-confirming price prediction for bidding in simultaneous ascending auctions. In *Proc. of UAI'05*, pp. 441–449, 2005.
10. V. Robu, H. Noot, J. A. La Poutré, and W.-J. van Schijndel. An agent platform for auction-based allocation of loads in transportation logistics. In *Proc. of AAMAS'08, Industry Track*, pages 3–10, 2008.
11. I. A. Vetsikas and N. R. Jennings. Bidding strategies for realistic multi-unit sealed-bid auctions. In *23rd Conference on Artificial Intelligence (AAAI'08), Chicago*, 2008.

Time Constraints in Mixed Multi-unit Combinatorial Auctions

Andreas Witzel[1]* and Ulle Endriss[2]

[1] CIMS, New York University, USA
[2] ILLC, University of Amsterdam, Netherlands

Abstract. We extend the framework of mixed multi-unit combinatorial auctions to include time constraints, present an expressive bidding language, and show how to solve the winner determination problem for such auctions using integer programming. Mixed multi-unit combinatorial auctions are auctions where bidders can offer combinations of transformations of goods rather than just simple goods. This model has great potential for applications in the context of supply chain formation, which is further enhanced by the integration of time constraints. We consider different kinds of time constraints: they may be based on either time points or intervals, they may determine a relative ordering of transformations, they may relate transformations to absolute time points, and they may constrain the duration of transformations.

1 Introduction

Cerquides et al. [2] have proposed an extension of the standard combinatorial auction model, called **mixed multi-unit combinatorial auctions** (or simply **mixed auctions**). In a mixed auction, bidders can offer transformations, consisting of a set of input goods and a set of output goods, rather than just plain goods. Bidding for such a transformation means declaring that one is willing to deliver the specified output goods after having received the input goods, for the price specified by the bid. Solving a mixed auction means choosing a sequence of transformations that satisfies the constraints encoded by the bids, that produces the goods required by the auctioneer from those he holds initially, and that maximizes the amount of money collected from the bidders (or minimizes the amount paid out by the auctioneer). Mixed auctions extend several other types of combinatorial auctions: direct auctions, reverse auctions, and combinatorial exchanges. A promising application is supply chain formation.

We propose extending the framework of mixed auctions by allowing bidders to specify constraints regarding the times at which they perform the transformations offered in their bids. The motivation for this extension is that, in a complex economy, the bidders (service providers) themselves may need services from others and have their own supply chains, so the bidders may have preferences over the timing of transformations and over their relative ordering. A notion of time is already implicit in the original framework as

* Most work was done at ILLC on a GLoRiClass fellowship funded by the European Commission (Early Stage Research Training Mono-Host Fellowship MEST-CT-2005-020841).

far as the auctioneer is concerned, who builds a sequence of transformations, but this is not the case for the bidders. In this work we seek to redress this imbalance.

Our contribution covers four types of **time constraints**:

- *Relative time points:* associate each transformation with a time point and allow bidders to express constraints regarding their relative ordering, e.g., transformation X must be executed before Y.
- *Absolute time points:* additionally allow references to absolute time, e.g., execute X at time 15, or at most 3 time units after Y.
- *Intervals:* associate transformations with intervals and specify constraints, e.g., X must be executed *during* Y.
- *Intervals with absolute durations:* allow intervals with absolute time, e.g., X should take at least 5 time units.

These constraint types can be freely mixed to, for instance, express an interval taking place after a time point. Furthermore, it is possible to model *soft constraints*, allowing bidders to offer discounts in return for satisfying certain time constraints, and to model the fact that an auctioneer may sometimes be able to quantify the monetary benefit resulting from a shorter supply chain. Our approach blends nicely into the existing framework of mixed auctions, requiring surprisingly few modifications. This facilitates integration with other extensions and optimizations.

In Sect. 2, we define a suitable **bidding language**. In Sect. 3, we define the **winner determination problem** and present an integer program to solve it. Section 4 presents the extension to time intervals. Section 5 discusses related work and concludes.

2 Bidding language

In a mixed auction, agents negotiate over transformations of goods that are equipped with time point identifiers. In this section we introduce an expressive bidding language that allows bidders to specify their valuations over sets of such transformations. We also present some purely syntactic extensions to the bidding language, and we show that it is fully expressive over the class of all "reasonable" valuations.

2.1 Transformations and time points

Let G be the finite set of all types of goods considered. A **transformation** is a pair $(\mathcal{I}, \mathcal{O}) \in \mathbb{N}^G \times \mathbb{N}^G$. An agent offering such a transformation declares that, when provided with the multiset of goods $\mathcal{I}$, he can deliver the multiset of goods $\mathcal{O}$. Let $\mathcal{T}$ be a finite (but big enough) set of **time point identifiers**. These time points are merely identifiers, not variables having an actual value. They can be referred to from bids in order to specify time constraints over the offered transformations. Agents negotiate over sets of transformations with time point identifiers $\mathcal{D} \subset \mathbb{N}^G \times \mathbb{N}^G \times \mathcal{T}$, which we can write as

$$\mathcal{D} = \{(\mathcal{I}^1, \mathcal{O}^1, \tau^1), \ldots, (\mathcal{I}^\ell, \mathcal{O}^\ell, \tau^\ell)\}.$$

For example, $\{(\{\}, \{q\}, \tau_1), (\{r\}, \{s\}, \tau_2)\}$ means that an agent is able to deliver q without any input at some time τ_1, and to deliver s if provided with r at some time τ_2 (possibly with constraints regarding τ_1 and τ_2).

2.2 Valuations

A **time line** Σ (for a given bidder) is a finite sequence of transformations and "clock ticks" c (when no transformation is allocated to the bidder). That is, $\Sigma \in (\mathbb{N}^G \times \mathbb{N}^G \cup \{c\})^*$. A **valuation** v maps a time line Σ to a real number p. Intuitively, $v(\Sigma) = p$ means that an agent with valuation v is willing to make a payment of p for getting the task of performing transformations according to the time line Σ (p is usually negative, so the agent is *being paid*). We write $v(\Sigma) = \bot$ if v is *undefined* for Σ, i.e., the agent would be unable to accept the corresponding deal. For example, the valuation v given by

$$\begin{aligned} v((\{oven, dough\}, \{oven, cake\})) &= -2 \\ v((\{oven, dough\}, \{oven, cake\}); (\{\}, \{bread\})) &= -3 \\ v((\{\}, \{bread\}); (\{oven, dough\}, \{oven, cake\})) &= \bot \end{aligned}$$

expresses that for two dollars I could produce a cake if given an oven and dough, also returning the oven; for another dollar I could do the same and afterwards give you a bread without any input; but I could not do it the other way round.

A valuation v uses **relative time** if for all Σ we have that $v(\Sigma) = v(\Sigma - c)$, where $\Sigma - c$ stands for Σ with all clock ticks c removed. That is, valuations depend only on the relative ordering of the transformations. Otherwise v is said to use **absolute time**.

2.3 Bids

An **atomic bid** $\text{BID}(\mathcal{D}, p)$ specifies a finite set of finite transformations with time points and a price. For **complex bids**, we restrict ourselves to the XOR-*language*, which, for mixed auctions, fully expresses most (if not all) intuitively sensible valuations [2]. Our framework can easily be extended to also handle the OR-operator. An XOR-bid,

$$\mathit{Bid} = \text{BID}(\mathcal{D}_1, p_1) \text{ XOR} \ldots \text{XOR BID}(\mathcal{D}_n, p_n),$$

says that the bidder is willing to perform *at most one* $\mathcal{D}_j$ and pay the associated p_j.

2.4 Time constraints

The **atomic constraints** for *relative time* are of the form $\tau < \tau'$; and for *absolute time*, with $\tau, \tau' \in \mathcal{T}$, $\xi, \xi' \in \mathbb{N}$:

$$\tau = \xi \qquad \tau < \xi \qquad \tau > \xi \qquad \tau + \xi < \tau' + \xi' \qquad \tau + \xi = \tau' + \xi'$$

As mentioned above, the τ are not variables but just identifiers for the associated transformations, and the above formulas are not assignments but rather constraints on the associated transformations, with semantics as given in Sect. 2.5. For example,

$$\begin{array}{ll} \text{BID}(\{\ (\{oven, dough\}, \{oven, cake\}, \tau_1), & \\ \qquad (\{\}, \{bread\}, \tau_2)\}, -3) & \qquad \tau_1 < \tau_2 \end{array}$$

expresses the above fact that I am willing to sell you the bread only *after* the cake.

Time **constraint formulas** are of the form $\varphi = \gamma_1 \wedge \cdots \wedge \gamma_\nu$ with atomic constraints γ_ι. A bidder submits a bid *Bid* together with a time constraint formula φ, expressing that he is willing to perform according to *Bid*, but *only under the condition* that φ is satisfied. This condition is *hard*: the bidder will only accept if it is met.

2.5 Semantics

Syntactically, we thus have *complex bids with time points* together with *constraint formulas* over these time points:

$$\text{BID}(\mathcal{D}_1, p_1) \text{ XOR} \ldots \text{XOR BID}(\mathcal{D}_n, p_n) \qquad\qquad \gamma_1 \wedge \cdots \wedge \gamma_\nu.$$

In order to make the intuitive meanings explained above explicit, we now specify a formal semantics. In the following, let Σ be a time line (clock ticks allowed), let $\tau, \tau' \in \mathcal{T}$, $\xi, \xi' \in \mathbb{N}$, and let φ and φ' be time constraint formulas. Let $\tau \in \Sigma$ denote the fact that τ is associated with some transformation in Σ, and let $\Sigma(\tau)$ denote the sequence number (starting from 1) of the transformation associated with τ, if $\tau \in \Sigma$. For clarity, we may include the time point identifiers in the sequence. For example, if $\Sigma = ((\mathcal{I}^1, \mathcal{O}^1, \tau^1); \ldots)$, then $\tau^1 \in \Sigma$ and $\Sigma(\tau^1) = 1$.

We inductively define a **satisfaction relation** $\models$ as follows:

$$\begin{array}{ll}
\Sigma \models \tau \circ \xi & \text{iff } \tau \notin \Sigma \text{ or } \Sigma(\tau) \circ \xi, \text{ for } \circ \in \{=, <, >\} \\
\Sigma \models \tau + \xi < \tau' + \xi' & \text{iff } \tau' \notin \Sigma \text{ or} \\
 & \quad \tau \in \Sigma \text{ and } \Sigma(\tau) + \xi < \Sigma(\tau') + \xi' \\
\Sigma \models \varphi \wedge \varphi' & \text{iff } \Sigma \models \varphi \text{ and } \Sigma \models \varphi'
\end{array}$$

Relative time constraints are covered by omitting the $+\xi$ and $+\xi'$, and $\tau + \xi = \tau' + \xi'$ is an abbreviation for

$$\tau + \xi < \tau' + (\xi' + 1) \wedge \tau' + \xi' < \tau + (\xi + 1).$$

According to this semantics, time constraints over time point identifiers that are fully included in Σ are interpreted as expected. Constraints over time point identifiers not in Σ are simply ignored (they are always satisfied). Note that the choice of semantics for constraints such as $\tau < \tau'$ is somewhat arbitrary in case only one of the time points being compared occurs in Σ. As an intuitive justification for this detail of the semantics, τ may be thought of as a precondition for τ', for instance, because some outcome of the first transformation is needed for the second. In the case of $\tau + \xi = \tau' + \xi'$, this has the effect that either none of the two mentioned transformations is included, or both are and must have the specified distance. However, the exact details do not matter all that much, since the bidding language allows specifying in all detail which transformations can occur together and which cannot.

Using a more technical justification, we prefer this interpretation of constraints because it turns out that it has a straightforward translation to integer constraints, which we need for the implementation described in Sect. 3.3.

We say that a set of transformations $\mathcal{D}$ **permits** Σ if Σ consists of *exactly* the transformations in $\mathcal{D}$ (and optionally clock ticks). In contrast to this definition, in [2], different assumptions concerning *free disposal* are distinguished. Informally, free disposal means that participants are always happy to accept *more* goods than they strictly require; if they really have absolutely no use for them (or are even bothered by them), they can dispose of them *for free*. Free disposal makes intuitive sense for most every-day goods; however it is not as appropriate for certain "goods" like nuclear waste. We do not delve further into this issue here and continue without any free-disposal assumptions; however, we emphasize that this is purely for the sake of clarity, and these assumptions could be built in with only minuscule changes. In particular, the issue of free disposal as far as bidders are concerned has no impact on the winner determination problem discussed in Sect. 3; it only affects the definition of the semantics of the bidding language.

We now define the valuation expressed by an atomic bid $\mathit{Bid} = (\mathcal{D}, p)$ together with a time constraint formula φ as

$$v_{\mathit{Bid},\varphi}(\Sigma) = \begin{cases} p & \text{if } \mathcal{D} \text{ permits } \Sigma \text{ and } \Sigma \models \varphi \\ \bot & \text{otherwise.} \end{cases}$$

Accordingly, the valuation expressed by a complex bid $\mathit{Bid} = \textsc{xor}_{j=1}^{n} \mathit{Bid}_j$ together with a time constraint formula φ is (interpreting $\bot$ as $-\infty$):

$$v_{\mathit{Bid},\varphi}(\Sigma) = \max\{v_{\mathit{Bid}_j,\varphi}(\Sigma) \mid j \in \{1, \ldots, n\}\}.$$

That is, out of all the applicable atomic bids Bid_j (i.e., where $v_{\mathit{Bid}_j,\varphi}(\Sigma) \neq \bot$), the auctioneer is allowed to choose the one giving him maximum profit.

2.6 Syntactic extensions

The time constraint language may seem limited, allowing only conjunctions of atomic constraints. However, additional expressive power can be "borrowed" from the bidding language. We discuss two extensions to the time constraint language.

Disjunctive time constraints. A bidder may want to offer $(\mathcal{I}^1, \mathcal{O}^1)$, $(\mathcal{I}^2, \mathcal{O}^2)$ and $(\mathcal{I}^3, \mathcal{O}^3)$ for price p, where the third should take place after the second *or* the first, i.e.,

$$\textsc{bid}(\{(\mathcal{I}^1, \mathcal{O}^1, \tau^1), (\mathcal{I}^2, \mathcal{O}^2, \tau^2), (\mathcal{I}^3, \mathcal{O}^3, \tau^3)\}, p) \qquad \tau^1 < \tau^3 \vee \tau^2 < \tau^3,$$

with the obvious meaning of the disjunction $\vee$. This is not directly possible in our time constraint language. However, it can be translated into

$$\begin{array}{l} \textsc{bid}(\{(\mathcal{I}^1, \mathcal{O}^1, \vartheta^1), (\mathcal{I}^2, \mathcal{O}^2, \vartheta^2), (\mathcal{I}^3, \mathcal{O}^3, \vartheta^3)\}, p) \\ \quad \textsc{xor}\ \textsc{bid}(\{(\mathcal{I}^1, \mathcal{O}^1, \zeta^1), (\mathcal{I}^2, \mathcal{O}^2, \zeta^2), (\mathcal{I}^3, \mathcal{O}^3, \zeta^3)\}, p) \end{array} \quad \vartheta^1 < \vartheta^3 \wedge \zeta^2 < \zeta^3.$$

The choice which of the disjuncts to satisfy has been moved into the bid expression and is determined by picking one of the atomic bids. Since their variables are disjoint, this pick makes one conjunct of the transformed time constraint formula vacuously true, while the other conjunct still needs to be satisfied. Since it may happen that both of the original disjuncts are satisfied in the end, disjunction is the right notion here, even though it is translated into an XOR of bids.

For a general formulation, we allow a bid expression in XOR normal form together with a time constraint formula in disjunctive normal form:

$$\textsc{xor}_{j=1}^{n} \mathit{Bid}_j \qquad\qquad \bigvee\nolimits_{\iota=1}^{\nu} \varphi_\iota,$$

where the φ_ι are standard (conjunctive) time constraint formulas. The bidder can thus conveniently express, e.g., several alternative partial orders over his transformations.

Let now σ_ι for $\iota \in \{1, \ldots, \nu\}$ be substitutions (with disjoint ranges), each mapping all variables occurring in the bid to fresh (used nowhere else) ones. The translation is:

$$\text{XOR}_{\iota=1}^{\nu} \, \text{XOR}_{j=1}^{n} \, \mathit{Bid}_j \sigma_\iota \qquad\qquad \bigwedge_{\iota=1}^{\nu} \varphi_\iota \sigma_\iota .$$

This may seem surprising, because in the original formulation the auctioneer has two choices (which of the time constraint disjuncts to satisfy and which bid to pick), and in the translation he loses the choice among the time constraints. However, in return he gets the freedom to choose over the outer XOR. As illustrated in the example above, this boils down to choosing one of the fresh variable spaces, which corresponds to choosing one of the original disjuncts. All the rest of the transformed time conjunction does not have any effect, because it talks about variables which do not occur in the chosen sub-bid. The auctioneer then proceeds to pick a bid from the inner XOR, just as before.

Soft time constraints. Soft constraints are constraints with associated costs. Intuitively, such a constraint does not have to be satisfied, but if it is, the price of the bid is modified by the given cost (usually a discount to the auctioneer).

For example, a bidder may want to bid on $(\mathcal{I}^1, \mathcal{O}^1)$ and $(\mathcal{I}^2, \mathcal{O}^2)$ for price p and offer a discount, i.e., raise his bid by δ, if he gets to do the first before the second:

$$\text{BID}(\{(\mathcal{I}^1, \mathcal{O}^1, \tau^1), (\mathcal{I}^2, \mathcal{O}^2, \tau^2)\}, p) \qquad\qquad (\tau^1 < \tau^2, \delta).$$

Again, this expression can be translated:

$$\begin{array}{ll} \text{BID}(\{(\mathcal{I}^1, \mathcal{O}^1, \vartheta^1), (\mathcal{I}^2, \mathcal{O}^2, \vartheta^2)\}, p) & \\ \quad \text{XOR BID}(\{(\mathcal{I}^1, \mathcal{O}^1, \zeta^1), (\mathcal{I}^2, \mathcal{O}^2, \zeta^2)\}, p + \delta) & \qquad \zeta^1 < \zeta^2. \end{array}$$

The general translation is analogous to the previous one and omitted for space reasons.

Note also that the transformations can be combined. For example, a soft time constraint could have a disjunctive condition.

The **blowup** resulting from the transformations is straightforwardly seen to be linear in the number of disjuncts or of alternative discounts, respectively. Our constructions and the resulting blowup straightforwardly carry over to any bidding language that allows XOR as outermost connective.

2.7 Expressive power

We say a valuation is **finite** if it has a finite domain (i.e., yields non-$\perp$ for finitely many time lines only) consisting of finite sequences of finite transformations (i.e., with finite input and output). The XOR language with time constraints is **fully expressive** for finite valuations: XOR bids with *relative* time constraints can express all finite valuations that use relative time; XOR bids with *absolute* time constraints can express all finite valuations. The proof is simple: Take an XOR bid with one atomic bid $\text{BID}(\mathcal{D}, p)$ for each Σ in the domain of v, with $\mathcal{D}$ set to permit Σ and p set to $v(\Sigma)$, and impose the order corresponding to Σ using time constraints (note that there may be several atomic bids with the same transformations in $\mathcal{D}$, but different time points).

3 Winner determination

We now study the winner determination problem (WDP). This is the problem, faced by the auctioneer, of determining which transformations to award to which bidder, so as to maximize (minimize) the sum of payments collected (made), given the bids of the bidders expressed in our bidding language. This may be interpreted as computing a solution that maximizes revenue for the auctioneer, or utilitarian social welfare for the collective of bidders (if we interpret prices offered as reflecting bidder utility). Note that we are interested in the *algorithmic* aspects of the WDP. Game-theoretical considerations, such as how to devise a more sophisticated pricing rule that would induce bidders to bid truthfully, are orthogonal to the algorithmic problem addressed here. (We briefly comment on mechanism design issues in Sect. 5, but this is not the topic of this paper.)

For symmetry between bidders and auctioneer, we do *not* assume free disposal for the auctioneer (just like for the bidders), i.e., he does not want to end up with any goods except the required ones. Note, however, that the formulations are easily adapted to allow free disposal (and we point out the necessary changes along the way).

After formulating the WDP, we give an integer program [9] solving it. We aim at keeping the descriptions short and focus on the changes compared to the version from [2]. The advantage of this approach, besides showing how few modifications are necessary, is that it is modular and can (hopefully) be combined without too much effort with other extensions or optimizations.

3.1 WDP with time constraints

The *input* to the WDP consists of

- a bid expression Bid_i in XOR normal form together with a conjunction of time constraints φ_i, for each bidder i;
- a multiset $\mathcal{U}_{in}$ of goods the auctioneer holds in the beginning;
- and a multiset $\mathcal{U}_{out}$ of goods the auctioneer wants to end up with.

Let Bid_{ij} denote the jth atomic bid $\text{BID}(\mathcal{D}_{ij}, p_{ij})$ occurring within Bid_i, let t_{ijk} be a unique label for the kth transformation in $\mathcal{D}_{ij}$ (for some arbitrary but fixed ordering of $\mathcal{D}_{ij}$), and let τ_{ijk} be the time point identifier associated with transformation t_{ijk}. Let $(\mathcal{I}_{ijk}, \mathcal{O}_{ijk})$ be the actual transformation labelled with t_{ijk}, and T be the set of all t_{ijk}.

An **allocation sequence** Σ resembles the time line we introduced before, but can only contain transformations actually offered by some bidder, and each one at most once. That is, Σ now is a permutation of a subset of T, possibly interspersed with clock ticks c.

We write $t_{ijk} \in \Sigma$ to say that the kth transformation in the jth atomic bid of bidder i has been selected, and we write $\Sigma(t_{ijk})$ to denote the sequence number of t_{ijk} (starting from 1) if $t_{ijk} \in \Sigma$. By Σ_i we denote the **projection** of Σ to bidder i, that is, Σ with each t_{ijk} replaced by $(\mathcal{I}_{ijk}, \mathcal{O}_{ijk}, \tau_{ijk})$ and all $t_{i'jk}$ replaced by c for $i' \neq i$. By $(\mathcal{I}^m, \mathcal{O}^m)$ we denote the mth transformation in Σ. Thus, we have two ways of referring to a selected transformation: by its position in the received bids (t_{ijk}) and by its position m in the allocation sequence.

Given Σ, we can inductively define the bundle of goods held by the auctioneer after each step (let $g \in G$ be any good, and let $\mathcal{M}^0 = \mathcal{U}_{in}$):[3]

$$\mathcal{M}^m(g) = \mathcal{M}^{m-1}(g) + \mathcal{O}^m(g) - \mathcal{I}^m(g) \tag{1}$$

under the condition that

$$\mathcal{M}^{m-1}(g) \geq \mathcal{I}^m(g). \tag{2}$$

Given a multiset $\mathcal{U}_{in}$ of goods available to the auctioneer, a multiset $\mathcal{U}_{out}$ of goods required by the auctioneer, and a set of bids Bid_i with time constraints φ_i, an allocation sequence Σ is a **valid solution** if

(i) for each bidder i, some $\mathcal{D}_{ij}$ permits Σ_i, or $\Sigma_i \in \{c\}^*$

(ii) for each bidder i, $\Sigma_i \models \varphi_i$

(iii) Eq. (1) and (2) hold for each transformation $(\mathcal{I}^m, \mathcal{O}^m) \in \Sigma$ and each good $g \in G$

(iv) for each good $g \in G$, $\mathcal{M}^{|\Sigma|}(g) = \mathcal{U}_{out}(g)$.[4]

The **revenue** for the auctioneer associated with a valid solution Σ is the sum of the prices of the selected atomic bids, i.e., $\sum \dot{\{} p_{ij} \mid \exists k : t_{ijk} \in \Sigma \dot{\}}$.

Given multisets $\mathcal{U}_{in}$ and $\mathcal{U}_{out}$ of initial and required goods and a set of bids with time constraints, the **winner determination problem** (WDP) consists in finding a valid solution that maximizes the auctioneer's revenue.

3.2 Original integer program

In this part, we closely follow Cerquides et al. [2]. The main issue is to decide, for each offered transformation, whether it should be selected for the solution sequence, and if so, at which position. Thus, we define a set of binary decision variables $x_{ijk}^m \in \{0, 1\}$, each of which takes on value 1 if and only if the transformation t_{ijk} is selected at the mth position of the solution sequence.

The position number m ranges from 1 to an upper bound M on the solution sequence length. For the time being, we take $M = |T|$, the overall number of transformations, accommodating all sequences that can be formed using only transformations (and not clock ticks). We consider an alternative way for specifying M at the end of Sect. 3.3.

Further, i ranges over all bidders; j ranges for each bidder i from 1 to the number of atomic bids submitted by i; and k ranges for each atomic bid j of bidder i from 1 to the number of transformations in that bid.

We use the following auxiliary binary decision variables: x^m takes on value 1 if and only if any transformation is selected at the mth position; x_{ijk} takes on value 1 if and only if transformation t_{ijk} is selected at all; and x_{ij} takes on value 1 if and only if any of the transformations in the jth atomic bid of bidder i are selected.

The following set of constraints define a valid solution *without* taking time constraints into account (i.e., neglecting (ii) in the valid solution definition above):

(1) Select either all or no transformations from an atomic bid (cf. (i) above):

$$x_{ij} = x_{ijk} \quad (\forall ijk)$$

[3] Given a multiset $\mathcal{S} \in \mathbb{N}^G$ and an item $g \in G$, we write $\mathcal{S}(g)$ to denote the number of copies of g in $\mathcal{S}$.

[4] Replace $=$ by $\geq$ to model free disposal.

(2) Select at most one atomic bid from each XOR normal form bid (cf. (i) above):
$$\sum_j x_{ij} \leq 1 \quad (\forall i)$$
(3) Select each transformation at most for one position: $x_{ijk} = \sum_m x_{ijk}^m \quad (\forall ijk)$
(4) For each position, select at most one transformation: $x^m = \sum_{ijk} x_{ijk}^m \quad (\forall m)$
(5) There should be no gaps in the sequence: $x^m \geq x^{m+1} \quad (\forall m)$
Note that this is strictly speaking not required; indeed we drop this constraint later on in order to allow clock ticks between transformations.
(6) Treating each $\mathcal{M}^m(g)$ as integer decision variable, ensure that necessary input goods are available (cf. (iii) above):

$$\mathcal{M}^m(g) = \mathcal{U}_{in}(g) + \sum_{\ell=1}^{m} \sum_{ijk} x_{ijk}^{\ell} \cdot (\mathcal{O}_{ijk}(g) - \mathcal{I}_{ijk}(g))$$
$$\mathcal{M}^m(g) \geq \sum_{ijk} x_{ijk}^m \cdot \mathcal{I}_{ijk}(g) \qquad (\forall g \in G, \forall m)$$

(7) In the end, the auctioneer should have the bundle $\mathcal{U}_{out}$ (cf. (iv) above):[5]
$$\mathcal{M}^M(g) = \mathcal{U}_{out}(g) \quad (\forall g \in G)$$

Solving the WDP now amounts to solving the following integer program:

$$\max \sum_{ij} x_{ij} \cdot p_{ij}, \quad \text{subject to constraints (1)–(7)}$$

A valid solution is then obtained by making transformation t_{ijk} the mth element of the solution sequence Σ exactly when $x_{ijk}^m = 1$.

3.3 Modified integer program

To implement time constraint handling (thus obeying (ii) in the definition of valid solution given above), we first introduce an additional set of auxiliary binary decision variables $y_{ijk}^m \in \{0, 1\}$, taking on value 1 if and only if transformation t_{ijk} is selected at the mth position *or earlier* in the solution sequence. This is achieved by adding the following constraint:

(8) y_{ijk}^m should be 1 iff $t_{ijk} \in \Sigma$ and $\Sigma(t_{ijk}) \leq m$: $\quad y_{ijk}^m = y_{ijk}^{m-1} + x_{ijk}^m \quad (\forall ijkm)$, with $y_{ijk}^0 = 0$.

We now give implementations for our two variants of time constraints.

Relative time. Each bidder i's time constraint formula is a conjunction of atomic time constraints, and all bidders' time constraints need to be satisfied. The following set of integer constraints takes care of this.

(9a) For each $\tau_{ijk} < \tau_{ij'k'}$ occurring in $\bigwedge_i \varphi_i$: $\quad y_{ijk}^m \geq y_{ij'k'}^{m+1} \quad (\forall m)$.

In accordance with the time constraint semantics, if neither t_{ijk} nor $t_{ij'k'}$ occurs in the solution sequence, this requirement is vacuously satisfied since both sides stay 0. If $t_{ij'k'}$ *does* occur, then $y_{ij'k'}^m$ will become 1 at some point m. In this case, the requirement boils down to y_{ijk}^{m-1} being 1 as well, so t_{ijk} must have occurred already.

Solving the WDP with relative time constraints thus amounts to the same optimization as before, but subject to constraints (1)–(8) and (9a).

Absolute time. In order to have an absolute notion of time, we need some way of mapping points of a possible solution sequence to an absolute time line. The simplest

[5] With free disposal, $=$ would become $\geq$.

way is to interpret each sequence point itself as a time unit (a minute, a day, a week, ...), and this is the approach we take.

Before giving the formalization, we need to discuss some conceptual details. If we interpret steps in the sequence as absolute time units, some issues arise which did not matter before. Firstly, while it may be acceptable to break time down into discrete steps of equal duration, it is not so easy to defend that any transformation that can possibly be offered should have exactly that duration. Secondly, there is no reason why the auctioneer should wait for one transformation to end before commissioning the next transformation, which may be offered by a different, idling bidder, unless the output of the former is needed as input to the latter. To some extent, these issues can be addressed by a purely conceptual extension presented in Sect. 4. However, we leave it to future work to design frameworks which handle time in a more flexible way and truly optimize for effective parallelizations. For our purposes, we simply assume that the auctioneer is busy when he is delivering or receiving goods of some particular transformation, and cannot deal with several bidders simultaneously.

To start the formalization, first of all we drop constraint (5). As mentioned, it is not strictly speaking necessary anyway, and since now the bidders can refer to arbitrary absolute time points, we actually might have to accept gaps in the sequence.

Now a technical issue arises: The length of possible solution sequences is no longer bounded by $|T|$. While it may be possible to find a correct bound by looking at all numbers occurring in the bidders' time constraints, we settle for a different solution: The auctioneer manually specifies M, the maximum length of the solution sequence.

At first glance this seems like a pure loss of generality; however the auctioneer may profit from having some control over the size of the WDP he has to solve, and he can always iterate over different values for M in his search for a good solution. Economically speaking, it also makes sense that the auctioneer wants some control over the length of his supply chain, rather than allowing an arbitrary length. Indeed, he might have graded preferences over the time his supply chain takes, as discussed in Sect. 3.4.

We now give the integer constraints for handling absolute time constraints.

(9b) For each $\tau_{ijk} + \xi < \tau_{ij'k'} + \xi'$ occurring in $\bigwedge_i \varphi_i$: $\quad y_{ijk}^{m+\xi'} \geq y_{ij'k'}^{m+\xi+1} \quad (\forall m)$

For each $\tau_{ijk} + \xi = \tau_{ij'k'} + \xi'$ occurring in $\bigwedge_i \varphi_i$: $\quad y_{ijk}^{m+\xi'} = y_{ij'k'}^{m+\xi} \quad (\forall m)$

(10) For each $\tau_{ijk} \circ \xi$, with $\circ \in \{=, <, >\}$, occurring in $\bigwedge_i \varphi_i$: $\quad x_{ijk}^{m} = 0 \quad (\forall m \not\circ \xi)$.

Constraint (9b) requires some explanation. First of all, note that (9a), the version for relative time, is covered as a special case. As indicated by the semantics, the absolute time variant is thus an extension of the relative time variant. Secondly, note that the second half of (9b) can be obtained from the first half if interpreted as an abbreviation, as in Sect. 2.5. Now consider the case where $\xi' = 0$. Intuitively speaking, the time constraint then says that t_{ijk} must take place at least $\xi + 1$ time steps before $t_{ij'k'}$. That is, whenever $t_{ij'k'}$ is selected, t_{ijk} must already have been selected for at least $\xi + 1$ time steps. In terms of the integer program, this means that, for all positions m, $y_{ij'k'}^{m}$ must be 0 *unless* $y_{ijk}^{m-\xi-1}$ was already 1. Now it is only a small step to the formulation in (9b).

Solving the WDP with absolute time constraints amounts to the same optimization as before, but subject to constraints (1)–(4), (6)–(8), (9b) and (10).

A valid solution is then obtained by making transformation t_{ijk} the mth element of the solution sequence Σ if and only if $x^m_{ijk} = 1$, and using a clock tick c as mth element when there is no x^m_{ijk} which equals 1 (i.e., when $x^m = 0$).

3.4 Valuation for the auctioneer

Given that we decided to require the auctioneer to specify the maximum length M of the solution sequence (for the absolute-time variant of the framework), we may also want to enable him to express more detailed preferences over durations. This can be achieved in a neat way, also enabling the auctioneer to express graded preferences over final bundles.

So assume the auctioneer derives a certain value from a given supply chain, depending on its overall duration and its outcome, the bundle of goods he owns in the end. Note that we here assume absolute time; with relative time, preferences over *durations* do not make much sense, but the results can easily be adjusted to only model preferences over *outcomes*.

We thus assume the auctioneer's valuation is a function $u : \mathbb{N} \times \mathbb{N}^G \to \mathbb{R} \cup \{\perp\}$, mapping duration/outcome pairs to a value or $\perp$, meaning the duration/outcome pair is not acceptable. This valuation can be incorporated into the WDP in the following way.

After receiving the bids, the auctioneer decides on a maximum duration M and creates an additional bid under an unused bidder identity:

$$\underset{\{(m,\mathcal{U}) \mid u(m,\mathcal{U}) \neq \perp\}}{\text{XOR}} \text{BID}(\{(\mathcal{U}, \{\circledcirc\}, \tau_{m,\mathcal{U}})\}, u(m,\mathcal{U})),$$

where $\circledcirc$ is a special token that does not occur as a good in any other bid, together with time constraints:

$$\bigwedge_{\{(m,\mathcal{U}) \mid u(m,\mathcal{U}) \neq \perp\}} \tau_{m,\mathcal{U}} = m.$$

The transformations in this bid are to be thought of as terminal transformations: they denote the possible time points and outcomes at which a solution sequence may end, and the associated values for the auctioneer. Using this method, the auctioneer's valuation can be expressed with very few changes to the integer program:

- The terminal transformations should only be used at the respective intended positions in the sequence; this is ensured by the given time constraints.
- At *most* one of them should be used; this is ensured by the XOR (and strictly speaking also follows from the last point below).[6]
- At *least* one[7] of them should be used; this can be ensured by setting $\mathcal{U}_{out} = \{\circledcirc\}$.
- The unique terminal transformation that is actually used should indeed be the end of the solution sequence.[8]

[6] Even more strictly speaking, it also follows from the next requirement and the fact that we assume no free disposal; we include it nevertheless for conceptual clarity and in order to accommodate a possible free disposal assumption.

[7] This could read "*exactly* one", but again, we want to accommodate a possible free disposal assumption.

[8] As a last remark, this requirement could be dropped if we *did* assume free disposal and all bids' prices were positive.

For this last point, we need an additional integer constraint:

(11) No transformations are scheduled after a terminal transformation:

$$x^{m+1}_{ijk} \leq 1 - y^{m}_{-1j'k'} \quad (\forall ijkj'k'm)$$

(-1 being the auctioneer's "bidder identity").

While the above requirements could also be encoded more directly and more efficiently into the integer program, for clarity we here restrict ourselves to this version using the high-level features of the bidding language.

Many further extensions and optimizations along these lines are conceivable. We do not try to exhaust them here, but sketch only one example. The auctioneer might want to extract some goods $\mathcal{U}$ from the supply chain by some intermediate time point ξ, not necessarily at its end. To express this, he can add a transformation $(\mathcal{U}, \{\diamond\}, \tau)$ with time constraint $\tau < \xi + 1$ to his bid, and add $\diamond$ to $\mathcal{U}_{out}$. Dropping constraint (11) for this transformation, he makes it non-terminal. He can also make this a soft requirement by including another transformation that yields $\diamond$ from no input and attaching appropriate prices to the corresponding bids.

3.5 Computational complexity

As in the original model [2], the WDP for mixed auctions with time constraints is NP-complete: NP-hardness follows from NP-hardness of the WDP for standard combinatorial auctions [8] and NP-membership follows from the fact that the validity of a given allocation sequence can clearly be verified in polynomial time. Fionda and Greco [4] have started to chart the tractability frontier for a slightly simplified version of the original framework by Cerquides et al. [2], using various criteria to restrict the class of allowed bids. Their results concerning the XOR language still hold in our extended framework. In particular, even with time constraints the WDP still remains tractable if only one transformation outputs any particular good [4, Theorem 3.7]. Further tractability islands may be identified in future work.

Regarding the integer program, while there is room for optimizations, the number of variables we introduce is of the same order as in the original formulation: $O(n^2)$, where n is the number of transformations occurring in the bids submitted. The most recent work on winner determination algorithms for mixed auctions has tried to reduce the number of decision variables needed so as to improve performance [5,7]. Due to the modular nature of our approach, we are optimistic that it will be possible to take advantage of these optimizations and integrate them with our extensions.

4 Intervals

It may be desirable to allow transformations to overlap or take place during others, and to allow transformations to have different durations. Interestingly, **intervals** can be handled in our framework without any additional machinery. A transformation with start time and end time can be rewritten into two transformations with single time points and an appropriate time constraint:

$$(\mathcal{I}, \mathcal{O}, [\tau, \tau']) \quad \rightsquigarrow \quad \begin{array}{l} (\mathcal{I}, \emptyset, \tau), (\emptyset, \mathcal{O}, \tau') \\ \tau < \tau' \end{array}$$

Since the replacement takes place within a single atomic bid, it is guaranteed that either both the start and end transformations will be selected, or neither. That is, the interval transformation remains intact.

The usual interval relations (see the interval calculus by Allen [1]; due to sequentiality we consider only strict relations) can be defined as macros:

$$\begin{aligned} [\tau_1, \tau_1'] \text{ BEFORE } [\tau_2, \tau_2'] &\rightsquigarrow \tau_1' < \tau_2 \\ [\tau_1, \tau_1'] \text{ OVERLAPS } [\tau_2, \tau_2'] &\rightsquigarrow \tau_1 < \tau_2 \wedge \tau_1' < \tau_2' \\ [\tau_1, \tau_1'] \text{ DURING } [\tau_2, \tau_2'] &\rightsquigarrow \tau_2 < \tau_1 \wedge \tau_1' < \tau_2 \end{aligned}$$

With absolute time, absolute restrictions on the durations can also be implemented:

$$duration([\tau, \tau']) \circ \xi \ \rightsquigarrow \ \tau' \circ \tau + \xi, \qquad \circ \in \{<, >, =\}$$

Note that expressions like $duration(\cdot) > duration(\cdot)$ are not so straightforwardly expressible, but arguably also much less useful in the context of specifying bids.

5 Conclusions and related work

We presented an extension to the existing framework of mixed multi-unit combinatorial auctions [2], enabling bidders to impose time constraints on the transformations they offer.

In the original framework, the auctioneer is free to schedule the offered transformations in any way suitable to achieve his desired outcome, while bidders are left with no control over this process. Our work redresses this asymmetry, thus representing an important step towards a more realistic model of supply chain formation, where bidders themselves may have supply chains or other factors restricting the possible schedules for performing certain transformations.

Starting from a very basic core language for expressing time constraints, we have given various extensions, many purely syntactic, showing the somewhat unexpected power inherent to the core language.

We have also extended the integer program given in [2] to handle time constraints. Our extensions are modular in a way that will facilitate combining them with other extensions and optimizations for mixed auctions, such as [5,7].

Time constraints have been applied to different types of combinatorial auctions in the literature. For example, Hunsberger and Grosz [6] extended an existing algorithm for winner determination in combinatorial auctions to allow precedence constraints when bidding on roles in a prescribed action plan ("recipe"). Collins [3] permitted relative time constraints in a combinatorial reverse auction over combinations of tasks, and tested the efficiency of various approaches to solving the winner determination problem.

Auction frameworks involving time have also been fruitfully applied to problems of distributed scheduling. In Wellman et al. [10], time constraints do not enter separately, rather time slots are the actual objects being auctioned, and game-theoretic and mechanism design issues are discussed.

While it would be interesting to examine whether the insights about efficiency and alternative approaches to handling time could be applied to our framework, the roles, tasks, and time slots being auctioned in those contributions are "atomic", and the formulations and results do not easily translate to transformations in the context of mixed auctions.

Concerning mechanism design, with finite valuations, the incentive-compatibility of the Vickrey-Clarke-Groves (VCG) mechanism carries over from standard combinatorial auctions to mixed multi-unit combinatorial auctions with time constraints (see also [2]). It is a question of independent interest, whether and how this can be extended to non-finite valuations when still allowing only finite bids. In this case the bidders would not be able to express their true valuations exactly, so it is not obvious how truthfulness and incentive compatibility should be defined.

Other topics for future work include the exact interplay between the various syntactic extensions we have given, defining a uniform general language, and determining whether some of the features can be implemented in more direct (and efficient) ways than through the translation to the core language used in this work. The same holds for the underlying bidding language, where operators such as OR may be executed more efficiently than through translation to XOR.

Changing the integer program to allow OR instead of XOR is straightforward; an XOR-of-OR language, generalizing both the XOR and the OR languages, can be accommodated with more extensive changes, buying the advantage of preserving our constructions for disjunctive and soft time constraints.

Finally, an empirical analysis needs to be performed, including testing and optimizing our integer program.

References

1. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
2. J. Cerquides, U. Endriss, A. Giovannucci, and J. A. Rodríguez-Aguilar. Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In *Proc. IJCAI-2007*, Hyderabad, India, 2007.
3. J. E. Collins. *Solving combinatorial auctions with temporal constraints in economic agents*. PhD thesis, University of Minnesota, 2002.
4. V. Fionda and G. Greco. Charting the tractability frontier of mixed Multi-Unit combinatorial auctions. In *Proc. IJCAI-2009*, Pasadena, CA, July 2009.
5. A. Giovannucci, M. Vinyals, J. A. Rodríguez-Aguilar, and J. Cerquides. Computationally efficient winner determination for mixed multi-unit combinatorial auctions. In *Proc. AAMAS-2008*. IFAAMAS, 2008.
6. L. Hunsberger and B. J. Grosz. A combinatorial auction for collaborative planning. In *Proc. 4th International Conference on MultiAgent Systems*. IEEE Computer Society, 2000.
7. B. Ottens and U. Endriss. Comparing winner determination algorithms for mixed multi-unit combinatorial auctions. In *Proc. AAMAS-2008*. IFAAMAS, 2008.
8. M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinational auctions. *Management Science*, 44(8):1131–1147, Aug. 1998.
9. A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
10. M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35(1–2):271–303, 2001.

Characterization of Revenue Monotonicity in Combinatorial Auctions

Taiki Todo, Atsushi Iwasaki, and Makoto Yokoo

Graduate School of Information Science and Electrical Engineering,
Kyushu University, Japan
{todo@agent., iwasaki@, yokoo@}is.kyushu-u.ac.jp

Abstract. An auction mechanism consists of an allocation rule and a payment rule. There have been several studies on characterizing strategy-proof allocation rules. One desirable property that an auction mechanism should satisfy is revenue monotonicity; a seller's revenue is guaranteed to weakly increase as the number of bidders grows. In this paper, we first identify a simple condition called *summation-monotonicity* for characterizing strategy-proof and revenue monotone allocation rules. To the best of our knowledge, this is the first attempt to characterize revenue monotone allocation rules. Based on this characterization, we also examine the connections between revenue monotonicity and *false-name-proofness*, which means a bidder cannot increase his utility by submitting multiple bids under fictitious names. In a single-item auction, we show that a mechanism is false-name-proof if and only if it is strategy-proof and revenue monotone. On the other hand, we show that there exists no combinatorial auction mechanism that is simultaneously revenue monotone and false-name-proof under some minor conditions.

1 Introduction

Mechanism design of combinatorial auctions has become an integral part of Electronic Commerce and a promising field for applying AI and agent technologies. Among various studies related to Internet auctions, those on combinatorial auctions have lately attracted considerable attention. Mechanism design is the study of designing a rule/protocol that achieves several desirable properties, assuming that each agent/bidder hopes to maximize his own utility.

One desirable property of a combinatorial auction mechanism is that it is *strategy-proof*. A mechanism is strategy-proof if, for each bidder, reporting his true valuation is a *dominant strategy*, i.e., an optimal strategy regardless of the actions of other bidders. In theory, the revelation principle states that in the design of an auction mechanism, we can restrict our attention to strategy-proof mechanisms without loss of generality [1]. In other words, if a certain property, e.g., Pareto efficiency or high revenue, can be achieved using an auction mechanism in a dominant strategy equilibrium, which is a combination of dominant strategies of bidders, then the property can also be achieved using a strategy-proof auction mechanism.

A combinatorial auction mechanism consists of an allocation rule that defines the allocation of goods for each bidder and a payment rule that defines the payment of each winner. There have been many studies on characterizing strategy-proof social choice functions (an allocation rule in combinatorial auctions) in the literature of social choice theory. This is also called the implementability of social choice functions. In particular, a family of *monotonicity* concepts was identified to characterize implementable social choice functions. For example, Lavi *et al.* [2] proposed *weak-monotonicity* and showed that it is a necessary and sufficient condition for strategy-proof mechanisms when several assumptions hold on the domain of types. Such a characterization of allocation rules is quite useful for developing/verifying strategy-proof mechanisms. These conditions are defined only on an allocation rule; if it satisfies such a condition, it is guaranteed that there exists an appropriate payment rule that achieves strategy-proofness. Thus, a mechanism designer can concentrate on allocation rules when developing a new mechanism or verifying an existing one.

Besides these studies, *revenue monotonicity* is recognized as one of desirable properties a mechanism should satisfy [3]. A mechanism is revenue monotone if the seller's revenue from an auction is guaranteed to weakly increase as the number of bidders grows. The property is quite reasonable, since a growing number of bidders increases competition. However, it is shown that even the theoretically well-founded Vickrey-Clarke-Groves (VCG) mechanism does not achieve revenue monotonicity. Nevertheless, there has been very little work on characterizing revenue monotone mechanisms. One notable exception is Rastegari *et al.* [4], who proved that there exists no mechanism that is revenue monotone, strategy-proof, and *weakly maximal*, where weak maximality is a weaker notion of Pareto efficiency. Furthermore, they mentioned a connection between revenue monotonicity and *false-name-proofness*, which is known as another desirable property of combinatorial auctions on the Internet.

False-name-proofness generalizes strategy-proofness by assuming that a bidder can submit multiple bids under fictitious identifiers, e.g., multiple e-mail addresses [5]. Several false-name-proof mechanisms have been developed so far. Furthermore, besides combinatorial auctions, false-name-proof mechanism design has been discussed in domains such as voting [6] and coalitional games [7]. Also, Todo *et al.* [8] fully characterized false-name-proof allocation rules by a condition called *sub-additivity*.

To the best of our knowledge, our paper is the first attempt to characterize revenue monotone allocation rules in combinatorial auctions. First, we identify a condition called *summation-monotonicity* and prove that weak-monotonicity and summation-monotonicity fully characterize strategy-proof and revenue monotone allocation rules. Second, our characterization successfully clarifies the connections between revenue monotonicity and false-name-proofness. Summation-monotonicity and sub-additivity look quite similar, but they are different and interact in a rather complicated way. In single-item auctions, we show that they are basically equivalent; a mechanism is false-name-proof if and only if it is strategy-proof and revenue monotone. On the other hand, we show that these two conditions cannot coexist in combinatorial auctions; under some minor conditions, there exists no combinatorial auction mechanism that is simultaneously revenue monotone and false-name-proof.

2 Preliminaries

Assume there exists a set of potential bidders $\mathbb{N} = \{1, 2, \ldots, n\}$ and a set of goods $G = \{g_1, g_2, \ldots, g_m\}$. Let us define $N \subseteq \mathbb{N}$ as the set of bidders participating in an auction. Each bidder $i \in N$ has his preferences for each bundle or goods $B \subseteq G$. Formally, we model this by supposing that bidder i privately observes a parameter (or signal) θ_i that determines his preferences. We refer to θ_i as the *type* of bidder i and assume it is drawn from a set Θ_i.

Let us denote the set of all possible type profiles as $\Theta_{\mathbb{N}} = \Theta_1 \times \ldots \times \Theta_n$ and a type profile as $\theta = (\theta_1, \ldots, \theta_n) \in \Theta_{\mathbb{N}}$. Observe that type profiles always have one entry for every potential bidder, regardless of the set of participating bidders N. We use a symbol $\mathbf{0}$ in the vector θ as a placeholder for each non-participating bidder $i \notin N$ and represent $(\theta_1, \ldots, \theta_{i-1}, \mathbf{0}, \theta_{i+1}, \ldots, \theta_n)$ as θ_{-i}, for $\theta = (\theta_1, \ldots, \theta_{i-1}, \theta_i, \theta_{i+1}, \ldots, \theta_n)$. When a set of bidders N participates in the auction, we denote the set of possible type profiles that can be reported by N as $\Theta_N (\subseteq \Theta_{\mathbb{N}})$. That is, Θ_N is the set of all type profiles θ for which $\theta_i = \mathbf{0}$ if and only if $i \notin N$.

We assume a *quasi-linear, private value* model with *no allocative externality*; the utility of bidder i, when i obtains a bundle, i.e., a subset of goods $B \subseteq G$ and pays p, is represented as $v(\theta_i, B) - p$. We also assume a valuation v is normalized by $v(\theta_i, \emptyset) = 0$, satisfies *free disposal*, i.e., $v(\theta_i, B') \geq v(\theta_i, B)$ for all $B' \supseteq B$, and satisfies *no externalities*, i.e., a valuation v is determined only by his obtained bundle. We call each Θ_i that satisfies these conditions a *rich domain* [9]. In other words, the domain of types Θ_i is rich enough to contain all possible valuations. This assumption is required so that weak-monotonicity characterizes strategy-proofness.

A combinatorial auction mechanism $\mathcal{M}$ consists of an allocation rule X and a payment rule p. When a set of bidders N participates, an allocation rule is defined as $X : \Theta_N \to A_N$, where A_N is a set of possible allocations over N. Similarly, a payment rule is defined as $p : \Theta_N \to \mathbb{R}^N_+$. Let X_i and p_i respectively denote the bundle allocated to bidder i and the amount bidder i must pay. We use notations $X(\theta_i, \theta_{-i})$ and $p(\theta_i, \theta_{-i})$ to represent the allocation and payment when the declared type of bidder i is θ_i and the declared type profile of other bidders is θ_{-i}.

For simplicity, we restrict our attention to *deterministic* mechanisms and assume that a mechanism is *almost anonymous* across bidders and goods; obtained results are invariant under the permutation of the identifiers of bidders/goods except for the case of ties. We also assume that a mechanism satisfies a property called *consumer sovereignty* [4]; there always exists a type θ_i for bidder i, where bidder i can obtain bundle B. In other words, if bidder i's valuation for B is high enough, then he can obtain B. This property is also called *player* *decisiveness* [2]. Furthermore, we restrict our attention to *individually rational* mechanisms. A mechanism is individually rational if $\forall N \subseteq \mathbb{N}$, $\forall i \in N$, $\forall \theta_i$, $\forall \theta_{-i}$, $v(\theta_i, X_i(\theta_i, \theta_{-i})) - p_i(\theta_i, \theta_{-i}) \geq 0$ holds. This means that no participant obtains negative utility by reporting his true type.

Let us introduce the notion called *strategy-proofness*.

Definition 1 (Strategy-proofness).
A combinatorial auction mechanism $\mathcal{M}(X, p)$ is strategy-proof if $\forall N \subseteq \mathbb{N}, \forall i \in N, \forall \theta_{-i}, \forall \theta_i, \forall \theta'_i$, $v(\theta_i, X_i(\theta_i, \theta_{-i})) - p_i(\theta_i, \theta_{-i}) \geq v(\theta_i, X_i(\theta'_i, \theta_{-i})) - p_i(\theta'_i, \theta_{-i})$ holds.

A mechanism is strategy-proof if reporting true type θ_i is a (weakly) dominant strategy for any bidder i with type θ_i and any type profile θ_{-i}; it maximizes his utility regardless of the other bidders' reports. A strategy-proof allocation rule is fully characterized by a simple condition called *weak-monotonicity*, assuming the type domain is rich [9].

Definition 2 (Weak-monotonicity).
An allocation rule X satisfies weak-monotonicity if $\forall N \subseteq \mathbb{N}, \forall i \in N, \forall \theta_{-i}, \forall \theta_i, \forall \theta'_i$, $v(\theta_i, X_i(\theta_i, \theta_{-i})) - v(\theta_i, X_i(\theta'_i, \theta_{-i})) \geq v(\theta'_i, X_i(\theta_i, \theta_{-i})) - v(\theta'_i, X_i(\theta'_i, \theta_{-i}))$ holds.

Lavi *et al.* [2] and Bikhchandani *et al.* [9] proved that if an allocation rule is weakly monotone, we can always find an appropriate payment rule to truthfully implement the allocation rule. Thus, when designing an auction mechanism, we can concentrate on designing an allocation rule and forget about the payment rule, at least for a while.

Next, let us introduce the notion of *revenue monotonicity* [3], which is known as another desirable property of combinatorial auctions.

Definition 3 (Revenue monotonicity).
A combinatorial auction mechanism $\mathcal{M}(X, p)$ is revenue monotone if $\forall N \subseteq \mathbb{N}, \forall \theta, \forall j \in \mathbb{N}$,

$$\sum_{i \in \mathbb{N}} p_i(\theta) \geq \sum_{i \in \mathbb{N} \setminus \{j\}} p_i(\theta_{-j}). \tag{1}$$

The left-hand side indicates the seller's revenue from the auction when the set of bidders N participates in the auction. The right-hand side indicates the seller's revenue when bidder j drops out. In other words, a combinatorial auction is revenue monotone if the seller's revenue does not increase by dropping a bidder.

This property is quite reasonable, since a growing number of bidders increases competition. However, even the theoretically well-founded VCG mechanism is not always revenue monotone. Rastegari *et al.* [4], a seminal work on revenue monotonicity, shows the impossibility result that there exists no deterministic strategy-proof, weakly maximal combinatorial auction mechanism that is revenue monotone. Roughly speaking, a combinatorial auction mechanism is *weakly maximal* if its allocation cannot be augmented to cause a losing bidder to win without hurting winning bidders. It is a weaker notion of Pareto efficiency.

Furthermore, Rastegari *et al.* [4] mentioned a connection between revenue monotonicity and *false-name-proofness*, which is known as another desirable property of combinatorial auctions [5]. This is a kind of generalization of strategy-proofness for an environment by assuming that a bidder can use multiple identifiers. To introduce this property along with our model, we add several notations. Let us consider a situation where bidder i uses s false identifiers $id_1, \ldots, id_s$ and define a mapping function ϕ such that $\phi(i) = \{id_1, \ldots, id_s\}$; i.e., $\phi(i)$ represents a set of identifiers owned by bidder i. Let us represent a type profile as $\theta = (\theta_{id_1}, \ldots, \theta_{id_s}, \theta_{s+1}, \ldots, \theta_n)$, and similarly represent a type profile reported by the set of identifiers $\phi(i)$ as $\theta_{\phi(i)} = (\theta_{id_1}, \ldots, \theta_{id_s})$. Here we use $\theta_{id_1}, \ldots \theta_{id_s}$ instead of $\theta_1, \ldots, \theta_s$ for convenience. On the other hand, let $(\theta_i, \mathbf{0}, \ldots, \mathbf{0})$ denote the type profile when bidder i reports θ_i with only one identifier although he can use s identifiers. That is, $\mathbf{0}$ means that the identifier is not used by bidder i. Note that these notations can be introduced w.l.o.g., since we assume almost anonymous mechanisms. Furthermore, to consistently address false-name-proofness, we represent a type profile reported by the set of participating bidders other than $\phi(i)$ as $\theta_{-\phi(i)} = (\mathbf{0}, \ldots, \mathbf{0}, \theta_{s+1}, \ldots, \theta_n)$.

Now we are ready to introduce false-name-proofness.

Definition 4 (False-name-proofness).
A combinatorial auction mechanism $\mathcal{M}(X, p)$ is false-name-proof if $\forall N \subseteq \mathbb{N}, \forall i \in N, \forall \phi(i), \forall \theta_{-\phi(i)}, \forall \theta_i, \forall \theta_{\phi(i)}$,

$$\begin{aligned} & v(\theta_i, X_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)})) - p_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)}) \\ & \geq v(\theta_i, \textstyle\bigcup_{l \in \phi(i)} X_l(\theta_{\phi(i)}, \theta_{-\phi(i)})) - \sum_{l \in \phi(i)} p_l(\theta_{\phi(i)}, \theta_{-\phi(i)}). \end{aligned} \quad (2)$$

Note that when $|\phi(i)| = 1$, the definition becomes equivalent to strategy-proofness. It has been shown that VCG is not false-name-proof and that there exists no false-name-proof, Pareto efficient mechanism [5].

3 Characterization of Revenue Monotonicity

This section introduces a simple condition called summation-monotonicity that fully characterizes revenue monotone allocation rules when coupled with weak-monotonicity.

Definition 5 (Summation-monotonicity).
An allocation rule X satisfies summation-monotonicity if $\forall N \subseteq \mathbb{N}, \forall \theta, \forall j \in \mathbb{N}$,

$$\begin{aligned} &\forall \theta'_i \;\text{ s.t. }\; X_i(\theta'_i, \theta_{-i}) \supseteq X_i(\theta) \text{ and } v(\theta'_i, X_i(\theta'_i, \theta_{-i})) = v(\theta'_i, X_i(\theta)), \\ &\forall \theta''_i \;\text{ s.t. }\; v(\theta''_i, X_i(\theta''_i, \theta_{-\{i,j\}})) = 0, \end{aligned}$$

$$\sum_{i \in \mathbb{N}} v(\theta'_i, X_i(\theta)) \geq \sum_{i \in \mathbb{N} \setminus \{j\}} v(\theta''_i, X_i(\theta_{-j})). \quad (3)$$

Note here that $\theta_{-\{i,j\}}$ denotes a type profile that excludes bidder i and j. This condition implies that for any set of participating bidders and any type profile, the revenue, which is the sum of the critical values of the bidders in an auction, weakly decreases when any bidder drops out from the auction.

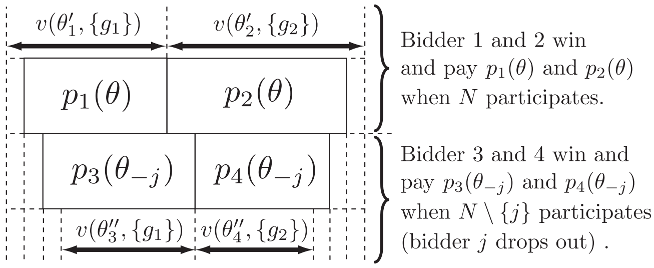


Fig. 1. Summation-monotonicity

The following is an intuitive explanation why summation-monotonicity holds for a strategy-proof and revenue monotone mechanism. Let us consider a combinatorial auction mechanism with two goods g_1 and g_2. Assume that it allocates g_1 to bidder 1 and g_2 to bidder 2 when a set of bidders N participates. On the other hand, also assume that it allocates g_1 to bidder 3 and g_2 to bidder 4 when bidder j drops out from the auction. The top two rectangles of Fig. 1 represent the total payments for bidder 1 and 2 and the bottoms for bidder 3 and 4. If the mechanism is revenue monotone, $p_1(\theta) + p_2(\theta) \geq p_3(\theta_{-j}) + p_4(\theta_{-j})$ holds.

The arrows at the top of Fig. 1 indicate the left-hand side of Eq. 3. θ'_1 means the minimal type where bidder 1 obtains g_1 or any superset, fixing other bidders' types than bidder 1. Under the mechanism, $v(\theta'_1, \{g_1\})$ must be greater than $p_1(\theta)$. Otherwise, bidder 1 has an incentive not to participate in the auction and individual rationality is violated. Similarly, θ'_2 means the minimal type where bidder 2 obtains g_2 or any superset, fixing other bidders' types than bidder 2 and $v(\theta'_2, \{g_2\})$ must be greater than $p_2(\theta)$.

The arrows at the bottom of Fig. 1 indicate the right-hand side of Eq. 3. θ''_3 means the maximal type where bidder 3 cannot obtain g_1; he obtains nothing, fixing other bidders' types than bidder 3. Under the mechanism, $v(\theta''_3, \{g_1\})$ must be smaller than $p_3(\theta_{-j})$. Otherwise, a bidder with θ''_3 as his true type has an incentive to pretend that his type is θ_3 to obtain g_1, and strategy-proofness is violated. Similarly, θ''_4 is the maximal type where bidder 4 cannot obtain g_2 and $v(\theta''_4, \{g_2\})$ must be smaller than $p_4(\theta_{-j})$.

From these facts, Lemma 1 proves that summation-monotonicity must hold for strategy-proof, revenue monotone mechanisms. Lemma 2 proves that, as long as summation-monotonicity and weak-monotonicity hold, we can find an appropriate payment rule p so that $p_1(\theta) + p_2(\theta) \geq p_3(\theta_{-j}) + p_4(\theta_{-j})$ holds. Thus, we derive the following theorem:

Theorem 1. *There exists an appropriate payment rule p such that a combinatorial auction mechanism $\mathcal{M}(X, p)$ is strategy-proof and revenue monotone if and only if X satisfies weak-monotonicity and summation-monotonicity.*

Lemma 1. *If a combinatorial auction mechanism $\mathcal{M}(X,p)$ is strategy-proof and revenue monotone, then the allocation rule X satisfies weak-monotonicity and summation-monotonicity.*

Proof. It was already proved in [2] that if $\mathcal{M}$ is strategy-proof, then X satisfies weak-monotonicity. To prove this lemma, it suffices to show that if $\mathcal{M}$ is strategy-proof and revenue monotone, then X satisfies summation-monotonicity.

Let W_N denote the set of winners when a set of bidders N participates, and let $W_{N\setminus\{j\}}$ denote the set of winners when bidder j drops out. Since $\mathcal{M}(X,p)$ is revenue monotone, from Eq. 1, we derive $\forall N \subseteq \mathbb{N}, \forall \theta \in \Theta_N, \forall j \in \mathbb{N}$,

$$\textstyle\sum_{i\in W_N} p_i(\theta) \geq \sum_{i\in W_{N\setminus\{j\}}} p_i(\theta_{-j}). \tag{4}$$

Each term $p_i(\theta)$ of the left-hand side must be smaller than the minimum bid in which bidder $i(\in N)$ still wins; otherwise $\mathcal{M}$ violates individual rationality. Thus we obtain

$$\forall i \in W_N, \forall \theta'_i \text{ s.t. } X_i(\theta'_i,\theta_{-i}) \supseteq X_i(\theta), v(\theta'_i, X_i(\theta'_i,\theta_{-i})) = v(\theta'_i, X_i(\theta)),$$
$$v(\theta'_i, X_i(\theta)) \geq p_i(\theta). \tag{5}$$

On the other hand, each term $p_i(\theta_{-j})$ of the right-hand side must be greater than the maximum bid in which bidder $i(\in N\setminus\{j\})$ loses; otherwise bidder i with type θ''_i has an incentive to pretend that his type is θ_i. Thus we obtain

$$\forall j, \forall i \in W_{N\setminus\{j\}}, \forall \theta''_i \text{ s.t. } v(\theta''_i, X_i(\theta''_i, \theta_{-\{i,j\}})) = 0,$$
$$p_i(\theta_{-j}) \geq v(\theta''_i, X_i(\theta_{-j})). \tag{6}$$

As a result, from Eqs. 4, 5, and 6, we obtain

$$\begin{aligned}\textstyle\sum_{i\in\mathbb{N}} v(\theta'_i, X_i(\theta)) &= \textstyle\sum_{i\in W_N} v(\theta'_i, X_i(\theta))\\ &\geq \textstyle\sum_{i\in W_N} p_i(\theta)\\ &\geq \textstyle\sum_{i\in W_{N\setminus\{j\}}} p_i(\theta_{-j})\\ &\geq \textstyle\sum_{i\in W_{N\setminus\{j\}}} v(\theta''_i, X_i(\theta_{-j}))\\ &= \textstyle\sum_{i\in\mathbb{N}\setminus\{j\}} v(\theta''_i, X_i(\theta_{-j}))\end{aligned}$$

and Eq. 3 holds. □

Lemma 2. *If an allocation rule X satisfies weak-monotonicity and summation-monotonicity, there exists an appropriate payment rule p such that a combinatorial auction mechanism $\mathcal{M}(X,p)$ is strategy-proof and revenue monotone.*

Proof. It was already proved in [2] that if X satisfies weak-monotonicity, there exists a payment rule p such that $\mathcal{M}(X,p)$ is strategy-proof. To prove this lemma, we show that if X satisfies weak-monotonicity and summation-monotonicity, we can choose p that also satisfies Eq. 1.

We are going to derive a contradiction by assuming Eq. 1 does not hold, although $\mathcal{M}(X,p)$ is strategy-proof and the allocation rule X satisfies weak-monotonicity and summation-monotonicity. More specifically, we assume that for any payment rule p with which $\mathcal{M}(X,p)$ is strategy-proof, $\forall p, \exists N \subseteq \mathbb{N}, \exists \theta, \exists j$, $\sum_{i \in \mathbb{N}} p_i(\theta) < \sum_{i \in \mathbb{N} \setminus \{j\}} p_i(\theta_{-j})$ holds. Now let us choose $\gamma (> 0)$ such that

$$\sum_{i \in \mathbb{N}} p_i(\theta) + \gamma = \sum_{i \in \mathbb{N} \setminus \{j\}} p_i(\theta_{-j}). \tag{7}$$

Then, choose a small enough value ϵ such that $0 < \epsilon < \frac{\gamma}{2|N|-1}$ holds. Also, let us define a type θ'_i for each $i \in N$ as

$$v(\theta'_i, B_i) = \begin{cases} p_i(\theta) + \epsilon & \text{if } B_i \supseteq X_i(\theta), \\ 0 & \text{otherwise.} \end{cases}$$

These types satisfy the preconditions of summation-monotonicity: $X_i(\theta'_i, \theta_{-i}) \supseteq X_i(\theta)$ and $v(\theta'_i, X_i(\theta'_i, \theta_{-i})) = v(\theta'_i, X_i(\theta))$.

Furthermore, let us define θ''_i for each $i \in N \setminus \{j\}$ as

$$v(\theta''_i, B_i) = \begin{cases} p_i(\theta_{-j}) - \epsilon & \text{if } B_i \supseteq X_i(\theta_{-j}), \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, these types satisfy the preconditions of summation-monotonicity: $v(\theta''_i, X_i(\theta''_i, \theta_{-\{j,i\}})) = 0$.

As a result, from Eq. 3, the following condition holds:

$$\sum_{i \in N} p_i(\theta) + |N| \cdot \epsilon \geq \sum_{i \in N \setminus \{j\}} p_i(\theta_{-j}) - (|N| - 1) \cdot \epsilon. \tag{8}$$

By substituting Eq. 7 into Eq. 8, we obtain $\gamma \leq (2|N| - 1) \cdot \epsilon$. This contradicts the assumption of $\epsilon < \frac{\gamma}{2|N|-1}$. □

4 Revenue Monotonicity and False-name-proofness

This section provides a theoretical consideration of a connection between revenue monotonicity and false-name-proofness.

As it was considered that there is a connection between revenue monotonicity and false-name-proofness [3, 4], the following case provides a common example where VCG is neither revenue monotone nor false-name-proof.

	$\{g_1\}$	$\{g_2\}$	$\{g_1, g_2\}$
bidder 1:	7	0	7
bidder 2:	0	0	8
bidder 3:	0	7	7

Let us consider a situation where bidder $1'$, who values 14 only on $\{g_1, g_2\}$, uses two identifiers 1 and 3. Since VCG allocates g_1 and g_2 to identifiers 1 and 3, respectively, bidder $1'$ obtains $\{g_1, g_2\}$ and pays 2. On the other hand, when only two bidders $1'$ and 2 participate in the auction, i.e., when bidder $1'$ does not use false identifiers, bidder $1'$ obtains $\{g_1, g_2\}$ and pays 8. As this example shows, increasing the number of participating bidders by or not by false identifiers can reduce the seller's revenue.

Therefore, a sub-additive allocation rule [8] apparently always coincides with a summation-monotone allocation rule, and vice versa. However, this is not true, although it is certain that sub-additivity looks quite similar to summation-monotonicity. Recall that summation-monotonicity implies that the sum of the critical values of bidders in an auction is guaranteed to weakly decrease when some of the bidders drop out from the auction. On the contrary, sub-additivity implies that the critical value of a bidder when he uses a single identifier is guaranteed to be smaller than or equal to that when he uses multiple identifiers.

4.1 Single-item auctions

We stated that, in general, revenue monotonicity cannot coexist with false-name-proofness. Nevertheless, in this subsection, we show that revenue monotonicity and false-name-proofness are equivalent for single-item auctions, assuming the following natural condition.

Assumption 1 *For any set of participating bidders N and for any bidder $j(\in N)$, if a mechanism allocates a good to a bidder when $N \setminus \{j\}$ participates, it always allocates the good to some bidder when N participates.*

We believe that introducing Assumption 1 is quite natural. From a seller's viewpoint, it is undesirable that a good is no longer allocated when more bidders join the auction. Under this assumption, the following theorem holds.

Theorem 2. *Under Assumption 1, a single-item auction mechanism is false-name-proof if and only if it is strategy-proof and revenue monotone.*

To show that this theorem holds, let us separately prove Lemmas 4 and 5. Before proving Lemma 4, we introduce the following lemma for strategy-proof, revenue monotone single-item auction mechanisms.

Lemma 3. *Let us consider strategy-proof, revenue monotone single-item auction mechanisms that sell good g. If bidder k wins when the set of bidders N participates, then bidder k also wins when any bidder $j(\neq k) \in N$ drops out from the auction.*

Proof. First, since we assume almost anonymous and strategy-proof mechanisms, a bidder can win a good only when his bid is higher than those of other participants. Formally, assume that bidder k wins when he reports θ_k. Then the left-hand side $v(\theta'_k, g)$ of Eq. 3 satisfies

$$v(\theta'_k, g) \leq v(\theta_k, g). \tag{9}$$

This means that the critical value to obtain the good is lower than $v(\theta_k, g)$.

Second, bidder k still has the largest valuation when bidder $j \neq k$ drops out. Now, assume that bidder k doesn't win in this situation. The critical value $cv_k^{N\setminus\{j\}}$ for k to win the good g is strictly greater than $v(\theta_k, g)$. Therefore, we can choose γ such that

$$v(\theta_k, g) = cv_k^{N\setminus\{j\}} - \gamma \tag{10}$$

holds. Let us also choose a small enough $\epsilon(0 < \epsilon < \gamma)$ and define a type θ''_k such that $v(\theta''_k, g) = cv_k^{N\setminus\{j\}} - \epsilon$. Since bidder k loses when he reports θ''_k, the type θ''_k satisfies $v(\theta''_k, X_k((\theta''_k, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\{k,j\}})) = 0$. Thus, from Eq. 10, we obtain

$$v(\theta_k, g) < v(\theta''_k, g). \tag{11}$$

Finally, from Eqs. 9 and 11, $v(\theta''_k, g) > v(\theta_k, g) \geq v(\theta'_k, g)$ holds and this violates summation-monotonicity. Thus, this mechanism contradicts the assumption that it is revenue monotone. □

Now, we are ready to prove Lemma 4.

Lemma 4. *Any strategy-proof, revenue monotone single-item auction mechanism satisfies false-name-proofness.*

Proof. To prove this lemma, we are going to derive a contradiction assuming that a single-item auction mechanism, which is strategy-proof and revenue monotone, is not false-name-proof. Specifically, we assume that for some $\theta_{-\phi(i)}$, there exists bidder i with type θ_i who can increase his profit using false identifiers $\phi(i)$:

$$\begin{aligned} &v(\theta_i, X_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)})) - p_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)}) \\ &< v(\theta_i, \textstyle\bigcup_{l \in \phi(i)} X_l(\theta_{\phi(i)}, \theta_{-\phi(i)})) - \sum_{l \in \phi(i)} p_l(\theta_{\phi(i)}, \theta_{-\phi(i)}). \end{aligned} \tag{12}$$

Since we consider the case where bidder i can increase his utility, the winner k must be in $\phi(i)$ when a set of bidders N participates. Let θ_k denote the type reported by bidder k. From Lemma 3, bidder i wins when he reports θ_k with only one identifier. We obtain

$$X_k(\theta_{\phi(i)}, \theta_{-\phi(i)}) = X_i((\theta_k, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)}). \tag{13}$$

Next, from Eq. 1, we obtain

$$p_i((\theta_k, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)}) \leq p_k(\theta_{\phi(i)}, \theta_{-\phi(i)}). \tag{14}$$

Furthermore, from strategy-proofness, we obtain

$$\begin{aligned} &v(\theta_i, X_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)})) - p_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)}) \\ &\geq v(\theta_i, X_i((\theta_k, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)})) - p_i((\theta_k, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)}). \end{aligned} \tag{15}$$

As a result, from Eqs. 13, 14, and 15,

$$\begin{aligned} &v(\theta_i, X_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)})) - p_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)}) \\ &\geq v(\theta_i, X_k(\theta_{\phi(i)}, \theta_{-\phi(i)})) - p_k(\theta_{\phi(i)}, \theta_{-\phi(i)}) \end{aligned}$$

holds. Thus, this contradicts Eq. 12. □

Lemma 5. *Under Assumption 1, any false-name-proof single-item auction mechanism satisfies strategy-proofness and revenue monotonicity.*

Proof. If every bidder uses only one identifier, false-name-proofness is equivalent to strategy-proofness. To prove this lemma, we show that if a mechanism is false-name-proof, then it is also revenue monotone. The model of revenue monotonicity assumes that the set of types of bidders who always participate is fixed. Thus, we can concentrate on the case where a bidder with θ_i uses s identifiers and submits $(\theta_i, \theta_{id_2}, \ldots, \theta_{id_s})$, in which he still submits his true type θ_i. Here, if there is no winner when a set of bidders $N \setminus \phi(i) \cup \{i\}$ participates, revenue monotonicity always holds, regardless of the allocation when N participates. Then, let us consider the case where the good is allocated to some bidder i when $N \setminus \phi(i) \cup \{i\}$ participates. From Assumption 1, if a good is allocated to bidder i, then the good is also allocated to bidder k when N participates.

Let us consider that bidder k belongs to $N \setminus \phi(i)$. From strategy-proofness, a bidder wins if he submits the highest bid. For the winning bidder i when $N \setminus \phi(i) \cup \{i\}$ participates,

$$v(\theta_i, g) \geq \max_{l \in N \setminus \phi(i)} v(\theta_l, g) \geq v(\theta_k, g). \tag{16}$$

Since the winning bid θ_i still exists when N participates, for the winning bidder $k \in N \setminus \phi(i)$ when N participates, $v(\theta_k, g) \geq v(\theta_i, g)$ holds. Here, since Eq. 16 is violated if $v(\theta_k, g) > v(\theta_i, g)$ holds, $v(\theta_k, g) = v(\theta_i, g)$ always holds. Accordingly, the payment when $N \setminus \phi(i) \cup \{i\}$ participates and i wins equals to that when N participates and k wins. In fact, i's payment $p_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)})$ is $v(\theta_k, g)$, while k's payment $p_k(\theta_{\phi(i)}, \theta_{-\phi(i)})$ is $v(\theta_i, g)$. Therefore, the mechanism satisfies revenue monotonicity.

On the other hand, let us consider that bidder k belongs to $\phi(i)$. In this case, we obtain $v(\theta_i, X_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)})) = v(\theta_i, X_k(\theta_{\phi(i)}, \theta_{-\phi(i)}))$. By substituting this into Eq. 2, $p_i((\theta_i, \mathbf{0}, \ldots, \mathbf{0}), \theta_{-\phi(i)}) \leq p_k(\theta_{\phi(i)}, \theta_{-\phi(i)})$ holds, so the mechanism satisfies revenue monotonicity. □

4.2 Combinatorial auctions

This subsection reveals that false-name-proofness and revenue monotonicity cannot coexist in combinatorial auction settings. To provide a clear proof, we introduce the following two assumptions.

Assumption 2 (Independence of irrelevant good) *Assume bidder i is winning all goods. If we add an additional good that is wanted only by bidder i, and his valuation for all goods is larger than or equal to some constant c, then he still wins all goods.*

The independence of irrelevant good (IIG) condition [10] is intuitively reasonable and is satisfied in almost all well-known mechanisms, in particular, in all existing false-name-proof mechanisms (to the best of our knowledge). This is true for a mechanism that uses predefined reserve prices assuming that c is large enough compared to the reserve price. Note that IIG is different from the typical Independence of Irrelevant Alternatives (IIA) conditions, which are often quite strong and apply to a wide variety of situations. Since IIG applies only to very specific situations, we consider it quite mild.

Assumption 3 *In a combinatorial auction mechanism, if there exists no bid on multiple goods, then for each good, the mechanism allocates the good to its highest bidder, as long as the highest bidder's valuation is larger than or equal to some constant c.*

This assumption is also quite natural and is satisfied in almost all well-known mechanisms.

Theorem 3. *Under Assumptions 2 and 3, there exists no combinatorial auction mechanism $\mathcal{M}$ that satisfies revenue monotonicity and false-name-proofness.*

Proof. Let us assume there exists a mechanism $\mathcal{M}$ that satisfies revenue monotonicity and false-name-proofness under Assumptions 2 and 3, and derive a contradiction. First, let us consider the following situation:

Case 1.

	$\{g_1\}$	$\{g_2\}$	$\{g_1, g_2\}$
bidder 1:	0	0	c
bidder 2:	$c-\epsilon$	0	$c-\epsilon$

Bidder 1 must win in Case 1. Let us assume bidder 1 is interested in $\{g_1\}$, rather than $\{g_1, g_2\}$. Then bidder 1 wins from Assumption 3. Then in Case 1, bidder 1 still wins from Assumption 2.

Next, we add another bidder 3:

Case 2.

	$\{g_1\}$	$\{g_2\}$	$\{g_1, g_2\}$
bidder 1:	0	0	c
bidder 2:	$c-\epsilon$	0	$c-\epsilon$
bidder 3:	0	$c/2-\epsilon$	$c/2-\epsilon$

We show that bidder 1 still wins in Case 2. If no bidder wins, then the revenue becomes 0. Thus, revenue monotonicity is violated. Also, if only bidder 3 wins, the revenue must be at most $c/2-\epsilon$ and revenue monotonicity is violated. Thus, let us assume bidder 2 and 3 win in Case 2. Then consider the following situation:

Case 3.

	$\{g_1\}$	$\{g_2\}$	$\{g_1, g_2\}$
bidder 1:	0	0	c
bidder 2:	$c-\epsilon$	0	$c-\epsilon$
bidder 3:	0	$c-\epsilon$	$c-\epsilon$

Bidder 3 must also win in Case 3, and the payment is at most $c/2-\epsilon$. Otherwise, bidder 3 has an incentive to under-declare his valuation to $c/2-\epsilon$ so that the situation becomes identical to Case 2. Also, since we assume the mechanism is almost anonymous across bidders and goods, bidder 2 also wins and pays at most $c/2-\epsilon$. However, in Case 1, bidder 2 can submit false-name bids and make the situation identical to Case 3 and obtain $\{g_1, g_2\}$ by paying $c-2\epsilon$. Thus, false-name-proofness is violated. In Case 2, bidder 1 must win and pays $c-\epsilon$.

Then, we add two more bidders 4 and 5:

Case 4.

	$\{g_1\}$	$\{g_2\}$	$\{g_1, g_2\}$
bidder 1:	0	0	c
bidder 2:	$c-\epsilon$	0	$c-\epsilon$
bidder 3:	0	$c/2-\epsilon$	$c/2-\epsilon$
bidder 4:	$c-2\epsilon$	0	$c-2\epsilon$
bidder 5:	0	$c/2-2\epsilon$	$c/2-2\epsilon$

Adding bidders 4 and 5 will not affect the outcome. Otherwise, revenue monotonicity is violated. Thus, in Case 4, bidder 1 still wins and pays $c-\epsilon$ for $\{g_1, g_2\}$.

Finally, let us consider the following situation:

Case 5.

	$\{g_1\}$	$\{g_2\}$	$\{g_1, g_2\}$
bidder 2:	$c-\epsilon$	0	$c-\epsilon$
bidder 3:	0	$c/2-\epsilon$	$c/2-\epsilon$
bidder 4:	$c-2\epsilon$	0	$c-2\epsilon$
bidder 5:	0	$c/2-2\epsilon$	$c/2-2\epsilon$

In Case 5, from Assumption 3, bidder 2 and 3 obtain g_1 and g_2, and pay $c-2\epsilon$ and $c/2-2\epsilon$, respectively. If bidder 1 joins, the situation becomes identical to Case 4. Then the revenue decreases from $\frac{3c}{2}-4\epsilon$ to $c-\epsilon$. Thus, revenue monotonicity is violated and this contradicts the assumption. □

Let us clarify the difference between our Theorem 3 and Rastegari *et al.*'s results [4]. They showed that there exists no deterministic false-name-proof combinatorial auction mechanism that is weakly maximal. They also proved that there exists no deterministic combinatorial auction mechanism that is strategy-proof, revenue monotone, and weakly maximal. Thus, revenue monotonicity and false-name-proofness cannot coexist assuming the mechanism is weakly maximal. On the other hand, we showed that revenue monotonicity and false-name-proofness cannot coexist assuming[1] the mechanism satisfies Assumptions 2 and 3.

Weak maximality and our two assumptions are independent; weak maximality does not mean the two assumptions hold, and the two assumptions do not mean weak maximality holds. We believe our assumptions are very mild, since

[1] To be precise, we also assume a mechanism is almost anonymous.

they apply only to very specific situations, while weak maximality applies to a wide variety of situations. Thus, it is more likely that a mechanism, which does not satisfy weak maximality, satisfies these two conditions.

5 Conclusions

This paper identified a simple condition called *summation-monotonicity* for characterizing strategy-proof and revenue monotone allocation rules. To the best of our knowledge, this is the first attempt to characterize revenue monotone allocation rules. In addition, our characterization enables us to examine the connections between revenue monotonicity and false-name-proofness. In a single-item auction, we showed that they are basically equivalent. Whereas, we also showed that they cannot coexist in combinatorial auctions under some minor conditions.

In future work, we hope to design a novel deterministic, revenue monotone combinatorial auction mechanism, since only a randomized mechanism has been proposed so far [11]. Furthermore, by utilizing our characterization, we hope to examine several theoretical properties of revenue monotone allocation rules, e.g., the upper bound on possible social surplus for revenue monotone mechanisms.

References

1. Myerson, R.B.: Optimal auction design. Mathematics of Operations Research **6**(1) (1981) 58–73
2. Lavi, R., Mu'alem, A., Nisan, N.: Towards a characterization of truthful combinatorial auctions. In: FOCS. (2003) 574–583
3. Rastegari, B., Condon, A., Leyton-Brown, K.: Revenue monotonicity in combinatorial auctions. In: AAAI. (2007) 122–127
4. Rastegari, B., Condon, A., Leyton-Brown, K.: Revenue monotonicity in deterministic, dominant-strategy combinatorial auctions. Technical report, UBC CS Technical Report TR-2009-09 (2009)
5. Yokoo, M., Sakurai, Y., Matsubara, S.: The effect of false-name bids in combinatorial auctions: New fraud in internet auctions. Games and Economic Behavior **46**(1) (2004) 174–188
6. Wagman, L., Conitzer, V.: Optimal false-name-proof voting rules with costly voting. In: AAAI. (2008) 190–195
7. Yokoo, M., Conitzer, V., Sandholm, T., Ohta, N., Iwasaki, A.: Coalitional games in open anonymous environments. In: AAAI. (2005) 509–515
8. Todo, T., Iwasaki, A., Yokoo, M., Sakurai, Y.: Characterizing false-name-proof allocation rules in combinatorial auctions. In: AAMAS. (2009) 265–272
9. Bikhchandani, S., Chatterji, S., Lavi, R., Mu'alem, A., Nisan, N., Sen, A.: Weak monotonicity characterizes deterministic dominant-strategy implementation. Econometrica **74**(4) (2006) 1109–1132
10. Iwasaki, A., Conitzer, V., Guo, M., Todo, T., Omori, Y., Sakurai, Y., Yokoo, M.: Worst-case efficiency ratio in false-name-proof combinatorial auction mechanisms. In: AAMAS. (2010) to appear.
11. Rastegari, B., Condon, A., Leyton-Brown, K.: Stepwise randomized combinatorial auctions achieve revenue monotonicity. In: SODA. (2009) 738–747

Network effects in double auction markets with automated traders

Kai Cai[1], Jinzhong Niu[1], and Simon Parsons[1,2]

[1] Department of Computer Science, Graduate Center, City University of New York,
365 5th Avenue New York, NY 10016, USA. {kcai,jniu}@gc.cuny.edu
[2] Department of Computer & Info. Science, Brooklyn College, City University of New York
2900 Bedford Avenue, Brooklyn, NY 11210, USA. parsons@sci.brooklyn.cuny.edu

Abstract. Many electronic markets are linked together into larger "network markets" where the links reflect constraints on traders. These constraints mean that a choice to trade in one market limits the trader's choice of other markets to use. This kind of network market is important because so many basic products, including gas, water, and electricity, are traded in such markets, and yet it has been little studied until now. This paper studies networks of double auction markets populated with automated traders, concentrating on the effects of different network topologies. We find that the topology has a significant effect on the equilibrium behavior of the set of markets.

1 Introduction

An *auction*, according to [8], is a market mechanism in which messages from traders include some price information — this information may be an offer to buy at a given price, in the case of a *bid*, or an offer to sell at a given price, in the case of an *ask* — and which gives priority to higher bids and lower asks. The rules of an auction determine, on the basis of the offers that have been made, the allocation of goods and money between traders. When well designed [13], auctions achieve desired economic outcomes like high *allocative efficiency* whilst being easy to implement. Auctions have been widely used in solving real-world resource allocation problems [16], in structuring stock or futures exchanges [8], and, despite the current recession, are the basis of a vast volume of trade in electronic markets.

There are many different kinds of auction. One of the most widely used kinds is the *double auction* (DA), in which both buyers and sellers are allowed to exchange offers simultaneously. Since double auctions allow dynamic pricing on both the supply side and the demand side of the marketplace, their study is of great importance, both to theoretical economists, and those seeking to implement real-world market places. The *continuous double auction* (CDA) is a DA in which traders make deals continuously throughout the auction. The CDA is one of the most common exchange institutions, and is in fact the primary institution for trading of equities, commodities and derivatives in markets such as the New York Stock Exchange (NYSE) and Chicago Mercantile Exchange (CME). Another common kind of double auction market is the *clearing-house* (CH) in which the market clears at a pre-specified time, allowing all traders to place offers before any matches are found. The CH is used, for example, to set stock prices at the beginning of trading on some exchange markets.

Our focus in this paper is on the behavior of multiple auctions for the same good. This interest is motivated by the fact that such situations are common in the real world. Company stock is frequently listed on several stock exchanges. US companies may be listed on both the NYSE, NASDAQ and, in the case of larger firms, non-US markets like the London Stock Exchange (LSE). Indian companies can be listed on both the National Stock Exchange (NSE) and the Bombay Stock Exchange (BSE). The interactions between such exchanges can be complex, like that when the newly created Singapore International Monetary Exchange (SIMEX) claimed much of the trade in index futures on Nikkei 225 from Japanese markets in the late 1980s [29], or when unfulfilled orders on the CME overflowed onto the NYSE during the global stock market crash of 1987 [17]. This kind of interaction between markets has not been widely studied, especially when the markets are populated by automated traders.

One multiple market scenario that is particularly interesting is that of *network markets*, markets in which individual markets are linked together into larger markets, where the links between markets reflect constraints on traders in the markets. Network markets are important because so many basic products, including gas [15], water, and electricity, are traded in such markets — the products proceed through a series of transactions at different locations from producer to final consumer, and the need to convey the product through a complex transportation network provides the constraints.

Our specific focus in this paper is to examine the differences between network markets with different topologies. We describe some experiments using network markets where the nodes in the network are double auction markets, traders can move between the markets, and the connections between markets are limitations on such moves. These experiments identify whether network topology has a significant effect on the steady state behavior of a set of connected markets and the speed with which the set of markets converges to that steady state. We see this work as a first step towards understanding the relationship between market topology and performance. Our long-term goal is to be able to use our understanding of this relationship to engineer network markets with appropriate properties.

2 Background

2.1 Double auctions

Double auctions have been extensively studied using agent-based methods. Gode and Sunder [10] were the first to use multi-agent simulations in this way, testing the hypothesis, suggested by [30], that the form of the market has more bearing on obtaining efficient allocation than the intelligence of traders in that market. [10] introduced a "zero-intelligence" trading strategy (denoted ZI-C) — which involves making offers at random under the constraint that they don't lead to loss-making trades — and showed that agents using this strategy could generate high efficiency. Indeed, such agents come close enough to the performance of human traders that Gode and Sunder claimed that trader intelligence is not necessary.

This position was attacked by Cliff [6], who showed that if supply and demand are asymmetric, the average transaction prices of ZI-C traders can vary significantly from the theoretical equilibrium. Cliff then introduced the *zero intelligence plus* (ZIP) trader, which uses a simple machine learning technique to decide what offers to make based on previous offers and the trades that have taken place. ZIP traders outperform ZI-C traders, achieving both higher efficiency and approaching equilibrium more closely across a wider range of market conditions, prompting Cliff to suggest that ZIP traders embodied the minimal intelligence required. A range of other trading algorithms has been proposed — including those that took part in the Santa Fe double auction tournament [28], the reinforcement learning *Roth-Erev* approach (RE) [26] and the *Gjer-stad-Dickhaut* approach (GD) [9] — and the performance of these algorithms has been evaluated under various market conditions. Despite the high performance of GD traders, research into automated trading mechanisms has continued.

This work on trading strategies is only one facet of the research on auctions. Gode and Sunder's results suggest that the structure of the auction mechanism plays an important role in determining the outcome of an auction, and this is further borne out by the work of [35] and [21], both of which show that the same set of trading strategies can have markedly different behaviors in different auction mechanisms. This leads us to anticipate that in a set of connected markets the way that the markets are connected will also have an effect on the behavior of the markets.

2.2 Methodology

The basis of our approach comes from Smith [31] via Gode and Sunder [10] and then Cliff [6]. We follow these authors in having all traders, whether human or machine, be chosen to be either buyers or sellers. No trader can both buy and sell in the same experiment. On any given day, each seller is given some number of indivisible goods that they are allowed to exchange for money, and is given a value for each good — the trader's *limit price* or *private value*. A typical restriction, which we adopt, is that no seller may sell a good for less than its private value. Buyers have a similar private value for a number of goods, but rather than goods, they are given an allocation of money which they may exchange for goods. No buyer is permitted to pay more than the private value for any good.

These conditions are what Smith [31] calls "conditions of normal supply and demand", the conditions in which the flow of goods through the market is at equilibrium and each day sellers bring to market the same goods that cost the same to produce, and buyers look to buy the same goods at the same price. The aim of our experiments is to identify what this equilibrium would be, and to allow us to find the equilibrium point — bearing in mind that there is a certain amount of learning going on that will take time to converge — we repeat the same trading conditions day after day, allowing our trading agents to recall the outcomes of trade on the previous day and trading multiple goods to speed convergence to equilibrium. Despite this, the slow convergence of the learning[3]

[3] Which we can attribute to the movement of traders between markets since we know that the trading strategies we use converge in a few days at most in single market experiments.

means that to get close to a steady state we run our experiments for 600 trading days under identical conditions with each day allowing for multiple rounds of trading.

Clearly, this is not a realistic model. There is no existing market in which the same set of traders will continue to trade with the same limit prices for more than a year of trading without some price shock altering prices or traders entering and leaving the market. The model is not intended to be realistic in this sense. The model is just intended to tell us about the steady state, and we know from the literature that introducing price shocks [9] and permitting traders to enter or leave the market [22] just slows convergence to the steady state.

Our justification for working with such a simplified model is that we see our work as fitting within the "class-of-models" approach, due to Sutton [18, 32]. According to Sutton, the aim of modelling economic systems is rarely to model a real market, but is to model an abstraction from a real market that captures the behavior of a whole class of markets — exactly those which are the instantiations of the abstract model. In this work we are trying to see what the steady state behavior is in all sets of competing markets, both those with price shocks and those without, both those in which traders enter and leave, and those that don't. To do that we look first at the most abstract market. We can take the results of our shock-free and fixed-trader experiments and use them to predict the results of removing these restrictions, and in the future we can investigate whether these predictions are true. This approach, of course, ties in with Rubinstein's suggestion [27] that economic modelling be used to help sharpen our economic intuitions about complex phenomena as well as being used to predict the behavior of real systems[4].

3 Experimental Setup

3.1 Software

To experiment with multiple markets, we used JCAT [11], the platform that supports the TAC Market Design Competition [5]. JCAT provides the ability to run multiple double auction markets populated by traders that use a variety of trading strategies. Auctions in JCAT follow the usual pattern for work on automated trading agents, running for a number of trading *days*, with each day being broken up into a series of *rounds*. A round is an opportunity for agents to make offers (shouts) to buy or sell, and we distinguish different days because at the beginning of a day, agents have their inventories replenished. As a result, every buyer can buy goods every day, and every seller can sell every day. Days are not identical because agents are aware of what happened on the previous day. Thus it is possible for traders to learn, over the course of several days, the optimal way to trade. In addition, JCAT allows traders to move between markets at the end of a day, and over the course of many days they learn which market they perform best in.

In JCAT there are no restrictions on the movement of traders. To study network effects, we extended JCAT to restrict the movement of traders. In particular, our extension

[4] [27] presents four purposes for economic modelling in general: to predict behavior; to guide decision-making by economic agents or policy-makers; to sharpen the intuition of economists when studying complex phenomena; and to establish linkages between theoretical economic concepts and everyday thinking.

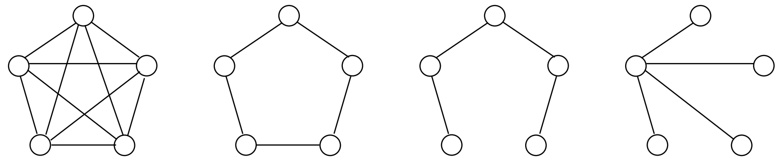


Fig. 1. The different topologies we consider. Each node is a market, each arc a connection between markets. (a) fully connected, (b) ring, (c) chain, (d) star.

allows us to specify which markets a given market is connected to. At the end of every day that a trader spends in that market, the trader has a choice of remaining in that market or moving to any of the markets to which there are connections. The decision mechanism employed by the traders to make this choice is discussed below.

In our experiments, market connections have four topologies (1) Fully connected. Each market is connected to every other market. (2) Ring. Each market is connected to exactly two other markets. This is what [36] calls a "local connected network". (3) Chain structure. All but two of the markets are connected to two other markets as in the ring. The remaining pair form the ends of the chain and are connected to exactly one market. (4) Star structure. One market is connected to every other market. There are no other connections between markets. This is the network topology studied in [25]. These topologies are illustrated in Fig. 1.

3.2 Traders

In JCAT markets, traders have two tasks. One is to decide how to make offers. The mechanism they use to do this is their *trading strategy*. The other task is to choose the market to make offers in. The mechanism for doing this is their *market selection strategy*. We studied markets in which all the traders used the same trading strategy, and considered two such strategies, Gode and Sunder's zero intelligence strategy ZI-C [10]; and Cliff's zero intelligence plus (ZIP) strategy [6]. The reason for picking the first of these is that given by [34], that since ZI-C is not making bids with any intelligence, any effects we see have to be a result of market structure, rather than a consequence of the trading strategy, and hence will be robust across markets inhabited by different kinds of trader. The reason for picking ZIP is that it is typical of the behavior of automated traders, rapidly converging to equilibrium in a single market.

In this work we use the standard market selection strategy used by JCAT. Traders treat the choice of market as an n-armed bandit problem that they solve using an ϵ-greedy exploration policy [33]. Using this approach, a trader chooses what it estimates to be the best available market, in terms of its average daily trading profit in that market on previous days, with probability $1 - \epsilon$, for some ϵ, and chooses one of the remaining available markets with equal probability otherwise. We choose ϵ to take a constant value of 0.1. Our previous work suggests that market selection behavior is rather insensitive to the parameters we choose here, and we choose ϵ to remain constant so that any convergence of traders to markets is due to traders picking markets that work for them rather than being forced by a reduction in their tendency to explore.

Each trader is permitted to buy or sell at most five units of goods per day, and each trader has a private value for these goods. Private values are set, just as in [6] to form perfect "staircase" supply and demand curves, with every buyer having a unique private value from the set $\{\$50, \$54, \$58 \ldots, \$246, \$250\}$. Sellers are allocated values in the same way. A given trader has the same private value for all goods that it trades throughout the entire experiment. All of our experiments used 100 traders, divided into 50 buyers and 50 sellers. Initially they are equally distributed between the markets, and subsequently use their market selection strategy to pick the market to operate in.

3.3 Markets

While JCAT allows us to charge traders in a variety of ways, we used just two kinds of charge in the work reported here:

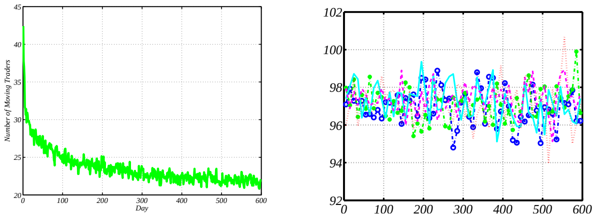
- Registration fees, charges made by the market for entering the market. We set this to a low constant value ($0.5) for every market following [23] which suggests that such a fee is effective in motivating extra-marginal traders to move between markets thus preventing their inertia from distorting results.
- Profit fees, charges made by the market on the bid/ask spread of any transactions they execute. The name arose because the bid/ask spread is the transaction surplus, and with the $k = 0.5$ rule that is usually used in JCAT for allocating the surplus, the amount charged by this fee is thus directly related to the profit realized by both agents.

Unlike previous work that used JCAT to investigate multiple market scenarios [22], we used a simple, non-adaptive scheme for the profit fee, placing a 5% profit charge on all markets. In all of our experiments we run five markets connected as described above, and we used both CDA and CH markets, both of which are provided in JCAT.

3.4 Hypotheses

The aim of this work was to investigate the effect on market performance of different topological connections between markets. In the context of the double auction markets that we consider, these connections might reflect a number of different constraints. For example, they might reflect the physical layout of market makers on a trading floor, or they might reflect affiliations between electronic markets, or they might reflect the relationship between the time-zones in which different markets operate.

In any case, we would expect that, as in [12], the topology of the relationships to have an effect on market behavior. In a model where traders move between markets, we would expect that placing different restrictions on movement between markets would lead to differences in the ease with which traders can explore the space of markets and then reach their preferred market, affecting the time it takes the set of markets to reach their steady state. In addition, we might expect that these different restrictions might lead to the steady state favoring some markets over other. These considerations give us two hypotheses that we will test:



(a) Traders moving (b) Transaction price

Fig. 2. How the markets change over time. (a) shows the total number of traders that move at the end of a given trading day, (b) shows the average transaction price each day for a set of five fully connected CDA markets with ZIP traders. The x-axis gives the trading day, the y-axis gives (a) the number of traders, (b) the transaction price.

1. The topology of the network market will affect the speed with which the set of markets reaches its steady-state configuration; and
2. The topology of the network will have a significant effect on the steady state configuration of the set of markets.

Note that in discussing these hypotheses, we find it helpful to distinguish the fact that some of the topologies we consider — the star and chain — are *asymmetric* in the sense that traders in some markets are more restricted in the markets that they can move to as opposed to the *symmetric* ring and fully-connected markets where, in terms of connections, all markets are equal.

3.5 Experiments

To test these hypotheses, we ran experiments that tested all the different combinations discussed above. That is we ran experiments for CH and CDA markets using each of the four different topologies, both the trading strategies described above and both the market selection strategies. Each of these experiments was run for 600 trading days, with each day being split into 50 0.5-second-long rounds. We repeated each experiment 50 times and the results that we give are averages across those 50 runs.

In order to assess the effect of the different topologies on the convergence of the markets, we looked at the number of traders that moved each day. The market selection strategy picks a random market with probability ϵ, so there will always be some movement of traders, but we would expect to see the number of traders decreasing from an initial high to a steady state, and the speed with which the steady state is reached is one way to measure how quickly the system of traders and markets converges.

To identify any differences between the steady state configurations of different market topologies we looked at two things — the number of traders in each market, and the efficiency of each market. The number of traders in each market gives us some idea of the preference that traders have for markets, and any time that there is an uneven distribution it is an indication that from the traders' point of view differences in market

ZIC CDA	Fully connected	141.25		ZIP CDA	Fully connected	142.91
	Ring	107.48			Ring	108.90
	Chain	93.47			Chain	95.61
	Star	93.34			Star	98.44
CH	Fully connected	184.56		CH	Fully connected	155.75
	Ring	143.95			Ring	120.73
	Chain	125.43			Chain	109.11
	Star	127.68			Star	113.43

Table 1. The average number of traders moving each day for the different topologies.

topologies have an effect. Efficiency, of course, is a standard measure of market behavior, and will indicate whether differences in the market topologies have an effect on the performance of the set of markets as a whole.

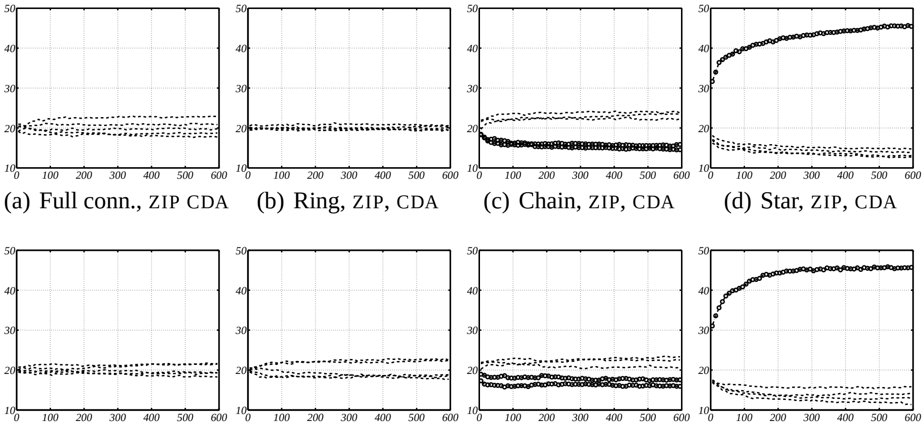
4 Results

4.1 Speed of convergence

When we look at the movement of traders between markets it is clear from Fig. 2 (a) that the markets make an exponential approach to the steady state (these results are for ZIP traders and fully connected arkets, but the results for other experiments are similar). This is despite the fact that the average transaction price in each market is, like that shown in Fig. 2 (b), far from steady[5]. Since, as described above, the market selection strategies we are using will mean that we always have some number of traders still moving at the end of each trading day, we can't determine equilibrium by looking for the point at which all traders stop moving. Instead we need to find a way to estimate the speed of convergence.

To do this we borrowed from the usual measure of the convergence of a market to equilibrium [31]. To compute this measure, Smith's *alpha* as it is known, we compute the average deviation between the price of each transaction and the equilibrium price suggested by theory. Here, we look at the number of traders moving each day and compute the average difference from the number we would expect if the only cause of trader movement was the ϵ in the market selection strategy (which would mean that, on average, 10% of the traders would move each day). Markets that are faster to converge to the steady state will have lower values of this difference. These results are shown in Table 1 and show that there is a clear difference between the speeds with which the different tologies converge. In particular, the asymmetric topologies converge much faster than the symmetric topologies.

[5] Because the transaction price is a function of the traders in a market (in particular it depends on both their private values and when and how they choose to bid), it changes as traders move between markets. Since the reward gained by traders is a function of the transaction price the dynamics are more complex than those of a set of n-armed bandit learners converging to static rewards.



(a) Full conn., ZIP CDA (b) Ring, ZIP, CDA (c) Chain, ZIP, CDA (d) Star, ZIP, CDA

(e) Full conn., ZIC CDA (f) Ring, ZIC, CDA (g) Chain, ZIC, CDA (h) Star, ZIC, CDA

Fig. 3. The number of traders in multiple connected CDA markets with different connection topologies on each trading day. The traders in (a)–(d) use the ZIP strategy, those in (e)–(h) use the ZIC strategy. The x axis gives the trading days, the y axis the number of traders in each of the five markets. In the chain markets, the dark lines give the numbers for the markets at the end of the chain, and for the star markets, the dark line gives the numbers for the market at the center. All other markets are marked with dashed lines.

4.2 Trader distribution

To examine the steady-state for differences due to connection topology, we looked at the number of traders in each market. Figure 3 shows this for each day of the experiment for both ZIC and ZIP traders in CDA markets (the other experiments give very similar results). The graphs in the figure show that the distribution of traders in fully-connected (Fig. 3(a), Fig. 3(e)) and ring (Fig. 3(b), Fig. 3(f)) markets is pretty uniform.

Chain markets, however, don't have the same symmetry, and this shows up in the distribution of traders. As Fig. 3(c) and Fig. 3(g) show, markets at the end of the chain end up with fewer traders than the markets in the middle of the chain. The effect of the loss of symmetry is even more marked in star markets, Here, as shown in Figures 3(d) and 3(h) the hub market in the star collects many more traders than the otherwise identical markets that are connected to it.

The graphs of Fig. 3 don't make it easy to decide what differences are significant so we show the actual trader numbers after the 600th trading day (that is at the end of the experiment) in Table 2. This includes the results of all the experiments on star and chain markets, not just those from Fig. 3 (the ones from the figure are in the first and third rows of the table). In the chain markets, the markets at the ends of the chain are M0 and M4. T-tests reveal that the numbers of traders in these markets are significantly different from the numbers of traders in markets M1, M2 and M3 at the 95% level. This holds for both CDA and CH markets whether the traders are ZI-C or ZIP. In the star markets, the market at the hub of the star is M0. T-tests show that the number of traders in this market is significantly different from that in all other markets at the 95% level again for both CDA and CH markets for ZI-C and ZIP traders.

			Star					Chain				
CDA	ZIC	No. of traders	43.67	13.65	15.82	14.14	12.72	16.24	22.74	20.88	22.21	17.93
		Stdev.	11.89	7.88	8.25	8.38	7.23	6.63	8.86	9.86	9.67	7.45
	ZIP	No. of traders	42.50	13.90	13.57	15.42	14.61	16.10	22.22	22.66	23.72	15.31
		Stdev.	9.08	5.13	5.19	5.57	4.76	4.91	5.52	7.08	5.89	4.45
CH	ZIC	No. of traders	44.71	13.16	13.83	14.40	13.89	16.45	23.66	20.33	22.28	17.28
		Stdev.	5.70	2.68	3.01	3.03	3.90	4.82	6.67	5.78	6.03	4.67
	ZIP	No. of traders	47.41	12.14	12.92	13.60	13.93	15.50	23.02	22.10	24.76	14.63
		Stdev.	8.44	3.32	3.07	4.40	4.58	4.80	6.32	7.01	6.31	4.66

Table 2. The number of traders in each market for star and chain configurations for both market selection strategies. In the star configuration, M0 is the hub, the market at the center. In the chain markets, markets M0 and M4 are the markets at the end of the chain. All markets make the same charges. In the star configuration the number of traders in M0 is significantly greater than that in all the other markets with 95% confidence in all cases and in the chain markets the number of traders in M0 and M4 is significantly smaller than in all the other markets with 95% confidence in all cases.

			Chain	Ring	Star	F.C.
ZIC	CDA	Efficiency	95.49	95.42	95.75	95.38
		Stdev.	0.30	0.25	0.22	0.16
	CH	Efficiency	96.61	96.51	96.81	96.56
		Stdev.	0.25	0.19	0.15	0.13

			Chain	Ring	Star	F.C.
ZIP	CDA	Efficiency	95.50	95.33	95.68	95.05
		Stdev.	0.24	0.19	0.22	0.17
	CH	Efficiency	96.86	96.77	96.96	96.54
		Stdev.	0.24	0.17	0.19	0.15

Table 3. The global efficiencies of sets of market with different connection topologies from left to right, chain, ring, star and fully connected networks. The table gives results for markets using both ZI-C and ZIP traders, and for both CDA and CH markets.

4.3 Allocative efficiency

The final results to consider are those in Table 3 which measures the allocative efficiency of sets of markets of different topologies. In particular what they measure is what we call "global efficiency", the ratio of the sum of profit made in all of the markets to the equilibrium profit that would be made in a market containing all the traders.

Pairwise t-tests on the efficiency values in Table 3 reveal that that there are differences between the efficiencies obtained with different configurations that are significant at the 95% level. In all the experiments the symmetric markets are significantly less efficient than the asymmetric markets. In all of the experiments except the CH with ZIC traders, fully-connected markets are less efficient than ring markets, ring markets are less efficient than chain markets, and chain markets are less efficient than star markets — all of these differences being significant at the 95% level. A possible explanation for this may be the fact that the asymmetric markets tend to concentrate traders in particular markets but results from our prior work [22] (on the effect of allowing traders to move in fully connected markets) suggests that such effects are only a partial explanation.

Note that the efficiency results we report for ZIP traders are somewhat lower than are reported for such traders in single markets (and are lower than the results we have obtained for the same implementation of ZIP in a single market, results which are similar to those seen in the literature). We believe that there are a couple of reasons for this. First, we are computing efficiency as the surplus obtained divided by the surplus that would be obtained were all the traders in one market and that market traded at theoretical equilibrium. It is easy to see that it is possible to match traders in such a way that individual markets are efficient, but the combined surplus will fall below what would be possible if all traders were in one market and that is what we believe is happening here. (ZIP achieves higher efficiency when the efficiency is computed in a more conventional fashion.). Second, traders are constantly moving between markets, which means that the equilibrium point of all the markets is constantly changing (recall the transaction prices of Fig. 2 (b)). We know from [6] that ZIP takes several trading days to identify market equilibrium, and since this is changing every day, ZIP is always playing catch-up. Naturally this will mean it is less than completely efficient. (When traders are constrained not to move, the efficiency of ZIP improves.)

4.4 Discussion

The aim of this work was to test the hypotheses that:

1. The topology of the network market will affect the speed with which the set of markets reaches its steady-state configuration; and
2. The topology of the network will have a significant effect on the steady state configuration of the set of markets.

The results in Table 1 suggest that the first of these hypotheses is correct — for most of the experiments that we carried out, the time we estimate it takes the set of markets to converge varies considerably from topology to topology.

To address the second hypothesis, we measured both the number of traders in each market and the overall efficiency of the set of markets. When we looked at the number of traders (Table 2), it was clear that many more traders congregated in the central market of the star configuration and many fewer traders choose the end markets of the chain configuration, and pairwise t-tests confirmed that the differences are statistically significant. This suggests that the second hypothesis is correct. This suggestion is supported by looking at the efficiency of different sets of markets (Table 3) where we find that sets of markets with different topologies have significantly different efficiencies.

5 Related work

While network markets have not been studied in the same detail as single markets, there is a growing body of work to consider. [25], for example, describes a study of a three-node star network with a uniform-price double auction at each node. The same authors [24] report experiments using a 9-node gas network that, in addition to buyers and sellers, also includes pipeline owners, and in [15] study another small gas market. A further small network model, including just two markets, is the basis of the study in [4] into the effects of cheating (that is, either not paying for goods, or failing to deliver goods that have been paid for) and [7] investigates how a 6-node railway network responds to two different pricing mechanisms. While these markets are similar to those in our study, the investigations all dealt with markets with human traders.

Agent-based methods were used by [3] to examine the effects of linked markets on financial crises, while [19, 20] consider the behavior of supply chains[6]. This work all studies smaller sets of markets than we have considered. The agent-based studies in [2] and [36] are larger but consider a set of connection topologies that overlap with, but does not contain, the set we consider. Both [2] and [36] deal with networks equivalent to our ring (their term is "local") as well as small-world networks, which we don't consider. Neither looks st chain or star topologies, the most interesting of the topologies we looked at, and neither study considers traders that move between markets.

The most closely related research we know of is [12], [37] and [14]. Judd and Kearns [12] describe experiments with human traders that clearly show that restrictions on who is allowed to trade with who — restrictions that are somewhat different from those imposed in our work — have a significant effect on market clearing performance. Wilhite [37], though mainly concentrating on results from network versions of the Prisoner's dilemma, describes agent-based experiments in the same kind of scenario as studied in [12] with similar results. Ladley and Bullock [14] looked at networked markets of ZIP traders and showed that differences in topology affected an agent's ability to make a profit. Like the results reported here, all of this work helps us to understand different aspects of the effect of network topology on market performance.

6 Conclusions

This paper has examined the effect of different connection topologies on network markets in which the constituent markets are double auctions and the connections denote the allowed movements of traders between markets. This work is the first systematic study of the effects of network topology on a set of double auction markets.

Traders in our experiments used either ZI-C or ZIP strategies, and markets were either CHs or CDAs. We looked at the behavior of four different topologies — fully connected, ring, chain and star — and considered the speed with which markets converge to a steady state, the distribution of traders across markets in the steady state, and the overall allocative efficiency in the steady state. We found that for all of these aspects, the connection topology can have a significant effect. In particular, the asymmetric topologies, chain and star, lead to an unequal distribution of traders, and in most cases an overal increase in efficiency of the markets.

Our main conclusion that topology affects steady state behavior is in line with previous work on network markets [12, 37]. In addition, since our results are consistent across different trading strategies (including the minimally rational ZI-C) and different market selection strategies, we believe that they will prove to be robust across other variants of our experimental scenario. With this in mind, we are currently working to analyze the performance of network markets with different topologies — in particular small-world, random and scale-free topologies — and to handle larger sets of markets than we considered here.

[6] The TAC supply chain competition also studies supply chains, but comes at it from the perspective of individual traders rather than from the perspective of overall market performance.

Acknowledgments This work was partially funded by NSF IIS-0329037 , and EP-SRC GR/T10657/01. We are grateful for use of the computational facility at the CUNY Graduate Center and to the reviewers whose comments helped us to make many improvements to the paper

References

1. K. Cai, J. Niu, and S. Parsons. On the economic effects of competition between double auction markets. In *Proceedings of the 10th Workshop on Agent-Mediated Electronic Commerce*, Estoril, Portugal, 2008.
2. A. Cassar. Coordination and cooperation in local, random and small world networks: Experimental evidence. *Games and Economic Behavior*, 58(2):209–230, February 2007.
3. A. Cassar and N. Duffy. Contagion of financial crises under local and global markets. In F. Luna and A. Perrone, editors, *Agent-based Methods in Economics and Finance: Simulations in Swarm*. Kluwer, New York, NY, 2001.
4. A. Cassar, D. Friedman, and P. H. Schneider. Cheating in markets: A laboratory experiment. Working paper, Learning and Experimental Economics Projects of Santa Cruz (LEEPS) Laboratory, 2006.
5. `http://www.marketbasedcontrol.com/`.
6. D. Cliff. Minimal-intelligence agents for bargaining behaviours in market-based environments. Technical Report HPL-97-91, Hewlett-Packard Research Laboratories, 1997.
7. J. Cox, T. Offerman, M. Olson, and A. Schram. Competition For versus On the Rails: A laboratory experiment. *International Economic Review*, 43(3):709–736, August 2002.
8. D. Friedman. The double auction institution: A survey. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories and Evidence*, pages 3–25. Perseus Publishing, Cambridge, MA, 1993.
9. S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22:1–29, 1998.
10. D. K. Gode and S. Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, 1993.
11. `http://jcat.sourceforge.net/`.
12. J. S. Judd and M. Kearns. Behavioral experiments in networked trade. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 150–159, Chicago, IL, July 2008.
13. P. Klemperer. How (not) to run auctions: The European 3G telecom auctions. *European Economic Review*, 46:829–845, 2002.
14. D. Ladley and S. Bullock. Who to listen to: Exploiting information quality in a ZIP-agent market. In *Proceedings of the Workshop on Trading Agent Design and Analysis*, 2005.
15. K. A. McCabe, S. J. Rassenti, and V. L.Smith. Auction design for composite goods. *Journal of Economic Behavior and Organization*, 14:127–149, 1990.
16. J. McMillan. *Reinventing the Bazaar: A Natural History of Markets*. W. W. Norton & Company, 2003.
17. M. H. Miller, J. D. Hawke, B. Malkiel, and M. Scholes. Findings of the Committee of Inquiry Examining the Events Surrounding October 19, 1987. Technical report, Chicago Mercantile Exchange, Spring 1988.
18. M. Morgan. Symposium on *Marshall's Tendencies*: 1 how models help economists to know. *Economics and Philosopy*, 18:5–16, 2002.
19. T. Moyaux and P. McBurney. Modeling a supply chain as a network of markets. In *Proceedings of the IEEE International Conference on Service Systems and Service Managament*, Troyes, France, 2006.

20. T. Moyaux and P. McBurney. Reduction of the bullwhip effect in supply chains through speculation. In *Proceedings of the Symposium on Artificial Economics*, Aalborg, Denmark, 2006.
21. J. Niu, K. Cai, S. Parsons, and E. Sklar. Reducing price fluctuation in continuous double auctions through pricing policy and shout improvement. In *Proceedings of the 5th International Conference on Autonomous Agents and Multi-Agent Systems*, Hakodate, Japan, 2006.
22. J. Niu, K. Cai, S. Parsons, and E. Sklar. Some preliminary results on the dynamic behavior of traders in multiple markets. In *Proceedings of the Workshop on Trading Agent Design and Analysis*, 2007.
23. J. Niu, P. McBurney, E. Gerding, and S. Parsons. Characterizing effective auction mechanisms: Insights from the 2007 TAC market design competition. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems*, Estoril, Portugal, 2008.
24. S. Rassenti, S. Reynolds, and V. Smith. Cotenancy and competition in an experimental auction market for natural gas pipeline networks. *Economic Theory*, 4(1):41–65, 1994.
25. S. Rassenti and V. Smith. Deregulating electric power: Market design issues and experiments in designing competitive electricity markets. In *International Series in Operations Research and Management Science*, volume 13, pages 105–120. Kluwer, Dordrecht, 1998.
26. A. E. Roth and I. Erev. Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term. *Games and Economic Behavior*, 8:164–212, 1995.
27. A. Rubinstein. *Modeling Bounded Rationality*. Zeuthen Lecture Book Series. MIT Press, Cambridge, MA, USA, 1998.
28. J. Rust, J. H. Miller, and R. Palmer. Characterizing effective trading strategies. *Journal of Economic Dynamics and Control*, 18:61–96, 1994.
29. A. Shah and S. Thomas. David and Goliath: Displacing a primary market. *Global Financial Markets*, 1(1):14–21, 18th June 2000.
30. V. L. Smith. An experimental study of competitive market behaviour. *Journal of Political Economy*, 70(2):111–137, April 1962.
31. V. L. Smith. An experimental study of competitive market behaviour. *The Journal of Political Economy*, 70(2):111–137, April 1962.
32. J. Sutton. *Marshall's Tendencies: What Can Economists Know?* MIT Press, Cambridge, MA, 2000.
33. R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, 1998.
34. V. Walia, A. Byde, and D. Cliff. Evolving market design in zero-intelligence trader markets. Technical Report HPL-2002-290, Hewlett-Packard Research Laboratories, Bristol, England, 2003.
35. W. Walsh, R. Das, G. Tesauro, and J. O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In *Proceedings of Workshop on Game-Theoretic and Decision-Theoretic Agents*, 2002.
36. A. Wilhite. Bilateral trade and 'small-world' networks. *Computational Economics*, 18:49–64, 2001.
37. A. Wilhite. Economic activity on fixed networks. In L. Tesfatsion and K. L. Judd, editors, *Handbook of Computational Economics*, volume 2, pages 1013–1045. Elsevier B. V., 2006.

Setting Fees in Competing Double Auction Marketplaces: An Equilibrium Analysis

Bing Shi, Enrico H. Gerding, Perukrishnen Vytelingum and Nicholas R. Jennings

School of Electronics and Computer Science,
University of Southampton, Southampton, UK
{bs07r,eg,pv,nrj}@ecs.soton.ac.uk

Abstract. *In this paper, we analyse competing double auction marketplaces that vie for traders and need to set appropriate fees to make a profit. Specifically, we show how competing marketplaces should set their fees by analysing the equilibrium behaviour of two competing marketplaces. In doing so, we focus on two different types of market fees: registration fees charged to traders when they enter the marketplace, and profit fees charged to traders when they make transactions. In more detail, given the market fees, we first derive equations to calculate the marketplaces' expected profits. Then we analyse the equilibrium charging behaviour of marketplaces in two different cases: where competing marketplaces can only charge the same type of fees and where competing marketplaces can charge different types of fees. This analysis provides insights which can be used to guide the charging behaviour of competing marketplaces. We also analyse whether two marketplaces can co-exist in equilibrium. We find that, when both marketplaces are limited to charging the same type of fees, traders will eventually converge to one marketplace. However, when different types of fees are allowed, traders may converge to different marketplaces (i.e. multiple marketplaces can co-exist).*

Key words: Competing Marketplaces, Nash Equilibrium, Evolutionary Game Theory, Double Auctions

1 Introduction

Financial exchanges, in which securities, futures, stocks and commodities can be traded, are becoming ever more prevalent. Now, many of these adopt the double auction market mechanism which is a particular type of two-sided market with multiple buyers (one side) and multiple sellers (the other side). Specifically, in such a mechanism, traders can submit offers at any time in a specified trading round, and can be matched by the marketplace at a specified time. The advantages of this mechanism are that traders can enter the marketplace at any time and they can trade multiple homogeneous or heterogeneous items in one place without travelling around several marketplaces. In addition, this mechanism provides high allocative efficiency [3]. These benefits have led many electronic marketplaces to also use this format. For example, Google owns DoubleClick Ad Exchange[1], which is a real-time double auction marketplace enabling large online ad publishers, on one side, and ad networks and agencies, on the other side, to buy and sell advertising space. However, because of the globalised economy, these marketplaces do not exist in isolation. Thus they compete against each other to attract traders and make profits by charging fees to traders. For example, stock exchanges compete to attract companies to list their stocks in their marketplaces and make profits by charging listing fees to these companies, and Google competes against other ad exchanges, such as Microsoft's AdECN and Yahoo!'s Right Media. However, there exists a conflict between attracting traders and making profits for the competing marketplace, since when the fees are increased, traders will leave the marketplace and eventually cause a decrease of profits for this marketplace. Against this background, in this paper, we analyse the equilibrium behaviour of competing marketplaces in terms of charging fees to traders, which can provide insights to guide how competing marketplaces should set their fees.

In more detail, there are two key issues in the research of competing marketplaces. The first is how traders should choose which marketplace to go to. Then, given the traders' market selection strategies, the second issue is how competing marketplaces should set their fees to maximise profits while at the same time maintaining market share at a good level in order to ensure profits in the long term. Now we have analysed the first issue in our previous work [1], so here we focus on how competing marketplaces set their fees.

Related to our work, a number of theoretical models have been proposed to analyse two-sided competing marketplaces (e.g. [2, 5, 6, 8]). However, these works do not consider auction mechanisms to match traders and set transaction prices. Instead, they assume that traders only select marketplaces based on the *number* of other traders in the marketplace. In doing so, they assume that all traders are homogeneous (i.e. have the same preferences), and the marketplace has complete information about the preferences (also called the types) of traders. In real-world auction marketplaces, however, traders are usually heterogeneous and they are likely to have privately known preferences. Moreover, transaction prices are usually set according to the marketplace's pricing policy, which is affected by current demand and supply. Also related to our work is the Market Design Competition (CAT), an annual competition and part of the Trading Agent Competition (TAC) which was introduced to promote research in the area of competing double auctions [4]. However, the work related to CAT is still largely empirical in nature. To tackle these limitations, in our previous work [1], we proposed a novel game-theoretic framework to analyse the competition between double auction marketplaces from a theoretical perspective, which assumes that traders are heterogeneous with different types[2] and the type of each specific trader is not known to the other traders and marketplaces. Based on this framework, we analysed the traders' Nash equilibrium (NEQ) market

[1] http://www.doubleclick.com/

[2] The types of buyers and sellers represent the buyers' limit prices and the sellers' cost prices respectively. The limit price is the highest price that the buyer is willing to buy the item for, and the cost price is the lowest price that the seller is willing to sell the item for.

selection strategies. Moreover, using evolutionary game theory (EGT) [10], we analysed how traders dynamically change their market selection strategies and determined which strategies traders eventually converge to.

In this paper, we extend this work by analysing how double auction marketplaces set their fees to make profits in a multiple competing marketplaces environment. In this environment, if the marketplace charges higher fees than its opponents, then it may make more profits in the short-term. However, eventually traders will choose to leave this marketplace and choose to migrate to the cheaper marketplace. This will therefore result in a decrease of profits. Thus the competing marketplace should charge appropriate fees to make profit and maintain market share given its opponents' fees. In this paper, we will analyse this pattern by considering the equilibrium charging behaviour between marketplaces. In reality, two types of fees are usually charged to traders. One is the *ex-ante* fee charged to traders before they make transactions. The other is the *ex-post* fee charged conditional on traders making a transaction [5]. Specifically, in our analysis, we consider the registration fee charged to traders when they enter the marketplace, and the profit fee charged to traders when they make transactions as a typical example of *ex-ante* and *ex-post* fees respectively. Furthermore, we analyse under what conditions several competing double auction marketplaces can co-exist when traders converge to their equilibrium market selection strategies. That is, we are interested in analysing whether competition can be maintained, or whether the marketplaces collapse to a monopoly setting where all traders move to one marketplace. This is important since competition drives efficiency and offers more and better choices to traders. In previous work, we found when competing marketplaces are only allowed to charge the profit fees, traders eventually converge to one marketplace. In [2] which considers the competition of two-sided marketplaces, researchers claim that when two competing marketplaces differentiate themselves from each other, they may co-exist. In this paper, we will analyse the co-existing issue in the context of competing double auction marketplaces.

In particular, the contributions of this paper are as follows. First, we provide a novel approach to estimate marketplaces' expected profits given the equilibrium strategies of the traders and the fees charged by other marketplaces. Second, based on the estimated expected profits of marketplaces, we are the first to analyse the equilibrium charging behaviour of competing double auction marketplaces. Finally, we show that in our framework, when different types of fees are allowed, traders may converge to different marketplaces, i.e. competing marketplaces can co-exist.

The structure of the paper is as follows. In Section 2, we briefly describe the general framework from [1] for analysing competing double auction marketplaces. In Section 3, we provide an approach to estimate marketplaces' expected profits. Then in Section 4, we analyse the equilibrium for two settings: when competing marketplaces charge the same type of fees, and with different type of fees. We also investigate under what under conditions competing marketplaces co-exist. Finally, we conclude this paper in Section 5.

2 General Framework

In this section, we briefly introduce the framework developed in [1]. We start by introducing basic notations of our framework. Then we introduce the marketplaces and their policies. Finally, we describe the market selection strategies in detail and give a general equation for a trader's expected utility.

2.1 Preliminaries

We assume that there are a set of buyers, $\mathcal{B} = \{1, 2, ...B\}$, and a set of sellers, $\mathcal{S} = \{1, 2, ...S\}$. Each buyer and seller has a type, which is denoted as θ^b and θ^s respectively. We assume that types of all buyers are independently drawn from the same cumulative distribution function F^b, with support $[\underline{l}, \bar{l}]$, and the types of all sellers are independently drawn from the cumulative distribution function F^s, with support $[\underline{c}, \bar{c}]$. The distributions F^b and F^s are assumed to be common knowledge and differentiable. The probability density functions are f^b and f^s respectively. In our framework, the type of each specific trader is not known to the other traders and marketplaces, and only the type distribution functions are public. In addition, we assume that there is a set of competing marketplaces $\mathcal{M} = \{1, 2, ...M\}$, that offer places for trade and provide a matching service between the buyers and sellers.

2.2 Marketplaces and Fees

Since we consider marketplaces to be commercial enterprises that seek to make a profit, we assume they charge fees for their service as match makers. The fee structure of a marketplace m is defined, as $\mathcal{P}_m = (p^b_m, p^s_m, q^b_m, q^s_m)$, $p^b_m, p^s_m \geq 0$ and $q^b_m, q^s_m \in [0, 1]$, where p^b_m, p^s_m are fixed flat fees charged to buyers and sellers respectively (in this paper, as an example, we consider registration fees charged to traders when they enter the marketplace as a typical kind of fixed flat fee), and q^b_m, q^s_m are percentage fees charged on profits made by buyers and sellers respectively (in the following, we refer to such fees as profit fees). Then the fees of all competing marketplaces constitute the *fee system* $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2, ...\mathcal{P}_M)$. Furthermore, the transaction price of a successful transaction in marketplace m is determined by a parameter $k_m \in [0, 1]$, i.e. a *discriminatory k-pricing* policy, which sets the transaction price of a matched buyer and seller at the point determined by k_m in the interval between their offers. The pricing parameters of all marketplaces constitute the pricing system $\mathcal{K} = (k_1, k_2, ..., k_M)$.

2.3 Trader Market Selection

We assume that traders can only choose a single marketplace at a time (called *single-homing*), but they can freely migrate to a different marketplace in the next trading round. A trading round proceeds as follows. First, all marketplaces publish their fees and pricing parameters. Second, based on the observed fees and pricing parameters, each trader selects a marketplace according to its market selection strategy. Third, traders submit their offers according to their bidding strategies. Finally, after all traders have submitted their offers, the marketplace matches buyers and sellers according to its matching policy and then executes transactions. For simplicity, we assume that only one unit of commodity can be traded by each trader in a giving trading round. Intuitively, we can see that the traders' choice of marketplaces is important since this significantly affects the marketplaces' positions in the competition. Given this, in the following, we present the traders' market selection strategies in more detail.

We consider a mixed market selection strategy, where each marketplace is selected with some probability. A pure strategy can be regarded as a degenerate case of a mixed strategy, where the particular pure strategy is selected with probability 1 and every other strategy with probability 0. Now, a mixed market selection strategy of buyer i is defined as $\omega_i^b : [\underline{l}, \overline{l}] \times \mathcal{M} \to [0, 1]$, which means the probability that buyer i with type θ^b chooses the marketplace m is $\omega_i^b(\theta^b, m)$, where $\sum_{m \in \mathcal{M}} \omega_i^b(\theta^b, m) \leq 1$. Here, $1 - \sum_{m \in \mathcal{M}} \omega_i^b(\theta^b, m)$ is the probability that buyer i with type θ^b chooses no marketplace. The complete mixed market selection strategy of buyer i with type θ^b is given by:

$$\delta_i^b(\theta^b) = \langle \omega_i^b(\theta^b, 1), \omega_i^b(\theta^b, 2), ... \omega_i^b(\theta^b, M) \rangle, \quad \delta_i^b(\theta^b) \in \Delta,$$

where Δ is the set of all possible mixed strategies of a trader:

$$\Delta = \left\{ \langle x_1, ..., x_M \rangle \in [0, 1]^M : \sum_{m=1}^{M} x_m \leq 1 \right\}$$

Similarly, we use $\omega_j^s : [\underline{c}, \overline{c}] \times \mathcal{M} \to [0, 1]$ to define the probability of selecting a marketplace of seller j, and write the complete strategy as:

$$\delta_j^s(\theta^s) = \langle \omega_j^s(\theta^s, 1), \omega_j^s(\theta^s, 2), ... \omega_j^s(\theta^s, M) \rangle, \quad \delta_j^s(\theta^s) \in \Delta$$

.

Now we use $\delta^b = \langle \delta_1^b(\cdot), \delta_2^b(\cdot), ... \delta_B^b(\cdot) \rangle$ to denote the strategy profile of buyers, and $\delta^s = \langle \delta_1^s(\cdot), \delta_2^s(\cdot), ... \delta_S^s(\cdot) \rangle$ that of the sellers. Given a buyers' strategy profile δ^b and a sellers' strategy profile δ^s, the expected utility of a buyer i with type θ^b is defined by:

$$\tilde{U}_i^b(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, \theta^b) = \sum_{m=1}^{M} \omega_i^b(\theta^b, m) \times \tilde{U}_{i,m}^b(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, \theta^b) \tag{1}$$

where $\tilde{U}_{i,m}^b(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, \theta^b)$ is buyer i's expected utility if it chooses to trade in the marketplace m, which depends on the specific matching policy adopted by marketplace m. The expected utility of the sellers is defined analogously.

3 Marketplace's Expected Profit

In the above we have specified a general framework for analysing competing double auction marketplaces. Before we can analyse the equilibrium charging behaviour of marketplaces, we need to know marketplaces' expected profits given their fees and given the behaviour of the traders. In this section, we will describe how to calculate marketplaces' expected profits.

In order to calculate the marketplaces' expected profits, we need to know which bidding strategy traders will use to submit their offers and which matching policy marketplaces will use to match buyers and sellers to make transactions. Specifically, traders' bidding strategy and marketplaces' matching policy used in this work are specified as follows. As we did in [1], we assume that traders use a truthtelling bidding strategy, which means they will submit their types as their offers during the trading process. For the matching policy, we consider *equilibrium matching* since this aims to maximise traders' profits and thus maximises the allocative efficiency for the marketplace. In detail, this policy will match the buyer with v-th highest limit price with the seller with v-th lowest cost price if the seller's cost price is not greater than the buyer's limit price. Furthermore, we assume that traders with the same type will employ the same market selection strategy. Thus in the following, we omit the trader's index i, j when it is intuitively clear.

Now in order to get insight from this complicated game with more traders and more types, we use the same simplifying assumptions made in [1]. We only consider the competition between two marketplaces, i.e. $M = 2$. In order to allow for tractable results, we restrict our analysis to discrete trader types. In particular, we assume that there are two types of buyers and two types of sellers: rich and poor, which are denoted by t_2^b and t_1^b respectively for buyers, and t_1^s and t_2^s for sellers. A rich buyer is defined as having a higher limit price than a poor buyer, i.e. $t_2^b > t_1^b$, and a rich seller is defined as having a lower cost price than a poor seller, i.e. $t_1^s < t_2^s$. Trader types are independently drawn from the discrete uniform distribution (i.e. both types are equally likely). In addition, since we focus on how to set marketplace fees, we keep the pricing parameter $k_m = 0.5$ ($m = 1, 2$). Then marketplaces can only affect traders' market selections by changing fees.

Now we are ready to derive the equations to calculate marketplaces' expected profits given the fee system $\mathcal{P}$. Firstly, we calculate marketplaces' expected profit given traders' market selection strategies: $\omega^b(t_1^b, m)$, $\omega^b(t_2^b, m)$, $\omega^s(t_1^s, m)$ and $\omega^s(t_2^s, m)$. In order to do this, we calculate the probability that there are exactly τ_1^b poor buyers and τ_2^b rich buyers choosing the marketplace m:

$$\varrho_m^b(\tau_1^b, \tau_2^b) = \binom{B}{\tau_1^b, \tau_2^b, B - \tau_1^b - \tau_2^b} * \left(\frac{\omega^b(t_1^b, m)}{2}\right)^{\tau_1^b} * \left(\frac{\omega^b(t_2^b, m)}{2}\right)^{\tau_2^b} * \left(1 - \frac{\omega^b(t_1^b, m)}{2} - \frac{\omega^b(t_2^b, m)}{2}\right)^{(B - \tau_1^b - \tau_2^b)} \quad (2)$$

where $\binom{B}{\tau_1^b, \tau_2^b, B - \tau_1^b - \tau_2^b}$ is the multinomial coefficient, $\frac{\omega^b(t_1^b, m)}{2}$ is the probability that a buyer is poor and chooses marketplace m. Similarly, we get the probability that there are exactly τ_1^s rich sellers and τ_2^s poor sellers in the marketplace m:

$$\varrho_m^s(\tau_1^s, \tau_2^s) = \binom{S}{\tau_1^s, \tau_2^s, S - \tau_1^s - \tau_2^s} * \left(\frac{\omega^s(t_1^s, m)}{2}\right)^{\tau_1^s} * \left(\frac{\omega^s(t_2^s, m)}{2}\right)^{\tau_2^s} * \left(1 - \frac{\omega^s(t_1^s, m)}{2} - \frac{\omega^s(t_2^s, m)}{2}\right)^{(S - \tau_1^s - \tau_2^s)} \quad (3)$$

Furthermore, marketplace m's expected profit when there are exactly τ_1^b poor buyers, τ_2^b rich buyers, τ_1^s rich sellers and τ_2^s poor sellers in this marketplace is calculated by:

$$\tilde{U}_m(\mathcal{P}, \tau_1^b, \tau_2^b, \tau_1^s, \tau_2^s) = (\tau_1^b + \tau_2^b) * p_m^b + (\tau_1^s + \tau_2^s) * p_m^s + \Lambda^b * q_m^b + \Lambda^s * q_m^s \quad (4)$$

where Λ^b, Λ^s are the buyers and sellers' share of the trading surplus respectively when τ_1^b poor buyers, τ_2^b rich buyers, τ_1^s rich sellers and τ_2^s poor sellers are matched according to the equilibrium matching policy[3]. At this moment, we can get the marketplace's expected profit given the traders' market selection strategies: $\omega^b(t_1^b, m)$, $\omega^b(t_2^b, m)$, $\omega^s(t_1^s, m)$ and $\omega^s(t_2^s, m)$:

$$\begin{aligned}&\tilde{U}_m\Big(\mathcal{P}, \omega^b(t_1^b, m), \omega^b(t_2^b, m), \omega^s(t_1^s, m), \omega^s(t_2^s, m)\Big)\\ &= \textstyle\sum_{\tau_1^b=0}^{B} \sum_{\tau_2^b=0}^{B-\tau_1^b} \sum_{\tau_1^s=0}^{S} \sum_{\tau_2^s=0}^{S-\tau_1^s} \varrho_m^b(\tau_1^b, \tau_2^b) * \varrho_m^s(\tau_1^s, \tau_2^s) * \tilde{U}_m(\mathcal{P}, \tau_1^b, \tau_2^b, \tau_1^s, \tau_2^s) \quad (5)\end{aligned}$$

Now given a fee system $\mathcal{P}$, we calculate the marketplaces' expected profits at the point where all traders use equilibrium market selection strategies. As we discussed in [1], there can exist multiple Nash equilibria of market selection strategies. In such cases, the marketplace's expected profit depends on which NEQ strategies traders will choose and the probability of choosing such NEQ strategies. In [1], given a fee system, we have used EGT to analyse how traders choose NEQ strategies. Similarly, in this paper, we use EGT to find which NEQ strategies traders will choose and with what probability.

In more detail, in EGT, players gradually adjust their strategies over time in response to the repeated observations of their opponents' strategies. In particular, the *replicator dynamics* equation is often used to specify the dynamic adjustment of the probability of which pure strategy should be played. Then in our work, the 4-population replicator equations (rich buyers, poor buyers, rich sellers and poor sellers) showing the dynamic changes of traders' selection strategies with respect to time t are given by:

$$\dot{\omega}^b(t_1^b, 1) = \frac{d\omega^b(t_1^b, 1)}{dt} = \Big(\tilde{U}_1^b(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_1^b) - \tilde{U}^b(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_1^b)\Big) * \omega^b(t_1^b, 1) \quad (6)$$

$$\dot{\omega}^b(t_2^b, 1) = \frac{d\omega^b(t_2^b, 1)}{dt} = \Big(\tilde{U}_1^b(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_2^b) - \tilde{U}^b(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_2^b)\Big) * \omega^b(t_2^b, 1) \quad (7)$$

$$\dot{\omega}^s(t_1^s, 1) = \frac{d\omega^s(t_1^s, 1)}{dt} = \Big(\tilde{U}_1^s(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_1^s) - \tilde{U}^s(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_1^s)\Big) * \omega^s(t_1^s, 1) \quad (8)$$

$$\dot{\omega}^s(t_2^s, 1) = \frac{d\omega^s(t_2^s, 1)}{dt} = \Big(\tilde{U}_1^s(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_2^s) - \tilde{U}^s(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_2^s)\Big) * \omega^s(t_2^s, 1) \quad (9)$$

As an example, $\dot{\omega}^b(t_1^b, 1)$ describes how the poor buyer with type t_1^b changes its probability of choosing marketplace 1. Here, $\tilde{U}_1^b(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_1^b)$ is the poor buyer's expected utility when choosing marketplace 1 given other traders' strategies, and $\tilde{U}^b(\mathcal{P}, \mathcal{K}, \delta^b, \delta^s, t_1^b)$ is the poor buyer's overall expected utility. Given

[3] This can be easily calculated. For example, when there are 2 rich buyers, 3 poor buyers, 3 rich sellers and 2 poor buyers in the marketplace m, $\Lambda^b = \Big(max(t_2^b - t_1^s, 0) * 2 + max(t_1^b - t_1^s, 0) + max(t_1^b - t_2^s, 0) * 2\Big) * k_m$ and $\Lambda^s = \Big(max(t_2^b - t_1^s, 0) * 2 + max(t_1^b - t_1^s, 0) + max(t_1^b - t_2^s, 0) * 2\Big) * (1 - k_m)$.

that traders use the truthtelling bidding strategy and marketplaces use the equilibrium matching policy, these equations have been derived in [1]. In order to get the dynamics of the strategies, we need to calculate *trajectories*, which indicate how the mixed strategies evolve. In more detail, initially, a mixed strategy is chosen as a starting point. For convenience, we use $\left(\omega^b(t_2^b,1),\omega^b(t_1^b,1),\omega^s(t_1^s,1),\omega^s(t_2^s,1)\right)$ to represent this starting point. The dynamics are then calculated according to the above replicator equations. According to the dynamic changes of traders' strategies, their current mixed strategy can be calculated. Such calculations are repeated until $\dot{\omega}^b(\cdot,1)$ and $\dot{\omega}^s(\cdot,1)$ become zero, at which point the equilibrium is reached. The replicator dynamics show the trajectories and how they converge to an equilibrium. When considering traders evolving from all possible starting points, we get several regions. The region where all trajectories converge to a particular equilibrium is called the *basin of attraction* of this equilibrium. The basin is very useful since its size indicates the probability of traders converging to that equilibrium, which is necessary for calculating marketplaces' expected profits.

However, since probabilities are continuous from 0 to 1, there are infinitely many mixed strategies available to each type of trader. Thus the set of possible starting points is also infinite, which results in the difficulty of accurately calculating the sizes of basins. In this work, we have to discretize the starting points to approximate the size of basin of attraction to each NEQ. By so doing, we know the probability of traders converging to each equilibrium. For example, in this paper we calculate the size of basin of attraction by discretizing the mixed strategy of each type from 0.01 to 0.99 with step size 0.049, which gives $21^4 = 194481$ different starting points. Note that if we use even more points, we can estimate the probability of traders' convergence to each equilibrium more accurately.

Now that we know, given a fee system $\mathcal{P}$, what NEQ strategies traders will choose and with what probabilities. We calculate the expected profit for a marketplace. Specifically, given that there are X possible NEQs market selection strategies, we use $(x_1, x_2, ..., x_X)$ to represent the probabilities of traders converging to these NEQs. Then the marketplace m's expected profit in the fee system $\mathcal{P}$ is:

$$\tilde{U}_m(\mathcal{P}) = \sum_{z=1}^{X} x_z * \tilde{U}_m\Big(\mathcal{P}, \omega^{zb}(t_1^b,m), \omega^{zb}(t_2^b,m), \omega^{zs}(t_1^s,m), \omega^{zs}(t_2^s,m)\Big) \quad (10)$$

where $\omega^{zb}(t_1^b,m)$, $\omega^{zb}(t_2^b,m)$, $\omega^{zs}(t_1^s,m)$ and $\omega^{zs}(t_2^s,m)$ denote the z-th NEQ market selection strategies.

4 Equilibrium Analysis of Market Fees

Given the equations to calculate the marketplaces' expected profits, we now define a Nash equilibrium for the marketplaces. Since the strategies of each marketplace consist of the range of possible fees, an equilibrium constitutes a fee system $\mathcal{P}$. Specifically, the Nash equilibrium fee system in our setting is defined as follows:

Definition 1 *The fees system $\mathcal{P}^* = (\mathcal{P}_1^*, \mathcal{P}_2^*, ...\mathcal{P}_M^*)$ constitutes a Nash equilibrium fee system, if $\forall m \in \mathcal{M}, \forall \mathcal{P} \in \Psi$,*

$$\tilde{U}_m(\mathcal{P}^*) = \sum_{z=1}^{X^*} x_z^* * \tilde{U}_m\Big(\mathcal{P}^*, \omega^{zb*}(t_1^b, m), \omega^{zb*}(t_2^b, m), \omega^{zs*}(t_1^s, m), \omega^{zs*}(t_2^s, m)\Big)$$
$$\geq \tilde{U}_m(\mathcal{P}) = \sum_{z=1}^{X} x_z * \tilde{U}_m\Big(\mathcal{P}, \omega^{zb}(t_1^b, m), \omega^{zb}(t_2^b, m), \omega^{zs}(t_1^s, m), \omega^{zs}(t_2^s, m)\Big)$$

where Ψ is the set of all possible fee systems.

We now analyse this equilibrium in detail. First, we need to calculate the marketplaces' expected profits given different fee systems. As we know, the range of possible fees is continuous, which results in infinitely many possible fee systems. In [7], researchers claim that for this kind of game, it is useful to approximate the game by restricting the strategy space, and results from the restricted strategy space still provide insights into the original game. Similarly in this paper, in order to obtain tractable results, we also restrict the fee space by discretizing these fees. For example, we discretize profit and registration fees from 0 to 1 with step size 0.1. Then we can calculate marketplaces' expected profits corresponding to these fees, and generate an expected profit matrix for marketplaces, by which we can analyse the equilibrium fee system.

We now analyse this issues in two cases. Specifically, in the first case, we consider two competing marketplaces that only charge the profit fees. In the second case, we consider a setting where one marketplace charges the registration fee, and the other marketplace charges the profit fee.

4.1 Both Marketplaces Charging Profit Fees

First we consider the case that only profit fees can be charged to traders. We assume that there are 5 buyers and 5 sellers, and the surpluses of buyers and sellers are symmetric. Specifically, we let $t_1^b = 4$, $t_2^b = 6$, $t_1^s = 0$ and $t_2^s = 2$. Furthermore, we assume that competing marketplaces charge the same profit fee to buyers and sellers (i.e. $q_m^b = q_m^s$)[4]. Now we discretize profit fee from 0 to 1 with step size 0.1. Therefore each marketplace can choose from 11 different profit fees. For two competing marketplaces, there are $11^2 = 121$ different fee systems. For each of these combinations, we use EGT to obtain the basin of attraction to each NEQ of market selection strategies. Then by approximating the size of each basin, we get the probability of traders choosing each NEQ. For example,

[4] At this moment, rich(poor) buyer and rich(poor) seller have the same behaviour of selecting marketplaces. Then Equations 6 and 9 are consistent and Equations 7 and 8 are consistent. By so doing, we reduce 4-population replicator dynamics to 2-population replicator dynamics. This is convenient for visualising how traders evolve their strategies and approximating the size of basins of attraction in a 2-dimensional space.

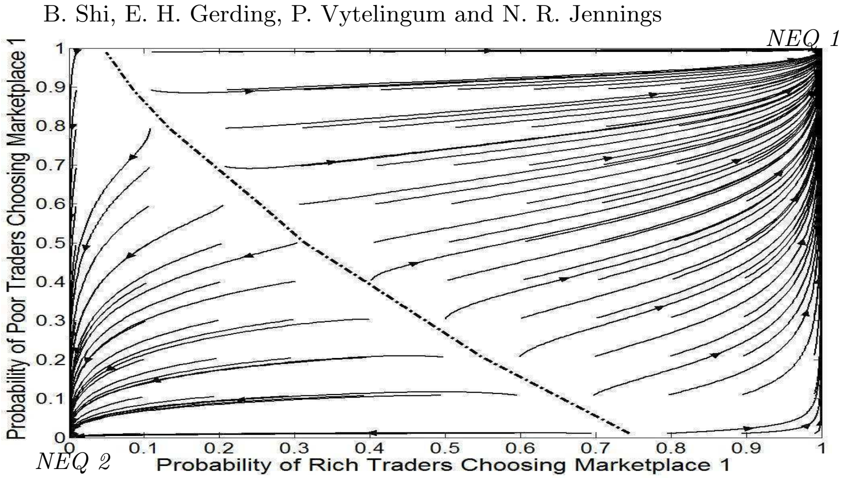
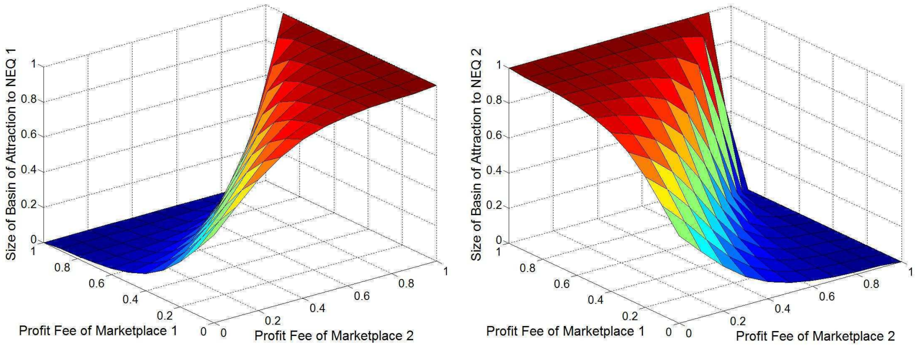


Fig. 1. Evolutionary process when both marketplaces 1 and 2 charge profit fees. The dotted line denotes the boundary between the basins of attractions.



(a) Size of basin of attraction to NEQ 1. (b) Size of basin of attraction to NEQ 2.

Fig. 2. Sizes of basins of attraction with respect to different fees.

when marketplace 1 charges 20% profit fee and marketplace 2 charges 30% profit fee, the basins of attraction are shown in Fig. 1[5]. From this figure, we find that all traders will converge to marketplace 1 (NEQ 1) or marketplace 2 (NEQ 2). By approximating the size of each basin of attraction, we can determine the probability of traders converging to each NEQ (the probabilities to NEQ 1 and NEQ 2 are 0.658 and 0.342 respectively in this case).

Now we explore traders' evolutionary process in all fee systems. The probability of traders choosing each NEQ of market selection strategies corresponding to all fee systems are shown in Fig. 2. Then using Equation 10, we calculate the marketplaces' expected profits. The results are shown in Table 1. From this table, by using Gambit[6], we find that both marketplaces charging 30% profit fee constitutes a unique pure NEQ fee system. Interestingly, in this equilibrium, both competing marketplaces charge non-zero profit fees and therefore make positive profits. This contrasts with competition between one-sided marketplaces (such

[5] Here we discretize the mixed strategy of each type of trader from 0.01 to 0.99 with a bigger step size, 0.098, for clearly visualising purpose.

[6] http://gambit.sourceforge.net/

	0.0	**0.1**	**0.2**	***0.3***	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
0.0	0.00,0.00	0.00,0.75	0.00,1.02	0.00,0.97	0.00,0.78	0.00,0.58	0.00,0.42	0.00,0.29	0.00,0.22	0.00,0.20	0.00,0.00
0.1	0.75,0.00	1.00,1.00	1.28,1.44	1.54,1.39	1.73,1.10	1.85,0.77	1.91,0.52	1.95,0.34	1.97,0.24	1.98,0.20	2.00,0.00
0.2	1.02,0.00	1.44,1.28	2.00,2.00	2.63,2.05	3.18,1.64	3.56,1.11	3.77,0.70	3.88,0.43	3.93,0.27	3.95,0.22	4.00,0.00
0.3	0.97,0.00	1.39,1.54	2.05,2.63	**3.00,3.00**	4.07,2.58	4.97,1.72	5.50,1.01	5.76,0.56	5.88,0.32	5.93,0.22	6.00,0.00
0.4	0.78,0.00	1.10,1.73	1.64,3.18	2.58,4.07	4.00,4.00	5.66,2.93	6.90,1.64	7.54,0.81	7.81,0.38	7.90,0.22	8.00,0.00
0.5	0.58,0.00	0.77,1.85	1.11,3.56	1.72,4.97	2.93,5.66	5.00,5.00	7.45,3.06	9.03,1.36	9.65,0.56	9.86,0.25	10.00,0.00
0.6	0.42,0.00	0.52,1.91	0.70,3.77	1.01,5.50	1.64,6.90	3.06,7.45	6.00,6.00	9.54,2.87	11.30,0.93	11.80,0.31	12.00,0.00
0.7	0.29,0.00	0.34,1.95	0.43,3.88	0.56,5.76	0.81,7.54	1.36,9.03	2.87,9.54	7.00,7.00	12.08,2.19	13.66,0.43	14.00,0.00
0.8	0.22,0.00	0.24,1.97	0.27,3.93	0.32,5.88	0.38,7.81	0.56,9.65	0.93,11.30	2.19,12.08	8.00,8.00	15.07,1.04	16.00,0.00
0.9	0.20,0.00	0.20,1.98	0.22,3.95	0.22,5.93	0.22,7.90	0.25,9.86	0.31,11.80	0.43,13.66	1.04,15.07	9.00,9.00	18.00,0.00
1.0	0.00,0.00	0.00,2.00	0.00,4.00	0.00,6.00	0.00,8.00	0.00,10.0	0.00,12.00	0.00,14.00	0.00,16.00	0.00,18.00	10.00,10.00

Table 1. Profits of marketplace 1 and marketplace 2. The first column is the profit fee of marketplace 1 and the first row is the profit fee of marketplace 2. The first element in each cell is marketplace 1's expected profit, and the second is marketplace 2's expected profit. Bold italic fees constitute a NEQ fee system.

as the classical Bertrand competition), in which competing marketplaces reduce their fees to their cost level and make zero profit in equilibrium [9]. This is because, in two-sided double auction marketplaces, there is still a probability that the traders converge to the marketplace which charges slightly higher fees compared to its opponent (as can be seen in Fig. 2).

Finally, we also analyse the case when both competing marketplaces charge registration fees. In this case, the traders' expected profits on both marketplaces may be negative, and then traders will not enter any marketplaces. However, other conclusions are similar to the case that both competing marketplaces charge profit fees. In particularly, we still find traders eventually converge to one marketplace. The same result holds if we change traders' types and the number of traders.

4.2 Asymmetric Market Fees

In this section, we will consider the case that different competing marketplaces charge different types of fees: marketplace 1 charges the profit fee, and marketplace 2 charges the registration fee. In previous section, we find that traders always converge to one of two equilibria when they are limited to charging the same type of fees. This means that marketplaces cannot co-exist when traders are in equilibrium. In this section, we first analyse whether competing marketplaces can co-exist when different types of fees are allowed. Then we analyse the equilibrium fee system.

We still assume that competing marketplaces charge the same fee to buyers and sellers (i.e. $p_m^b = p_m^s$ and $q_m^b = q_m^s$). Then we discretize registration and profit fees from 0 to 1 with step size 0.1. Then there are again 121 different fee systems. By evolving traders' strategies from different starting points and under different fee systems, we find co-existence of competing marketplaces. For example, when marketplace 1 charges 50% profit fee and marketplace 2 charges 0.8 registration fee, the evolution of traders' selection strategies from different starting points is shown in Fig. 3. From this figure, we find that there are now three basins of attraction: all traders converge to marketplace 1 (NEQ 1); all traders converge to marketplace 2 (NEQ 2); rich traders converge to marketplace 2, and poor traders converge to marketplace 1 (NEQ 3), i.e. two

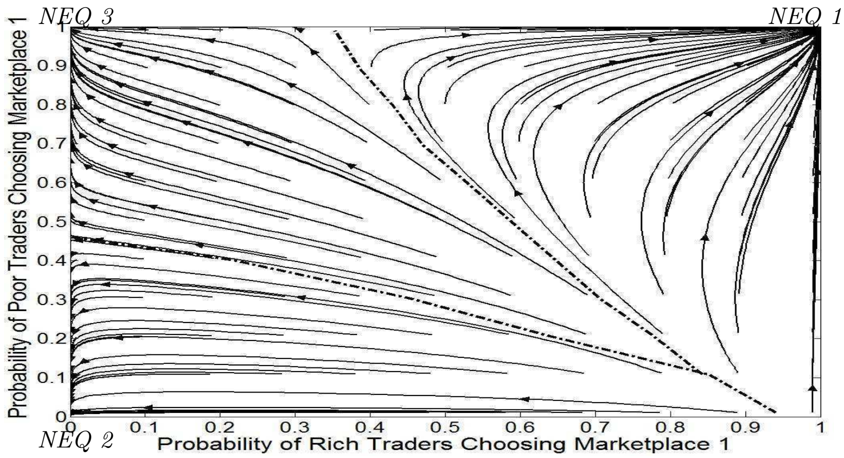
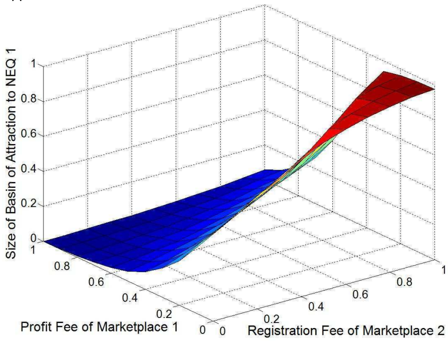
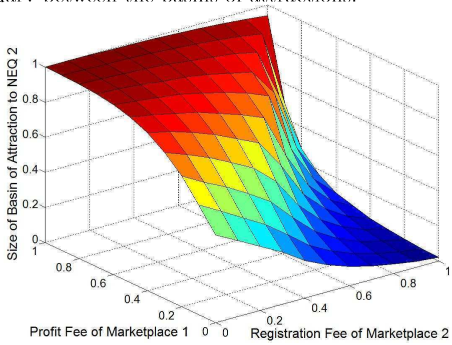


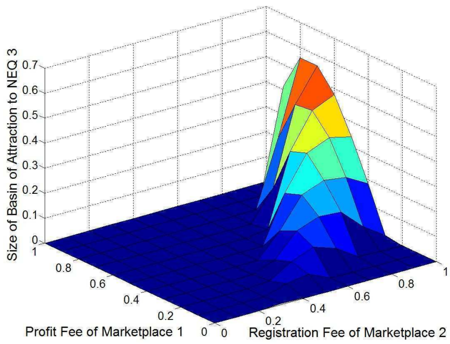
Fig. 3. Evolutionary process when marketplace 1 charges profit fee and marketplace 2 charges registration fee. The dotted line denotes the boundary between the basins of attractions.



(a) Size of basin of attraction to NEQ 1.



(b) Size of basin of attraction to NEQ 2.



(c) Size of basin of attraction to NEQ 3.

Fig. 4. Sizes of basins of attraction with respect to different fees.

competing marketplaces co-exist in equilibrium. By exploring traders' market selection strategies under all possible fee systems, we obtain the probabilities of traders converging to each NEQ, which are shown in Fig. 4. Fig. 4(c) shows in which fee systems, traders may converge to different marketplaces.

After estimating the probabilities of traders' convergence to each NEQ, we then calculate the marketplaces' expected profits using Equation 10. The mar-

	0.0	**0.1**	**0.2**	**0.3**	**0.4**	***0.5***	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
0.0	0.00,0.00	0.00,0.43	0.00,0.73	0.00,0.90	0.00,0.86	0.00,0.77	0.00,0.67	0.00,0.54	0.00,0.43	0.00,0.31	0.00,0.23
0.1	0.75,0.00	0.89,0.56	1.03,0.97	1.18,1.22	1.32,1.32	1.49,1.20	1.69,0.92	1.79,0.73	1.86,0.54	1.91,0.39	1.95,0.27
0.2	1.02,0.00	1.31,0.67	1.55,1.22	1.84,1.62	2.15,1.85	2.48,1.81	2.83,1.59	3.22,1.22	3.62,0.75	3.76,0.54	3.86,0.36
0.3	1.00,0.00	1.29,0.79	1.59,1.47	1.97,2.01	2.37,2.42	**2.88,2.55**	3.44,2.33	4.00,1.99	4.49,1.64	5.20,1.01	5.67,0.53
0.4	0.80,0.00	1.03,0.87	1.34,1.66	1.72,2.36	2.19,2.90	2.66,3.34	3.35,3.30	4.17,2.89	4.91,2.48	5.62,2.03	6.32,1.53
0.5	0.57,0.00	0.79,0.92	1.07,1.79	1.36,2.59	1.77,3.29	2.20,3.90	2.72,4.37	3.71,3.99	4.62,3.49	5.56,2.99	6.38,2.56
0.6	0.38,0.00	0.54,0.96	0.76,1.87	0.92,2.77	1.28,3.57	1.66,4.31	2.12,4.94	2.64,5.44	3.92,4.64	4.99,4.03	5.97,3.54
0.7	0.22,0.00	0.32,0.98	0.48,1.93	0.70,2.85	0.85,3.76	1.15,4.59	1.53,5.35	1.93,6.03	2.73,6.14	4.09,5.18	5.19,4.55
0.8	0.11,0.00	0.18,0.99	0.29,1.96	0.37,2.93	0.54,3.86	0.77,4.76	1.01,5.62	1.26,6.45	1.71,7.14	2.86,6.66	4.35,5.54
0.9	0.04,0.00	0.09,1.00	0.16,1.98	0.20,2.97	0.29,3.94	0.45,4.88	0.61,5.80	0.81,6.69	1.10,7.51	1.48,8.26	3.24,6.92
1.0	0.00,0.00	0.00,1.00	0.04,2.00	0.10,2.99	0.14,3.97	0.22,4.95	0.36,5.89	0.50,6.83	0.68,7.73	0.90,8.60	1.18,9.41

Table 2. Profits of marketplace 1 and marketplace 2. The first column is the profit fee of marketplace 1 and the first row is the registration fee of marketplace 2. The first element in each cell is marketplace 1's expected profit, and the second is marketplace 2's expected profit. Bold italic fees constitute a NEQ fee system.

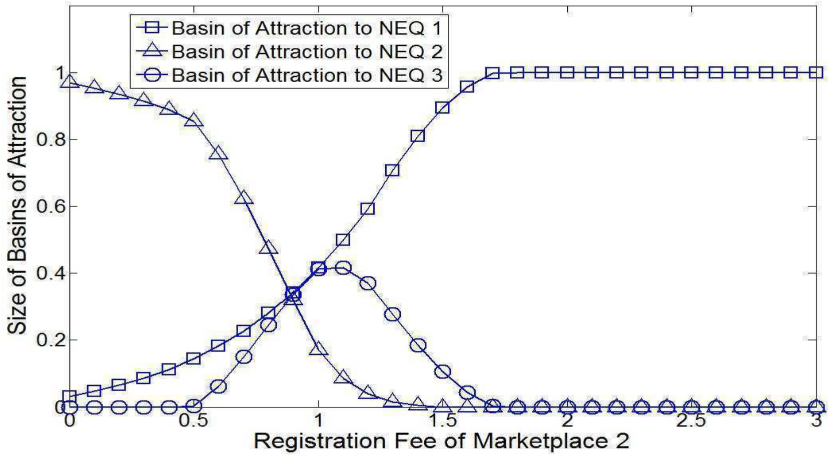


Fig. 5. Size of basins of attraction with respect to changed registration fees.

ketplaces' expected profits are shown in Table 2, from which we can see that in this case, marketplace 1 charging 30% profit fee, and marketplace 2 charging 0.5 registration fee constitutes the unique NEQ fee system.

Now we analyse how registration and profit fees affect the market selections of traders with different types. We note that, since the surplus of transaction between rich traders is higher than that between poor traders, then for the same profit fee, rich traders will lose more absolute profits than poor traders. Specifically, we let marketplace 1 charge 60% profit fee, marketplace 2 charge registration fee from 0 to 3 with step size 0.1. The sizes of basins of attraction with respect to the registration fee are shown in Fig. 5. From the figure, we can see that when the registration fee is low, traders prefer to choose marketplace 2. This is shown by the line with triangle, which is above two other lines in the beginning. Then as the registration fee increases, poor traders will choose to leave marketplace 2 since increased registration fee cause decreased and even negative profits for them. Rich traders may still prefer marketplace 2 since compared to marketplace 1 which extracts more absolute profits by charging the profit fee, marketplace 2 is still cheaper. This is shown by the increased lines with circle and square when the registration fee increases from 0 to about 1. However, when the registration fee becomes very high, both rich and poor traders will leave since their profits will be negative if they still choose to stay. This is shown by the decreased lines with triangle and circle and the increased line with square when the registration fee increases from 1 to 3.

5 Conclusions

In this paper, we provided an approach to estimate marketplaces' expected profits for a given fee system, based on the equilibrium of the traders' market selection strategies, and the likelihood of various equilibria occurring. We then analysed the NEQ fee system in two different cases: where marketplaces charge the same type of fees and where they charge different types of fees. Such analysis is useful to guide the charging behaviour of competing marketplaces. For the settings analysed in this paper, we found that a pure Nash equilibrium always exists in which both competing marketplaces charge non-zero fees. In addition, we found that when one marketplace charges the registration fee and the other charges the profit fee, two competing marketplace may co-exist in which the traders are in equilibrium. In this equilibrium, rich traders converge to the marketplace charging the registration fee, and poor traders converge to the marketplace charging the profit fee.

In this analysis, we have assumed that traders use the truthtelling bidding strategy, which results in traders' true profits being revealed to the marketplaces. However, when traders adopt a bidding strategy that can shade their true types, traders can keep more profits even when the profit fee is high. In the future, we intend to analyse the equilibrium fee system in this case. Moreover, in practice, traders may never converge to any NEQ market selection strategies since competing marketplaces may keep adapting their fees. In this dynamic process, we want to address how competing marketplaces dynamically change their fees corresponding to the opponents' fees and traders' current market selections. Furthermore, we also would like to generalise our analysis of the traders' equilibrium behaviour of market selection strategies as well as the bidding strategies by considering traders with continuous types, and then analyse how to find a NEQ fee system for competing marketplaces.

References

1. Shi, B., Gerding, E. H., Vytelingum, P., Jennings, N. R.: A Game-Theoretic Analysis of Market Selection Strategies for Competing Double Auction Marketplaces. Proc. 9th Int Conf. on Autonomous Agents and Multi-Agent Systems (2010)
2. Lee, R. S.: Competing Platforms. http://pages.stern.nyu.edu/ rslee/papers. Department of Economics, NYU Stern School of Business and Yahoo! Research (2008)
3. Friedman, D., Rust, J.: The double auction market: institutions, theories and evidence. XIV of Santa Fe Institute Studies in the Science of Complexity. Perseus Publishing (1993)
4. Gerding, E. H., McBurney, P., Niu, J., Parsons, S., Phelps, S.: Overview of CAT: A market design competition. Technical Report, Dept. of Computer Science, University of Liverpool (2007)
5. Caillaud, B., Jullien, B.: Chicken & egg: Competition among intermediation service providers. The RAND Journal of Economics. 34(2), 309-328 (2003)
6. Rochet, J., Tirole, J.: Platform Competition in Two-Sided Markets. Journal of European Economic Association. 1(4), 990-1029 (2003)
7. Wellman, M. P.: Methods for empirical game-theoretic analysis. Proc. 21st National Conference on Artificial Intelligence. 1552-1555 (2006)
8. Armstrong, M.: Competition in Two-Sided Markets. The RAND Journal of Economics. 37(3), 668-691 (2006)
9. Bertrand, J.: Book review of theorie mathematique de la richesse sociale and of recherches sur les principles mathematiques de la theorie des richesses. Journal de Savants. 67, 499-508 (1883)
10. Weibull, J. W.: Evolutionary Game Theory. The MIT Press (1996)

A Grey-Box Approach to Automated Mechanism Design

Jinzhong Niu[1*], Kai Cai[2], and Simon Parsons[2,3]

[1] School of Computer Science, University of Birmingham
Edgbaston, Birmingham B15 2TT, UK
jniu@cs.bham.ac.uk
[2] Department of Computer Science, Graduate Center, City University of New York
365, 5th Avenue, New York, NY 10016, USA
[3] Department of Comp. and Info. Science, Brooklyn College, City University of New York
2900 Bedford Avenue, Brooklyn, NY 11210, USA
parsons@sci.brooklyn.cuny.edu

Abstract. This paper presents an approach to automated mechanism design in the domain of double auctions. We describe a novel parameterized space of double auctions, and then introduce an evolutionary search method that searches this space of parameters. The approach evaluates auction mechanisms using the framework of the TAC Market Design Game and relates the performance of the markets in that game to their constituent parts using reinforcement learning. Experiments show that the strongest mechanisms we found using this approach not only win the Market Design Game against known, strong opponents, but also exhibit desirable economic properties when they run in isolation.

1 Introduction

Auctions play an important role in electronic commerce, and have been used to solve problems in distributed computing. A major problem to solve in these fields is: *Given a certain set of restrictions and desired outcomes, how can we design a good, if not optimal, auction mechanism; or when the restrictions and goals alter, how can the current mechanism be improved to handle the new scenario?*

The traditional answer to this question has been in the domain of auction theory [9]. A mechanism is designed by hand, analyzed theoretically, and then revised as necessary. The problems with the approach are exactly those that dog any manual process — it is slow, error-prone, and restricted to just a handful of individuals with the necessary skills and knowledge. In addition, there are classes of commonly used mechanisms, such as the double auctions that we discuss here, which are too complex to be analyzed theoretically, at least for interesting cases [21].

Automated mechanism design (AMD) aims to overcome the problems of the manual process by designing auction mechanisms automatically. AMD considers design to be a search through some space of possible mechanisms. For example, Cliff [2] and Phelps *et al.* [16, 17] explored the use of evolutionary algorithms to optimize different aspects of the continuous double auction. Around the same time, Conitzer and Sandholm [4] were examining the complexity of building a mechanism that fitted a particular specification.

*This work was carried out as part of the first author's PhD research at CUNY.

These different approaches were all problematic. The algorithms that Conitzer and Sandholm considered dealt with exhaustive search, and naturally the complexity was exponential. In contrast, the approaches that Cliff and Phelps *et al.* pursued were computationally more appealing, but gave no guarantee of success and were only searching tiny sections of the search space for the mechanisms they considered. As a result, one might consider the work of Cliff and Phelps *et al.*, and indeed the work we describe here, to be what Conitzer and Sandholm [5] call "incremental" mechanism design, where one starts with an existing mechanism and incrementally alters parts of it, aiming to iterate towards an optimal mechanism. Similar work, though work that uses a different approach to searching the space of possible mechanisms has been carried out by [20] and has been applied to several different mechanism design problems [18].

The problem with taking the automated approach to mechanism design further is how to make it scale — though framing it as an incremental process is a good way to look at it, it does not provide much practical guidance about how to proceed. Our aim in this paper is to provide more in the way of practical guidance, showing how it is possible to build on a previous analysis of the most relevant components of a complex mechanism in order to set up an automated mechanism design problem, and then describing one approach to solving this problem.

2 Grey-box AMD

We propose a *grey-box* AMD approach, which emerged from our previous work on the analyses of the CAT games.

2.1 From analyses of CAT games towards a grey-box approach

The CAT game, a.k.a. the Trading Agent Competition Market Design game, which has run for the last three years, asks entrants to design a market for a set of automated traders which are based on standard algorithms for buying and selling in a double auction, including ZI-C [8], ZIP [3], RE [6], and GD [7]. The game is broken up into a sequence of *days*, and each day every trader picks a market to trade in, using a market selection strategy that models the situation as an n-armed bandit problem [19, Section 2]. Markets are allowed to charge traders in a variety of ways and are scored based on the number of traders they attract (market share), the profits that they make from traders (profit share), and the number of successful transactions they broker relative to the total number of shouts placed in them (transaction success rate). Full details of the game can be found in [1].

We picked the CAT game as the basis of our work for four main reasons. First, the double auctions that are the focus of the design are a widely used mechanism. Second, the competition is run using an open-source software package called JCAT which is a good basis for implementing our ideas. Third, after three years of competition, a number of specialists have been made available by their authors, giving us a library of mechanisms to test against. Fourth, there have been a number of publications that analyze different aspects of previous entrants, giving us a good basis from which to start searching for new mechanisms.

With colleagues we have carried out two previous studies of CAT games [11, 13], which mirror the white-box and black-box analyses from software engineering. [13] provides a white-box analysis, looking inside each market mechanism in order to identify which components it contains, and relating the performance of each mechanism to the operation of its components. [11] provides a black-box analysis, which ignores the detail of the internal components of each market mechanism, but provides a much more extensive analysis of how the markets perform. These analyses make a good combination for examining the strengths and weaknesses of specialists. The white-box approach is capable of relating the internal design of a strategy to its performance and revealing which part of the design may cause vulnerabilities, but it requires internal structure and involves manual examination. The black-box approach does not rely upon the accessibility of the internal design of a strategy. It can be applied to virtually any strategic game, and is capable of evaluating a design in many more situations. However, the black-box approach tells us little about what may have caused a strategy to perform poorly and provides little in the way of hints as to how to improve the strategy. It is desirable to combine these two approaches in order to benefit from the advantages of both. Following the GA-based approach to trading strategy acquisition and auction mechanism design in [2, 15, 17], we propose what we call a *grey-box* approach to automated mechanism design that solves the problem of automatically creating a complex mechanism by searching a structured space of auction components. In other words, we concentrate on the components of the mechanisms as in the white-box approach, but take a black-box view of the components, evaluating their effectivenesses by looking at their performance against that of their peers.

More specifically, we view a market mechanism as a combination of auction rules, each as an atomic building block. We consider the problem: *how can we find a combination of rules that is better than any known combination according to a certain criterion, based on a pool of existing building blocks?* The black-box analysis in [11] maintains a population of strategies and evolves them generation by generation based on their fitnesses. Here we intend to follow a similar approach, maintaining a population of components or building blocks for strategies, associating each block with a *quality score*, which reflects the fitnesses of auction mechanisms using this block, exploring the part of the space of auction mechanisms that involves building blocks of higher quality, and keeping the best mechanisms we find.

Having sketched our approach at a high level, we now look in detail at how it can be applied in the context of the CAT game.

2.2 A search space of double auctions

The first issues we need to address are *what composite structure is used to represent auction mechanisms?* and *where can we obtain a pool of building blocks?*

Viewing an auction as a structured mechanism is not a new idea. Wurman *et al.* [22] introduced a conceptual, parameterized view of auction mechanisms. Niu *et al.* [13] extended this framework for auction mechanisms competing in CAT games and provided a classification of entries in the first CAT competition that was based on it. The extended framework includes multiple intertwined components, or *policies*, each regulating one aspect of a market. We adopt this framework, include more candidates for each type of policy and take into consideration parameters that are used by these policies.

These policies are either inferred from the literature [10], taken from our previous work [11, 13, 14], or contributed by entrants to the CAT competitions. The set of policies, each a building block, form a solid foundation for the grey-box approach.

Figure 1 illustrates the building blocks as a tree structure which we describe after we review the blocks themselves. Below we describe the different types of policies just briefly due to space limitations. An in-depth understanding of these policies is not required in understand the grey-box approach, but a full description of these policies can be found in the extended version of this paper [12].

Matching policies, denoted as M in Figure 1, define how a market matches shouts made by traders, including *equilibrium matching* (ME), *max-volume matching* (MV), and *theta matching* (MT). ME clears the market at the equilibrium price, matching asks (offers to sell) lower than the price with bids (offers to buy) higher than the price. MV maximizes transaction volume by considering also less-competitive shouts that would not be matched in ME. MT uses a parameter, $\theta \in [-1,1]$, to realize a transaction volume that is proportional to 0 and those realized in ME and MV.

Quote policies, denoted as Q in Figure 1, determine the quotes issued by markets, including *two-sided quoting* (QT), *one-sided quoting* (QO), and *spread-based quoting* (QS). Typical quotes are ask and bid quotes, which respectively specify the upper bound for asks and the lower bound for bids that may be placed in a quote-driven market. QT defines the quotes based on information from both the seller side and the buyer side, while QO does so considering only information from a single side. QS extends QT to maintain a higher ask quote and a lower bid quote for use with MV.

Shout accepting policies, denoted as A in Figure 1, judge whether a shout made by a trader should be permitted in the market, including *always accepting* (AA), *never accepting* (AN), *quote-beating accepting* (AQ), *self-beating accepting* (AS), *equilibrium-beating accepting* (AE), *average-beating accepting* (AD), *history-based accepting* (AH), *transaction-based accepting* (AT), and *shout type-based accepting* (AY). AE uses a parameter, w, to specify the size of a sliding window in terms of the number of transactions, and a second parameter, δ, to relax the restriction on shouts [14]. AD is basically a variant of AE and uses the standard deviation of transaction prices in the sliding window rather than w to relax the restriction on shouts. AH is derived from the GD trading strategy and accepts only shouts that will be matched with probability no lower than a specified threshold, $\tau \in [0,1]$. AY stochastically allows shouts based merely on their types, i.e., asks or bids, and uses a parameter, $q \in [0,1]$, to control the chances that shouts of either type are allowed to place.

Clearing conditions, denoted as C in Figure 1, define when to clear the market and execute transactions between matched asks and bids, including *continuous clearing* (CC), *round clearing* (CR), and *probabilistic clearing* (CP). CP uses a parameter, $p \in [0,1]$, to define a continuum of clearing rules with CR and CC being the two ends.

Pricing policies, denoted as P in Figure 1, set transaction prices for matched ask-bid pairs, including *discriminatory k-pricing* (PD), *uniform k-pricing* (PU), *n-pricing* (PN), and *side-biased pricing* (PB). Both PD and PU use a prefixed parameter, $k \in [0,1]$, to control the bias in favor of buyers or sellers, and PB adjusts an internal k aiming to

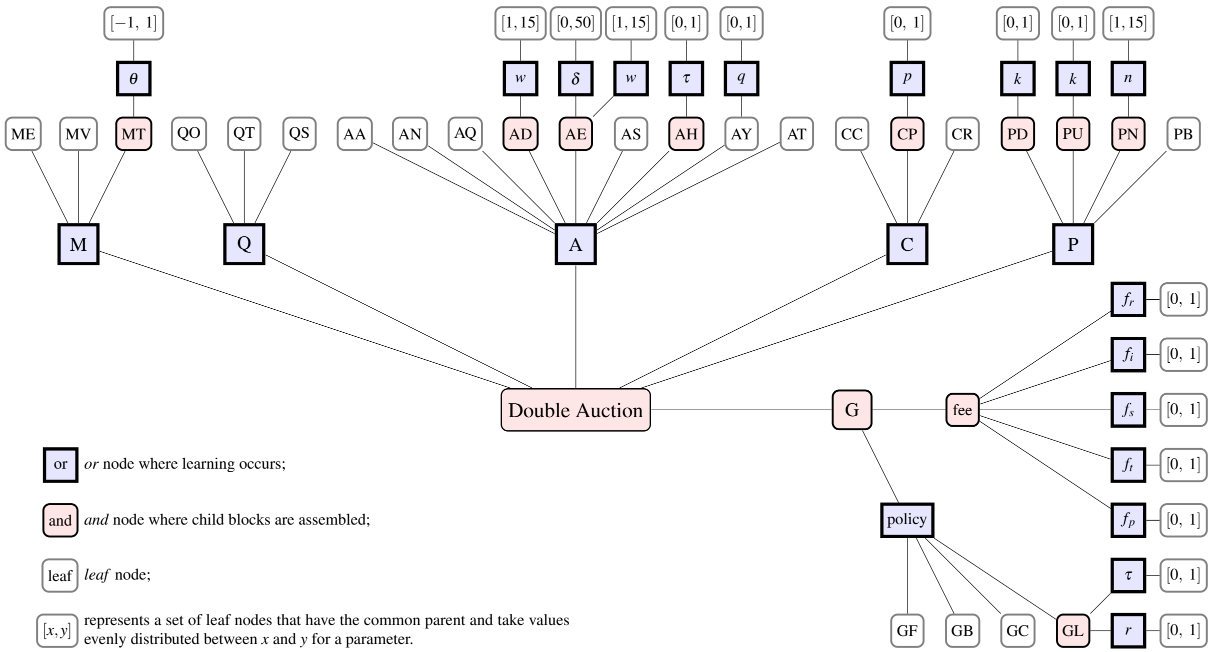


Fig. 1: The search space of double auctions modeled as a tree, discussed in details in Section 2.

obtain a balanced demand and supply. PN was introduced in [14] and sets the transaction price as the average of the latest n pairs of matched asks and bids.

Charging policies, denoted as G in Figure 1, determine the charges imposed by a market, including *fixed charging* (GF), *bait-and-switch charging* (GB), and *charge-cutting charging* (GC), *learn-or-lure-fast charging* (GL). GF imposes fixed charges while the rest three policies adapt charges over time in different ways. GL relies upon two parameters, τ and r, to achieve dynamic adjustments. All these charging policies require an initial set of fees on different activities, including fee on registration, fee on information, fee on shout, fee on transaction, and fee on profit, denoted as f_r, f_i, f_s, f_t, and f_p respectively in Figure 1.

2.3 The GREY-BOX-AMD algorithm

The tree model of double auctions in Figure 1 illustrates how building blocks are selected and assembled level by level. There are *and* nodes, *or* nodes, and *leaf* nodes in the tree. An *and* node, rounded and filled, combines a set of building blocks, each represented by one of its child nodes, to form a compound building block. The root node, for example, is an *and* node to assemble policies, one of each type described in the previous section, to obtain a complete auction mechanism. An *or* node, rectangular and filled, represents the decision making of selecting a building block from the candidates represented by the child nodes of the *or* node based on their quality scores. This selection occurs not only for those major aspects of an auction mechanism, i.e. M, Q, A, P, C, and G (at G's child node of 'policy' in fact), but also for minor components, for example, a learning component for an adaptive policy (following Phelps *et al.*'s work on acquiring a trading strategy [15]), and for determining optimal values of parameters in a policy, like θ in MT and k in PD. A *leaf* node represents an atomic block that can either be for selection at its *or* parent node or be further assembled into a bigger block by its *and* parent node. A special type of *leaf* node in Figure 1 is that with a label in the format of $[x, y]$. Such a *leaf* node is a convenient representation of a set of *leaf* nodes that have a common parent — the parent of this special *leaf* node — and take values evenly distributed between x and y for the parameter labeled at the parent node.

or nodes contribute to the variety of auction mechanisms in the search space and are where exploitation and exploration occur. We model each *or* node as an n-armed bandit learner that chooses among candidate blocks, and use the simple softmax method [19, Section 2.3] to solve this learning problem.

Given a set of building blocks, $\mathbb{B}$, and a set of fixed markets, $\mathbb{FM}$, as targets to beat, we define the skeleton of the grey-box algorithm in Algorithm 1. The GREY-BOX-AMD algorithm runs a certain number of steps (NUM_OF_STEPS in Line 2). At each step, a single CAT game is created (CREATE-GAME() in Line 3) and a set of markets are prepared for the game. This set of markets includes all markets in $\mathbb{FM}$, a certain number (NUM_OF_SAMPLES in Line 5) of markets sampled from the search space, denoted as $\mathbb{SM}$, and a certain number (NUM_OF_HOF_SAMPLES in Line 11) of markets, denoted as $\mathbb{EM}$, chosen from a Hall of Fame, $\mathbb{HOF}$. All these markets are put into the game, which is run to evaluate the performance of these markets (RUN-GAME(G, $\mathbb{FM} \cup \mathbb{EM} \cup \mathbb{SM}$) in Line 12). $\mathbb{HOF}$ has a fixed capacity, CAPACITY_OF_HOF, and maintains markets that performed well in games at previous steps in terms of their average scores across games

```
GREY-BOX-AMD(𝔹, 𝔽𝕄)
 1  ℍ𝕆𝔽 ← {}
 2  for s ← 1 to NUM_OF_STEPS
 3  do G ← CREATE-GAME()
 4      𝕊𝕄 ← {}
 5      for m ← 1 to NUM_OF_SAMPLES
 6      do M ← CREATE-MARKET()
 7          for t ← 1 to NUM_OF_POLICYTYPES
 8          do B ← SELECT(𝔹_t, 1)
 9              ADD-BLOCK(M, B)
10          𝕊𝕄 ← 𝕊𝕄 ∪ {M}
11      𝔼𝕄 ← SELECT(ℍ𝕆𝔽, NUM_OF_HOF_SAMPLES)
12      RUN-GAME(G, 𝔽𝕄 ∪ 𝔼𝕄 ∪ 𝕊𝕄)
13      for each M in 𝔼𝕄 ∪ 𝕊𝕄
14      do UPDATE-MARKET-SCORE(M, SCORE(G, M))
15          if M not in ℍ𝕆𝔽
16             then ℍ𝕆𝔽 ← ℍ𝕆𝔽 ∪ {M}
17          if CAPACITY_OF_HOF < |ℍ𝕆𝔽|
18             then ℍ𝕆𝔽 ← ℍ𝕆𝔽 − {WORST-MARKET(ℍ𝕆𝔽)}
19          for each B used by M
20          do UPDATE-BLOCK-SCORE(B, SCORE(G, M))
21  return ℍ𝕆𝔽
```

Algorithm 1: The GREY-BOX-AMD algorithm.

they participated. $\mathbb{HOF}$ is empty initially, updated after each game, and returned in the end as the result of the grey-box process.

Each market in $\mathbb{SM}$ is constructed based on the tree model in Figure 1. After an 'empty' market mechanism, M, is created (CREATE-MARKET() in Line 6), building blocks can be incorporated into M (ADD-BLOCK(M,B) in Line 9, where $B \in \mathbb{B}$). NUM_OF_POLICYTYPES in Line 7 defines the number of different policy types, and from each group of policies of same type, denoted as $\mathbb{B}_t$ where t specifies the type, a building block is chosen for M (SELECT($\mathbb{B}_t$, 1) in Line 8). For simplicity, this algorithm illustrates only what happens to the *or* nodes at the high level, including M, Q, A, C, and P. Markets in $\mathbb{EM}$ are chosen from $\mathbb{HOF}$ in a similar way (SELECT($\mathbb{HOF}$, NUM_OF_HOF_SAMPLES) in Line 11).

After a CAT game, G, completes at each step, the game score of each participating market $M \in \mathbb{SM} \cup \mathbb{EM}$, SCORE($G$, M), is recorded and the game-independent score of M, SCORE(M), is updated (UPDATE-MARKET-SCORE(M, SCORE(G, M)) in Line 14). If M is not currently in $\mathbb{HOF}$ and SCORE(M) is higher than the lowest score of markets in $\mathbb{HOF}$, it replaces that corresponding market (WORST-MARKET($\mathbb{HOF}$) in Line 18).

SCORE(G, M) is also used to update the quality score of each building block used by M (UPDATE-BLOCK-SCORE(B, SCORE(G, M)) in Line 20). Both UPDATE-MARKET-SCORE and UPDATE-BLOCK-SCORE calculate respectively game-independent scores of markets and quality scores of building blocks by averaging feedback SCORE(G, M) over time. Because choosing building blocks occurs only at *or* nodes in the tree, only child nodes of an *or* node have quality scores and receive feedback after a CAT game. Initially, quality scores of building blocks are all 0, so that the probabilities of choosing them are even. As the exploration proceeds, fitter blocks score higher and are chosen more often to construct better mechanisms.

3 Experiments

This section describes the experiments that are carried out to acquire auction mechanisms using the grey-box approach.

3.1 Experimental setup

We extended JCAT with the parameterized framework of double auctions and all the individual policies described in Section 2.2. To reduce the computational cost, we eliminated the exploration of charging policies by focusing on mechanisms that impose a charge of 10% on trader profit, which we denote as $\mathsf{GF}_{0.1}$. Analysis of CAT games [11] and what entries have typically charged in actual CAT competitions, especially in the latest two events, suggest that such a charging policy is a reasonable choice to avoid losing either intra-marginal or extra-marginal traders. Even with this cut-off, the search space still contains more than $1,200,000$ different kinds of auction mechanisms, due to the variety of policies on aspects other than charging and the choices of values for parameters.

The experiments that we ran to search the space each last 200 steps. At each step, we sample two auction mechanisms from the space, and run a CAT game to evaluate them against four fixed, well known, mechanisms plus two mechanisms from the Hall of Fame. To sample auction mechanisms, the softmax exploration method used by *or* nodes starts with a relatively high temperature ($\tau = 10$) so as to explore randomly, then gradually cools down, τ scaling down by 0.96 (α) each step, and eventually maintains a temperature ($\tau = 0.5$) that guarantees a non-negligible probability of choosing even the worst action any time. After all, our goal in the grey-box approach is not to converge quickly to a small set of mechanisms, but to explore the space as broadly as possible and avoid being trapped in local optima.

The fixed set of four markets in every CAT game includes two CH markets — CH_l and CH_h — and two CDA markets — CDA_l and CDA_h — with one of each charging 10% on trader profit, like $\mathsf{GF}_{0.1}$ does, and the other charging 100% on trader profit (denoted as $\mathsf{GF}_{1.0}$). The CH and CDA mechanisms are two common double auctions and have been used in the real world for many years, in financial marketplaces in particular due to their high allocative efficiency. Earlier experiments we ran, involving CH and CDA markets against entries into CAT competitions, indicate that it is not trivial to win over these two standard double auctions. Markets with different charge levels are included to avoid any sampled mechanisms taking advantage otherwise. Based on the parameterized framework in Section 2.2, the CH and CDA markets can be represented as follows:

$$\begin{aligned} \text{CH}_l \,/\, \text{CH}_h &= \mathsf{ME} + \mathsf{QT} + \mathsf{AQ} + \mathsf{CR} + \mathsf{PU}_{k=0.5} + \mathsf{GF}_{0.1} \,/\, \mathsf{GF}_{1.0} \\ \text{CDA}_l \,/\, \text{CDA}_h &= \mathsf{ME} + \mathsf{QT} + \mathsf{AQ} + \mathsf{CC} + \mathsf{PD}_{k=0.5} + \mathsf{GF}_{0.1} \,/\, \mathsf{GF}_{1.0} \end{aligned}$$

The Hall of Fame that we maintain during the search contains ten 'active' members and a list of 'inactive' members. After each CAT game, the two sampled mechanisms are compared with those active Hall of Famers. If the score of a sampled mechanism is higher than the lowest average score of the active Hall of Famers, the sampled mechanism is inducted into the Hall of Fame and replaces the corresponding Hall of Famer, which becomes inactive and ineligible for CAT games at later steps (lines 15–18 in Algorithm 1). An inactive Hall of Famer may be reactivated if an identical mechanism happens to be sampled from the space again and scores high enough to promote its average score to surpass the lowest score of active Hall of Famers. In addition, the softmax method used to choose two Hall of Famers out of the ten active ones involves a constant $\tau = 0.3$. Since the scores of the Hall of Famers gradually converge in the experiments and the difference between the best and the worst Hall of Famers is less than 25% (see Figure 2b below), this value of τ guarantees that the bias towards the best Hall of Famers is modest and all Hall of Famers have fairly big chances to be chosen.

Each CAT game is populated by 120 trading agents, using ZI-C, ZIP, RE, and GD strategies, a quarter of the traders using each strategy. Half the traders are buyers, half are sellers. The supply and demand schedules are both drawn from a uniform distribution between 50 and 150. Each CAT game lasts 500 days with ten rounds for each day. This setup is similar to that of actual CAT competitions except for a smaller trader population that helps to reduce computational costs. A 200-step grey-box experiment takes around sixteen hours on a WINDOWS PC that runs at 2.8GHz and has a 3GB memory. To obtain reliable results, we ran the grey-box experiments for 40 iterations and the results that are reported in the next section are averaged over these iterations.

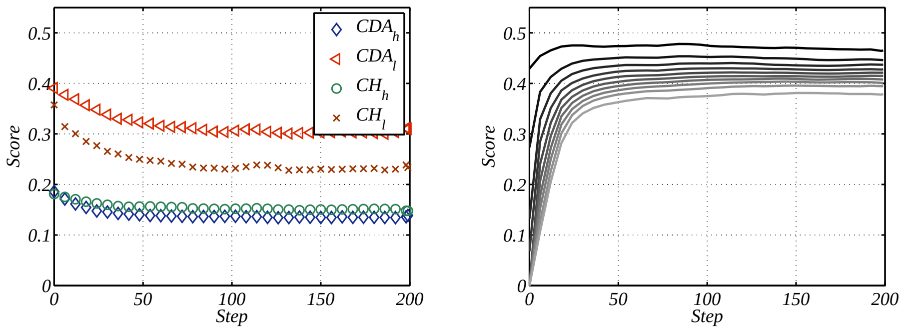
3.2 Experimental results

We carried out four experiments to check whether the grey-box approach is successful in searching for good auction mechanisms.

First, we measured the performance of the generated mechanisms indirectly, through their effect on other mechanisms. Since the four standard markets participate in all the CAT games, their performance over time reflects the strength of their opponents — they will do worse as their opponents get better — which in turn reflects whether the search generates increasingly better mechanisms. Figure 2a shows that the scores of the four markets (more specifically, the average daily scores of the markets in a game) decrease over 200 games, especially over the first 100 games, suggesting that the mechanisms we are creating get better as the learning process progresses.

Second, we measured the performance of the set of mechanisms we created more directly. The mechanisms that are active in the Hall of Fame at a given point represent the best mechanisms that we know about at that point and their performance tells us more directly how the best mechanisms evolve over time. Figure 2b shows the scores of the ten active Hall of Famers at each step over 200-step runs.[4] As in Figure 2a, the first 100 steps sees a clear, increasing trend. Even the scores of the worst of the ten at the end

[4] Note that the active Hall of Famers will be different mechanisms at different steps in the process, so what we see in the figure is the performance of the best mechanisms we know of up to the point we collected the data.



(a) The four fixed auction mechanisms. (b) The top ten active Hall of Famers.

Fig. 2: Scores of market mechanisms across 200 steps (games), averaged over 40 runs.

are above 0.35, higher than the highest score of the four fixed markets from Figure 2a, and the difference is statistically significant at the 95% confidence level. Thus we know that our approach will create mechanisms that outperform standard mechanisms, though we should not read too much into this since we trained our new mechanisms directly against them.

Third, a better test of the new mechanisms is to run them against those mechanisms that we know to be strong in the context of CAT games, asking what would have happened if our Hall of Fame members had been entered into prior CAT competitions and had run against the carefully hand-coded entries in those competitions. We chose three Hall of Famers, which are internally labeled as `SM7.1`, `SM88.0`, and `SM127.1` and can be represented in the parameterized framework in Section 2.2 as follows:

$$\begin{aligned}\texttt{SM7.1} &= \mathsf{MV} + \mathsf{QO} + \mathsf{AH}_{\tau=0.4} + \mathsf{CP}_{p=0.3} + \mathsf{PN}_{n=11} + \mathsf{GF}_{0.1}\\ \texttt{SM88.0} &= \mathsf{MT}_{\theta=0.4} + \mathsf{QT} + \mathsf{AA} + \mathsf{CP}_{p=0.4} + \mathsf{PU}_{k=0.7} + \mathsf{GF}_{0.1}\\ \texttt{SM127.1} &= \mathsf{MV} + \mathsf{QS} + \mathsf{AS} + \mathsf{CP}_{p=0.4} + \mathsf{PU}_{k=0.7} + \mathsf{GF}_{0.1}\end{aligned}$$

We ran these three mechanisms against the best recreation of past CAT competitions that we could achieve given the contents of the TAC agent repository,[5] where competitors are asked to upload their entries after the competition. There were enough entries in the repository at the time we ran the experiments to create reasonable facsimiles of the 2007 and 2008 competitions, but there were not enough entries from the 2009 competition for us to recreate that year's competition. The CAT games were set up in a similar way to the competitions, populated by 500 traders that are evenly split between buyers and sellers and between the four trading strategies — ZI-C, ZIP, RE, and GD — and the private values of sellers or buyers were drawn from a uniform distribution between 50 and 150. For each recreated competition, we ran three games.

Table 1 lists the average cumulative scores of all the markets across their three games along with the standard deviations of those scores. The three new mechanisms we obtained from the grey-box approach beat the actual entries for CAT 2007 and CAT

[5] `http://www.sics.se/tac/showagents.php`.

Table 1: The scores of markets in CAT games including the best mechanisms from the grey-box approach and entries in prior CAT competitions, averaged over three CAT games respectively.

(a) Against CAT 2007 entries.

Market	Score	SD
SM7.1	199.4500	5.9715
SM88.0	191.1083	10.3186
SM127.1	180.1277	9.0289
MANX	154.6953	1.3252
CrocodileAgent	142.0523	9.0867
TacTex	138.4527	5.8224
PSUCAT	133.1347	5.6565
PersianCat	124.3767	11.2409
jackaroo	108.8017	8.6851
IAMwildCAT*	106.8897	4.4006
Mertacor	89.1707	4.9269

(b) Against CAT 2008 entries.

Market	Score	SD
SM7.1	196.7240	9.2843
SM88.0	186.9247	4.2184
SM127.1	183.5887	9.7835
jackaroo	177.5913	2.5722
Mertacor	161.5440	5.8741
MANX	147.3050	15.7718
IAMwildCAT	142.9167	8.9581
PersianCat	139.1553	17.9783
DOG	130.2197	18.9782
MyFuzzy	125.9630	1.9221
CrocodileAgent*	71.4820	5.8687
PSUCAT*	68.3143	6.7389

* IAMwildCAT from CAT 2007, and CrocodileAgent and PSUCAT from CAT 2008 worked abnormally during the games and tried to impose invalid fees, probably due to competition from the three new, strong opponents. Although we modified JCAT to avoid kicking out these markets on those trading days when they impose invalid fees — which JCAT does in an actual CAT competition — these markets still perform poorly, in contrast to their rankings in the actual competitions.

2008 by a comfortable margin in both cases. The fact that we can take mechanisms that we generate in one series of games (against the fixed opponents and other new mechanisms) and have them perform well against a separate set of mechanisms suggests that the grey-box approach learns robust mechanisms.

In passing, we note that the rankings of the entries from the repository do not reflect those in the actual CAT competitions. This is to be expected since the entries now face much stronger opponents and different markets will, in general, respond differently to this. Excluding the markets that attempt to impose invalid fees and are marked with '*', we can see that the overall performance of entries into the 2008 CAT competition is better than that of those into the 2007 CAT competition when they face the three new, strong, opponents, reflecting the improvement in the entries over time.

Finally, we tested the performance of SM7.1, SM88.0, and SM127.1 when they are run in isolation, applying the same kind of test that auction mechanisms are traditionally subject to. We tested the mechanisms both for allocative efficiency and, following our work in [14], for the extent to which they trade close to theoretical equilibrium as measured by the coefficient of convergence, α, even when populated by minimally rational traders. In [14] we investigated a class of double auctions, called NCDAEE, which can be represented as:

Table 2: Properties of the best mechanisms from the grey-box experiments and the auction mechanisms explored in [14]. All NCDAEE mechanisms are configured to have $w = 4$ in their AE policies and $n = 4$ in their PN policies.The best result in each column is shaded. Data in the first four rows are averaged over 1,000 runs and those in the last four are averaged over 100 runs.

Market	ZI-C				GD			
	E_a		α		E_a		α	
	Mean	SD	**Mean**	SD	**Mean**	SD	**Mean**	SD
CDA	97.464	3.510	13.376	4.351	99.740	1.553	4.360	3.589
NCDAEE$_{\delta=0}$	98.336	3.262	4.219	3.141	9.756	28.873	14.098	1.800
NCDAEE$_{\delta=10}$	98.912	2.605	5.552	2.770	23.344	41.727	7.834	5.648
NCDAEE$_{\delta=20}$	98.304	2.562	7.460	3.136	89.128	30.867	4.826	3.487
NCDAEE$_{\delta=30}$	97.708	3.136	8.660	3.740	99.736	1.723	4.498	3.502
SM7.1	99.280	1.537	4.325	2.509	58.480	47.983	4.655	4.383
SM88.0	98.320	2.477	11.007	4.251	99.920	0.560	4.387	2.913
SM127.1	97.960	3.225	11.152	4.584	99.520	1.727	4.751	3.153

$$\text{NCDAEE} = \text{ME} + \text{AE}_{w,\delta} + \text{CC} + \text{PN}_n$$

The advantage of NCDAEE is that it can give significantly lower α — faster convergence of transaction prices — and higher allocative efficiency (E_a) than a CDA when populated respectively by homogeneous ZI-C traders and can perform comparably to a CDA when populated by homogeneous GD traders.

We replicated these experiments using JCAT and ran additional ones for the three new mechanisms with similar configurations. The results of these experiments are shown in Table 2.[6] The best result in each column is shaded. We can see that both SM7.1 with ZI-C traders and SM88.0 with GD traders give higher E_a than the best of the existing markets respectively, and both of these increases are statistically significant at the 95% level. Both cases also lead to low α, not the lowest in the column but close to the lowest, and the differences between them and the lowest are not statistically significant at the 95% level. Thus the grey-box approach can generate mechanisms that perform as well in the single market case as the best mechanisms from the literature.

4 Conclusions and future work

This paper describes a practical approach to the automated design of complex mechanisms. The approach that we propose breaks a mechanism down into a set of components each of which can be implemented in a number of different ways, some of

[6] Our results are slightly different from those in [14], but the pattern of these results still holds. In addition, we ran an NCDAEE variant ($\delta = 30$) that was not tested in [14], observing that those with $\delta \leq 20$ do not perform well when populated by GD traders.

which are also parameterized. Given a method to evaluate candidate mechanisms, the approach then uses machine learning to explore the space of possible mechanisms, each composed from a specific choice of components and parameters. The key difference between our approach and previous approaches to this task is that the score from the evaluation is not only used to grade the candidate mechanisms, but also the components and parameters, and new mechanisms are generated in a way that is biased towards components and parameters with high scores.

The specific case-study that we used to develop our approach is the design of new double auction mechanisms. Evaluating the candidate mechanisms using the infrastructure of the TAC Market Design competition, we showed that we could learn mechanisms that can outperform the standard mechanisms against which learning took place and the best entries in past Market Design competitions. We also showed that the best mechanisms we learned could outperform mechanisms from the literature even when the evaluation did not take place in the context of the Market Design game. These results make us confident that we can generate robust double auction mechanisms and, as a consequence, that the grey-box approach is an effective approach to automated mechanism design.

Now that we can learn mechanisms effectively, we plan to adapt the approach to also learn trading strategies, allowing us to co-evolve mechanisms and the traders that operate within them.

Acknowledgments

This work is partially supported by NSF under grant IIS-0329037, *Tools and Techniques for Automated Mechanism Design*, and EPSRC under grants GR/T10657/01 and GR/T10671/01, *Market Based Control of Complex Computational Systems*. We thank the entrants to the Market Design game for releasing binaries of their market agents and the reviewers for their valuable comments.

References

1. K. Cai, E. Gerding, P. McBurney, J. Niu, S. Parsons, and S. Phelps. Overview of CAT: A market design competition. Technical Report ULCS-09-005, Department of Computer Science, University of Liverpool, Liverpool, UK, 2009. Version 2.0.
2. D. Cliff. Evolution of market mechanism through a continuous space of auction-types. Technical Report HPL-2001-326, Hewlett-Packard Research Laboratories, Bristol, England, Dec. 2001.
3. D. Cliff and J. Bruten. Minimal-intelligence agents for bargaining behaviours in market-based environments. Technical Report HPL-97-91, Hewlett-Packard Research Laboratories, Bristol, England, Aug. 1997.
4. V. Conitzer and T. Sandholm. Automated mechanism design: Complexity results stemming from the single-agent setting. In *Proceedings of the Fifth International Conference on Electronic Commerce (ICEC'03)*, pages 17–24, Pittsburgh, PA, USA, September 2003. ACM Press.
5. V. Conitzer and T. Sandholm. Incremental mechanism design. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1251–1256, Hyderabad, India, January 2007.

6. I. Erev and A. E. Roth. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *The American Economic Review*, 88(4):848–881, September 1998.
7. S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22:1–29, 1998.
8. D. K. Gode and S. Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, 1993.
9. V. Krishna. *Auction Theory*. Academic Press, San Diego, CA, 2002.
10. K. A. McCabe, S. J. Rassenti, and V. L. Smith. Designing a uniform price double auction. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories and Evidence*, Santa Fe Institute Studies in the Sciences of Complexity, chapter 11, pages 307–332. Westview Press, Perseus Books Group, Cambridge, MA, 1993.
11. J. Niu, K. Cai, P. McBurney, and S. Parsons. An analysis of entries in the First TAC Market Design Competition. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT)*, Sydney, Australia, December 2008.
12. J. Niu, K. Cai, and S. Parsons. A grey-box approach to automated mechanism design. *Computing Research Repository (CoRR)*, abs/1002.0378, 2010.
13. J. Niu, K. Cai, S. Parsons, E. Gerding, and P. McBurney. Characterizing effective auction mechanisms: Insights from the 2007 TAC Mechanism Design Competition. In Padgham, Parkes, Müller, and Parsons, editors, *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 1079–1086, Estoril, Portugal, May 2008.
14. J. Niu, K. Cai, S. Parsons, and E. Sklar. Reducing price fluctuation in continuous double auctions through pricing policy and shout improvement rule. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, pages 1143–1150, Hakodate, Japan, May 2006.
15. S. Phelps, M. Marcinkiewicz, S. Parsons, and P. McBurney. A novel method for automatic strategy acquisition in n-player non-zero-sum games. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, pages 705–712, New York, NY, USA, 2006. ACM Press.
16. S. Phelps, P. McBurney, S. Parsons, and E. Sklar. Co-evolutionary auction mechanism design: a preliminary report. In *Proceedings of the Workshop on Agent Mediated Electronic Commerce IV (AMEC IV)*, 2002.
17. S. Phelps, S. Parsons, E. Sklar, and P. McBurney. Using genetic programming to optimise pricing rules for a double auction market. In *Proceedings of the Workshop on Agents for Electronic Commerce,*, Pittsburgh, PA, 2003.
18. J. L. Schvartzman and M. P. Wellman. Exploring large strategy spaces in empirical game modeling. In *Proceedings of the 11th Workshop on Agent-Mediated Electronic Commerce*, 2009.
19. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
20. Y. Vorobeychik, D. M. Reeves, and M. P. Wellman. Constrained automated mechanism for infinite games of incomplete information. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, pages 400–407, 2007.
21. W. Walsh, R. Das, G. Tesauro, and J. O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In P. Gmytrasiewicz and S. Parsons, editors, *Proceedings of 2002 Workshop on Game-Theoretic and Decision-Theoretic Agents (GTDT-02)*, Edmonton, Alberta Canada, July 2002. AAAI.
22. P. R. Wurman, M. P. Wellman, and W. E. Walsh. A parametrization of the auction design space. *Games and Economic Behavior*, 35:304–338, 2001.

Modeling Seller Listing Strategies

Quang Duong*, Neel Sundaresan†, Nish Parikh†, and Zeqian Shen†

*Computer Science & Engineering
University of Michigan
Ann Arbor, MI 48109-2121 USA

†eBay Research Labs
San Jose, CA 95125 USA

Abstract. Online markets have enjoyed explosive growths and emerged as an important research topic in the field of electronic commerce. Researchers have mostly focused on studying consumer behavior and experience, while largely neglecting the seller side of these markets. Our research addresses the problem of examining strategies sellers employ in listing their products on online marketplaces. In particular, we introduce a Markov Chain model that captures and predicts seller listing behavior based on their present and past actions, their relative positions in the market, and market conditions. These features distinguish our approach from existing models that usually overlook the importance of historical information, as well as sellers' interactions. We choose to examine successful sellers on eBay, one of the most prominent online marketplaces, and empirically test our model framework using eBay's data for fixed-priced items collected over a period of four and a half months. This empirical study entails comparing our most complex history-dependent model's predictive power against that of a semi-random behavior baseline model and our own history-independent model. The outcomes exhibit differences between different sellers in their listing strategies for different products, and validate our models' capability in capturing seller behavior. Furthermore, the incorporation of historical information on seller actions in our model proves to improve its predictions of future behavior.

1 Introduction

Online marketplaces have fundamentally transformed businesses' sale strategies and consumers' shopping experiences. Online retail in the United States alone generated a revenue of $175 billion in the year of 2007 [6]. The rise of electronic commerce has come with an influx of user-activity data extensively employed in improving shopping sites' profitability in many different ways, such as recommending relevant products or personalizing search queries' results. Researchers have indeed put an ever-increasing focus on developing and deploying machine learning and mining techniques to extract valuable information about users' behavior. However, this surge of research on online marketplaces has been largely limited to buyer-related issues, such as consumer behavior, feedback mechanism, and fraud [4, 10]. As a result, there is a lot to be done in terms of analyzing these marketplaces from the sellers' perspective. Considering the fast growing importance of electronic commerce across many industry sectors, we attempt to bridge this gap between buyer and seller behavior studies with this pilot study on seller listing behavior. Online marketplace websites allow sellers to customize a wide variety of their product listings' features, such as listing duration, price, product description, shipping, pictures, and advertising banners, giving rise to a diverse and innovative ways for sellers to list products on these sites. The diversity of products and sellers and the sheer volume of items listed and sold on eBay motivate our study of analyzing successful eBay sellers' strategies. In particular, our work aims to identify and capture listing strategies that help attract sizable numbers of shoppers on the site. Intuitively, in order to maintain and gain their competitive advantage over other sellers, successful high-volume eBay sellers need to devise and adopt listing strategies through different mechanisms, such as adjusting their prices, number of listings, and listings' features. Moreover, different sellers may employ different listing strategies, which prompts us to study how differences in sellers' strategies translate into differences in sales.

In this paper, we propose a Markov Chain model to capture sellers' strategic choices in listing their products, and apply it to the study of eBay's successful sellers. Specifically, we first process and categorize data on sellers' activities by grouping listings of similar product categories and computing statistical measures that summarize same-group listings. The processed data then constitute building blocks for our Markov chain model of seller behavior. Furthermore, we demonstrate how one can validate our model using testing data, and compare different instances of the model built for different products and sellers. Learning sellers' strategies helps us to identify good seller listing practices that yield high and profitable sales, and moreover, suggest these practices to average sellers as a way to foster better and more efficient listings by the sellers. In addition, we can also estimate how changes in eBay platform impact sellers' strategies and thus can make changes catered to sellers' needs.

1.1 Related Work

Extensive research on online behavior of electronic markets' participants has played a vital role in improving buyers' shopping experience and sellers' profitability. The ability to learn and predict buying behavior is the main focus of studies on recommender systems, which employ data of buyers' activities in tailoring their websites to the users' needs and preferences [1]. Sellers can also utilize models of buyers' clicking and viewing behaviors to suggest and test changes to their websites' features and design to maximize profits [8, 5]. In dynamic pricing markets, such as eBay's auctions, where one buyer's actions may affect other buyers' trades, predictions of buyers' bids and the expected final price of an auctioned product are instrumental to both sellers and buyers' strategy construction. For instance, one solution to the problem of detecting unusual harmful bidding behaviors on eBay's marketplace, such as bid sniping, is data-mining eBay bidders' strategies for categorizing bidding patterns [12]. In addition, when competing in a series of auctions, bidders can learn from past bidding-action data and incorporate its predictions of other bidders' behavior in forming their own bidding strategies that may be the best response to the expected auction outcome [13, 7].

Seller behavior studies are usually of smaller scope and smaller number than those on buyer behavior, as for many cases, only online market designers and operators, such as eBay, Amazon, and Overstock, have the capability and resources to conduct such investigations. As electronic commerce is more amenable to dynamic pricing than traditional commerce is, examining and modeling online stores' pricing strategies prove to be valuable to maximizing sellers' profits [9]. With the advent of eBay come several studies on the product listing's formats that eBay sellers should adopt to maximize their profit. In particular, the "Buy It Now" option, which allows the seller to convert the auction into a fixed price market, encourages higher bids for the auctioned items and attracts more bidders [2]. A more general eBay seller's strategy model that incorporates not only the "Buy It Now" option, but other features such as the auction's length, starting price, and product description, and employs statistical regressions, shows that different types of sellers pursue systematically different strategies on listing their items [3]. A similar study that employs clustering algorithms in grouping sellers based on their characteristics of successful sales, reputation, and sale volume, and their listing choices of price, duration, quantity, and volume, identifies important factors to support sellers' decision and recommend selling practices [11].

Our model of seller listing strategies involves a variety of factors that sellers can decide for their fixed-price listings. However, unlike the aforementioned models [3, 11], which only focus on one particular seller's decisions, our model investigates seller behavior within the context of the entire market, in which sellers compete for buyers, and consequently include market conditions and sellers' relative positions in its representations. Moreover, we account for the time-variant characteristic of sellers' listing strategies by including past listing choices in its state representations, instead of conditioning seller behavior entirely on present conditions. Furthermore, our probabilistic Markov model differs from the previously mentioned models as it induces a probability distribution of listing actions, which can be employed in predicting sellers' listing behavior.

1.2 Contributions

1. We propose using a Markov Chain model for capturing and predicting seller listing strategies on online marketplaces, and moreover, incorporate information on sellers' competition and historical data of listing activities in the model.
2. Our seller listing strategy model demonstrates that eBay sellers indeed do consistently rely on certain strategies to decide how to list their inventory on eBay. We empirically illustrate the top-ranked seller's strategy variations across three different product categories that we examined.
3. Our investigation of different sellers' behaviors in the same product category suggests that the best and second-best sellers adopt similar strategies in some categories, and differ in others. The best seller's strategy diverges significantly from a typical seller's in the categories that we studied.
4. We empirically show that the inclusion of information on sellers' past listing choices indeed enhances the model's predictive power for seller strategies.

In Section 2, we will proceed to describe the seller data obtained from eBay and the data processing methods we employ to extract relevant information for our model construction. Section 3 introduces our Markov Chain model of seller listing strategies, and specifies some of its variations, as well as proposes a validation process for these behavior models. Our empirical study in Section 4 analyzes and compare the performance of our model and some baseline models in predicting seller listing behavior. Section 5 concludes the paper with some remarks and future research directions.

2 Overview of eBay Seller Data

We use data that capture eBay sellers' sale activities over a certain time period in our investigation of the sellers' fixed-price product listing strategies on eBay's marketplace. Each tuple or listing in the data set describes different features and characteristics of a product listing, among which are seller identification, product identification, listing's start date, listing's end date, price, listing's title, and average shipping cost.

Example 1. A listing from seller 'ABC' of a black Apple iPod Nano for a price of $100 and average shipping cost of $7.5, active from 03/20/2009 to 03/27/2009, a listing from the same seller of the same product for a price of $150 and average shipping cost of $0, active from 03/21/2009 to 03/29/2009, and one from 'DEF' of a red Apple iPod Nano for a price of $150 and average shipping cost of $0, active from 03/20/2009 to 03/29/2009, are stored as follows. Note that these products belong to the same product category of Apple iPod Nano, specified in the given product catalog.

Seller ID	Product ID	Start Date	End Date	Price	Title	Shipping
'ABC'	111	03/20/2009	03/27/2009	$100	'silver Apple iPod Nano'	$7.5
'ABC'	222	03/21/2009	03/29/2009	$150	'iPod Nano Apple NEW'	$0
'DEF'	333	03/20/2009	03/29/2009	$150	'iPod Nano Apple NEW RED'	$0

2.1 Product Clustering

As sellers can choose many different ways to describe the same product, the first step of data processing is to group listings of the same product in order to capture and summarize sellers' listing strategies for different products. We use available catalogs to address this problem. We first filter out data entries for bundled items, where two or more items are grouped in one listing. We then compute the matching score $\pi_{L,P}$ between every listing L in the seller data and every product P in the catalog based on its brand name and description. L will then be classified as product P that has the highest matching score $\pi_{L,P}$. A product category $\hat{P}$ consists of products P of different brands and models. For instance, iPhone charger and Blackberry charger belong to the product category "charger." Our study assumes that sellers adopt the same strategy for all products in the same product category, as managing each product's listings individually appears to require an enormous amount of detail and effort, especially for a sizable inventory, that we do not expect eBay sellers to be able to afford. In our model, sellers follow a coherent listing strategy for each product category, such as charger, screen protector, battery, and so on. The three listings described in Example 1 will be clustered as the same product although their descriptions and titles are different.

2.2 Data Summarizing

Different sellers may adjust their listing strategies in different time frequency: some may update their listing strategies every day while others may choose to change theirs monthly. We simplify the problem by imposing a fixed time frequency of seller strategy updates. As a listing can only stay active for a maximum of four weeks, and sold items are mostly purchased within two weeks after their start date, we assume that sellers updates their listing strategies every week.

For each product P and for each time interval t of one week, we capture and summarize the state $s^t_{i,P}$ of seller i's own business and the market's conditions for P using the following set of parameters:

- $p^t_{i,P}$: the average price of all listings of P starting in time interval t by i
- rp^t_P: the average price of all listings of P starting in t
- $l^t_{i,P}$: the total number of listings created by seller i in t for product P
- $sh^t_{i,P}$: percentage of free-shipping listings by seller i for product P in t

We only integrate in our models the above parameters, as they are identified as major representative indicators of the market conditions sellers face. Our observation of a significant difference between the rates of successful sales for free-shipping listings and those that charge shipping fees prompts us to consider the percentage of free-shipping listings, rather than the average shipping cost. In Example 1, under the assumptions that seller 'ABC' (index 1) only had those two listings for Apple iPod Nano, and he was the sole seller of this product during the week starting on 03/20/2009, the Nano's market conditions will be: $p^t_{i,P} = \frac{150+100}{2} = \$125; rp^t_P = \frac{150+150+100}{3} = \$133.33; l^t_{i,P} = 2; sh^t_{i,P} = 50\%$.

3 Learning Sellers' Listing Strategies

3.1 Overview

In this section, we detail the construction of predictive Markov Chain models of seller listing behaviors using information abstracted from the seller data, as specified in Section 2. These models will then help verify whether successful eBay sellers do employ and exhibit any consistent strategies in listing their merchandise, and if so, how well these learned models can predict their listing decisions. Figure 1 summarizes how data are processed and employed in constructing eBay sellers' listing strategy models.

3.2 Markov Models of Listing Strategies

For each pair of seller i and product category $\hat{P}$, we construct a corresponding Markov Chain model of the i's listing strategies for $\hat{P}$, under the assumption that sellers make their strategic decisions based solely on what they observe in the present state $\hat{s}^t_{i,\hat{P}}$. Note that this state notation slightly differs from the notation $s^t_{i,P}$ above as $\hat{s}$ is computed over all $P \in \hat{P}$. The assumption that sellers' strategies are coherent across different

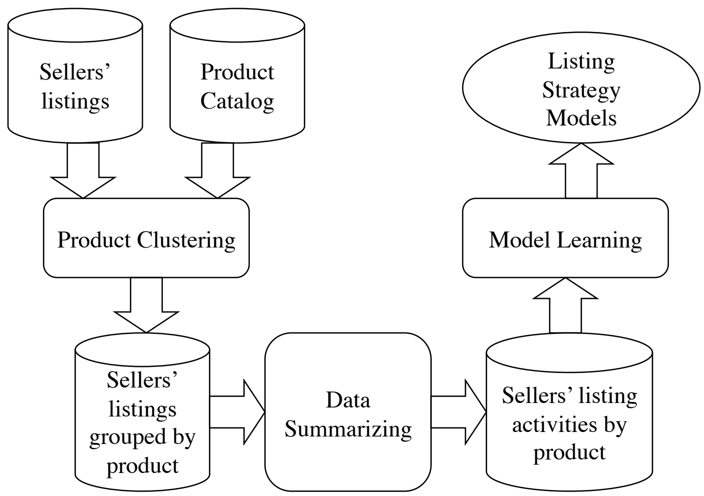


Fig. 1. Seller data are fed to the product clustering unit that uses a given product catalog to categorize the listings. These processed listings are then grouped based on their owners (sellers), and summarized in week long intervals. The model construction phase then learns the transitions between states represented by these statistics.

products P in the same product category $\hat{P}$ allows to us to employ $\hat{s}$, instead of s, for constructing the listing strategy model.

Seller i executes action a^t at time t that transitions its current state to another state. One can certainly add more complexity to this model by incorporating the history of states that sellers have gone through. Given that eBay sellers are businesses of much smaller size than online retail stores , we decided to only include seller i's immediately past action a^{t-1} in the state representation at time t. Moreover, we will demonstrate the benefit of including additional historical information in our models in the empirical study.

State representation Each state of the model contains information about the market condition and the seller's new listings created in a time interval. Because our model's state space is discrete while the state representation s is continuous, we propose to discretize s's parameters to construct the model's state $\bar{s}$ as follows. For simplicity, we drop the subscript notations i and P.

- $\bar{p}^t \in \{0, 1, 2\}$ is the average price level of listings by seller i at time t, in which 0 corresponds to low price, 1 to medium, and 2 to high. We first compute the min and max prices, p_{min} and p_{max}, across all sellers and all time periods for product P, and then divide the interval $[p_{min}, p_{max}]$ prices into 3 equal intervals, corresponding to 0, 1 and 2. Each price value p^t is subsequently assigned a price level.
- $\bar{rp}^t \in \{0, 1, 2\}$ corresponds to lower, equal, and higher prices by seller i relative to the average price at time t. In each time interval t, we compute the average price

across all sellers rp^t and then categorize $(p^t - rp^t)$ into one of the 3 categories that equally divide the interval $[p_{min} - p_{max}, p_{maax} - p_{min}]$

- $\bar{l}^t \in \{0, 1, 2\}$ denotes the volume of listings created by seller i. Since different sellers have different inventory capacity, we compute the min and max number of listings across time periods, but not across sellers. The number of listings is then discretized and categorized as in price.
- $\bar{sh}^t \in \{0, 1\}$ represents the percentage of free-shipping listings by seller i for product P in t, where 0 corresponds to the interval $[0\%, 50\%]$ and 1 to the remaining.

Action representation An action by seller i at time t, a_i^t, explicitly represents the changes in $\bar{p}$, $\bar{l}$, and $\bar{sh}$ for the new listings created by seller i at time t, relative to those created at time $t-1$:

$$a^t = (\bar{p}^t - \bar{p}^{t-1}, \bar{l}^t - \bar{l}^{t-1}, \bar{sh}^t - \bar{sh}^{t-1}) = (\triangle\bar{p}^t, \triangle\bar{l}^t, \triangle\bar{sh}^t). \tag{1}$$

Note that since the seller does not have control over relative price $\bar{rp}$, we treat it as an external parameter.

However, our primitive results showed that in a majority of the states, sellers take no action, i.e. $a = (0, 0, 0, 0)$. This is possibly a result of the state representation's coarseness: a seller's transition from one state to another usually takes several time periods. We therefore propose an alternative action representation where an action $\bar{a}^t$ explicitly specifies the discretized percentage difference between 2 states' prices, listings, shipping and feature:

$$\triangle\bar{p}^t = \begin{cases} 2 & \text{if } p^t - p^{t-1} > 2\epsilon_p \\ 1 & \text{if } p^t - p^{t-1} \in [\epsilon_p, 2\epsilon_p] \\ -1 & \text{if } p^t - p^{t-1} \in [-2\epsilon_p, \epsilon_p] \\ -2 & \text{if } p^t - p^{t-1} < -2\epsilon_p \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The other action parameters $\triangle\bar{l}$ and $\triangle\bar{sh}$ are defined similarly. We choose $\epsilon_p = 0.05$, and $\epsilon_l = \epsilon_{sh} = 0.1$. Note that Equation 2 defines $\triangle\bar{p}^t$ using the original continuous parameter p, instead of the derived discrete parameter $\bar{p}$ in Equation 1. As a result, the new $\triangle\bar{p}$ captures more accurately changes in market conditions and seller listings.

State-action model Imposing the Markov property on our model may limit its expressive power, since sellers may review their past actions' effectiveness and incorporate such evaluations in their decision for the next move. In an effort to incorporate such information, we extend our model's state representation such that a state not only contains information immediately available at time t but also the seller's last action. However, instead of adding the entire past action $\bar{a}^{t-1}$ in the new state representation $\dot{s}^t$, we only incorporate the two more important parameters $\triangle\bar{p}^{t-1}$ and $\triangle\bar{l}^{t-1}$. In other words, $\dot{s}^t = (\bar{s}^t, \triangle\bar{p}^{t-1}, \triangle\bar{l}^{t-1})$. The exclusion of $\triangle\bar{sh}^{t-1}$ in this state-action representation is to balance the model's complexity and its expressiveness.

State-action transition model Since we are mostly interested in the actions the sellers take in certain market environments, we compute and present the transition model from state to action, instead of the typical state-state transition model. In other words, we would like to estimate for the probability distribution of actions a seller may take in a particular state: $\Pr(\bar{a}|\dot{s})$. An edge from a state to an action indicates the probability and the number of times that the seller's taking $\bar{a}$ in state $\dot{s}$, $\Pr(\bar{a}|\dot{s})$ and $c(\bar{a}|\dot{s})$ respectively. Figure 2 displays an example state-action transition model. We will next present the learning method for our state-action transition model, and show empirical study of its prediction performance in comparison with some baseline models.

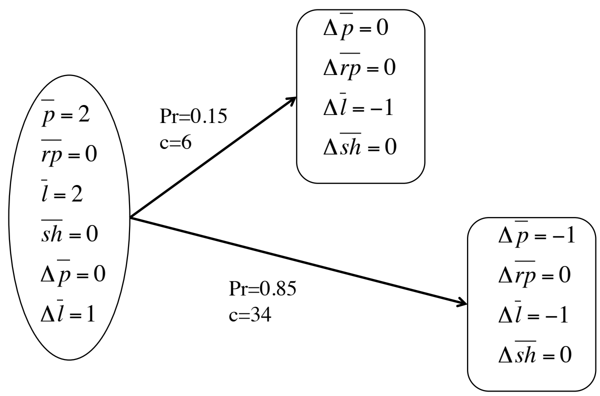


Fig. 2. Markov state-action transition model

3.3 Model Learning

Given seller data $\mathbf{D}$, we are able to divide the listings based on their time interval, products, and sellers, as described in Section 2. For each of these listing sets, we can compute the corresponding state $\dot{s}_D$ and action $\bar{a}_D$, employing the methods in Section 2.2 and Section 3.2. Consequently, we can learn the Markov Chain strategy model from $\mathbf{D}$:

$$c(\bar{a}|\dot{s}) = \sum_{D \in \mathbf{D}} I^{\{\bar{a}_D = \bar{a} | \dot{s}_D = \dot{s}\}}$$

$$\Pr(\bar{a}|\dot{s}) = \frac{\sum_{D \in \mathbf{D}} I^{\{\bar{a}_D = \bar{a} | \dot{s}_D = \dot{s}\}}}{\sum_{D \in \mathbf{D}} I^{\{\dot{s}_D = \dot{s}\}}}$$

where $I^{\{\bar{a}_D = \bar{a} | \dot{s}_D = \dot{s}\}} = 1$ if the state and action induced from data entry $D \in \mathbf{D}$ are $\dot{s}$ and $\bar{a}$ respectively, and 0 otherwise. Similarly, $I^{\{\dot{s}_D = \dot{s}\}} = 1$ if the state is $\dot{s}$ for data entry $D \in \mathbf{D}$.

3.4 Evaluation

In order to evaluate the predictive power of a model, we calculate the testing data $\mathbf{D}'$'s log likelihood induced by the model:

$$L_M = \frac{1}{|\mathbf{D}'|} \sum_{D \in \mathbf{D}'} \log(\Pr(\bar{a}_D | \dot{s}_D)$$

In order to compare two models M_1 and M_2, we subsequently compute the ratio between their log likelihood measures on the same data set $\mathbf{D}'$:

$$L_{M_1,M_2} = \frac{L_{M_1}}{L_{M_2}}. \tag{3}$$

When we compare two strategy models M_1 and M_2 constructed from two different data source $\mathbf{D_1}$ and $\mathbf{D_2}$, we need to compute the corresponding L_{M_1,M_2} (L_{M_2,M_1}) on the same testing data set $\mathbf{D_1'}$ ($\mathbf{D_2'}$) to qualitatively determine the similarity between the strategies that the two models represent. Furthermore, we introduce a new notation $\hat{L}_{M_1,M_2} = 1 - L_{M_1,M_2}$ to evaluate how better in prediction one model (M_1) is against another (M_2). We would like to emphasize that both measures L_{M_1,M_2} and $\hat{L}_{M_1,M_2}$ are proper scoring functions, and thus, encourage accurate predictions [14].

4 Empirical Results

4.1 Experiment Overview

We employ our models to learn seller strategies in the subcategory “Cell phone and PDA accessories.” The seller data set consists of all listings in this subcategory for the span of three months, which consists information of over 100000 listings or auctions. The output data set spans over a period of one and a half months. Three product categories, cell phones’ “charger,” “battery,” and “screen protector," are chosen for our experimental study, as there are a sufficiently large number of listings and reasonably small number of product variations for these categories.

Let S_0 be the top seller, S_1 is the second best seller, and S_2 is the typical seller. S_2 is chosen such that S_2’s sale-through rate is approximately half of S_0’s and S_2’s number of listings is less than one half of S_0’s.

Semi-random Model First, we would like to determine if successful sellers in general do employ any listing strategies by comparing our model with a baseline model M_0, which entails computing the relative measure L_{M,M_0}. The observation that many actions taken are $\bar{a} = (0, 0, 0, 0)$ and sellers tend not to change their listings too dramatically leads to setting $\Pr((0, 0, 0, 0)|\dot{s}) = 0.5$ for all $\dot{s}$ in M_0. Moreover, M_0 assumes that the seller chooses “extreme" actions that have either 2 and/or -2 in their representation tuple $\bar{a}$ with zero probability, and the remaining actions are chosen uniformly randomly. Therefore, M_0 is a pre-determined semi-random model of listing strategies that incorporates some rule-of-thumb knowledge about seller behavior.

History-independent Model Our model, described last in Section 3.2, is a history-dependent model as it incorporates information about both the current market environment and the past market conditions in its state representation $\dot{s}^t = (\bar{s}^t, \triangle\bar{p}^{t-1}, \triangle\bar{l}^{t-1})$. For the purpose of assessing the benefits of including historical information in representing sellers' behavior, we empirically compare this model with a history-independent baseline model M_h, whose state representation is simply $\bar{s}^t$,. In other words, this additional baseline model simply dismisses the importance of past actions in predicting future actions, and bases its predictions solely on information available at the present time period.

4.2 Results

In Table 1, we can observe that our models for sellers S_0 and S_1 outperform the corresponding semi-random baseline model with significant margins of approximately 70%. This outcome confirms that sellers do adopt actively strategies in listing their items for sale, instead of randomly deciding on their listings' specifications or using the same features for their listings.

	Charger	Battery	Screen Protector
$\hat{L}_{M_{s_0},M_0}$	77.9%	69.8%	77.4%
$\hat{L}_{M_{S_1},M_0}$	67.1%	62.8%	57.7%

Table 1. Comparison against the baseline model M_0

We next investigate for the top seller in this subcategory: we are interested in testing if he adopts different strategies for different product categories. Note $\hat{L}_{M_1,M_2|\mathbf{D'_1}}$ corresponds to the testing scenario where we validate models M_1 and M_2using model M_1's testing data set $\mathbf{D'_1}$. The results displayed in Table 2 show that the top seller, S_0, appears to execute relatively different strategies for different product categories.

	$\hat{L}_{M_1,M_2\vert\mathbf{D'_1}}$	$\hat{L}_{M_2,M_1\vert\mathbf{D'_2}}$
Charger vs. Battery	30.1%	25.3%
Charger vs. Screen Protector	36.9%	22.1%
Battery vs. Screen Protector	32.7%	40.6%

Table 2. Strategy models of different product categories for the top seller

Moreover, we are interested in comparing strategies of different sellers for the same product category. Table 3 shows that the best and second-best sellers have similar strategies in the two product categories: charger and battery, but different strategies for the screen protector category. The gap between the top seller and the typical seller is much more prominent: their strategies diverge significantly for both charger and screen protector (There are an insufficient number of observations in the battery category for S_2).

	Charger	Battery	Screen Protector
$\hat{L}_{M_{S_0},M_{S_1}\vert\mathbf{D}'_{\mathbf{S_0}}}$	10%	5.6%	45%
$\hat{L}_{M_{S_0},M_{S_2}\vert\mathbf{D}'_{\mathbf{S_0}}}$	69%	N/A	60%

Table 3. Strategy models of different product categories for the top seller

The next study assesses the effect of incorporating past actions in our models' state representations by comparing their performances with that of history-independent models M_h, as previously outlined. Table 4 shows that our models consistently outperform history-independent models by 60% to 70%, affirming the benefits of including information about past actions in capturing and predicting sellers' listing strategies. Note that these differences are smaller than those between M_{s_0} and M_0, which suggests that the history-independent models do possess more predictive power than the semi-random baseline. A further study on models that only incorporate past actions but no present market conditions will help to evaluate more thoroughly the importance of different past and present information sources.

	Charger	Battery	Screen Protector
$\hat{L}_{M_{s_0},M_h}$	76.1%	67.9%	61.2%

Table 4. Comparison against the history-indepedent model M_h

In order to examine more closely these strategies' effectiveness, we proceed to compare S_0 an S_2's strategies in similar state scenarios. We propose to evaluate an action based on the change in sale-through rate, the ratio of successful sales over all listings, and the seller's average revenue per listing. Given that listings created at time t may affect sales of formerly created listings that are still active, it is non-trivial to compute these measures of effectiveness. We decided to include items sold within 2 weeks from time t in the calculation of $\bar{a}^t$'s effectiveness, while including items sold after 2 weeks from their original listing date in the computation of $\bar{a}^{t'}$'s effectiveness, where t' is the time interval where they are sold. For instance, at time $t = 0$, if seller 'ABC' listed 5 items, 2 of which were sold at time $t = 1$ and the other 3 were sold at $t = 4$, his sale-through rate at $t = 0$ is 40%, while his sale through rate at time $t = 4$, assuming that he has nothing else to sell, is 60%.

We observe that when price, relative price and listing are all at the medium level, seller S_0 seems to mix different actions from adjusting price to increasing the number of listings. All these actions produce considerable gains in average revenue, ranging from \$2.5 to \$29.13, with no considerable decline in sale-through rate. At the same time, seller S_2's main action is decreasing listings, which gives him a negligible boost in revenue. When price is low, relative price is medium and listing is high, seller S_0 again appears more effective: decreasing price only helps him to improve sale-through rate and revenue by 0.3% and \$12.5; decreasing listing only does not hurt sale-through rate; increasing both listing and price cause his revenue to grow by \$4.7. On the other hand, seller S_2 mostly increases listing, which leads to a decline in his sale-through rate by 0.35%. These results suggest that seller S_0 executes a wider range of generally more complex actions, which often result in more gains in both his sale-through rate and average revenue. For other states, the difference between their actions is much less pronounced.

5 Conclusions

This paper introduces a Markov Chain strategy model that captures sellers' listing activities, accommodates probabilistic reasoning about their behavior, and enables the inclusion of historical information. We also describe and demonstrate the application of our model in comparing listing strategies from different sellers across different product categories. The empirical results exhibit the difference in listing behavior between sellers of different success levels, as models best predict listing behavior of the users whose data are used in constructing those same models. An investigation on the top seller's strategies reveals that he employs moderately different strategies for different product categories, which illuminates the complexity of seller listing strategies. Furthermore, the experimental section also highlights the model's capability of capturing sellers' listing strategies, in comparison to a semi-random baseline model and a history-independent variation of the model. We empirically show that the incorporation of past listings does significantly improve our model's predictive power. In addition, using sale-through rate and average revenue per listing to measure the effects of an action is the first step to convey and promote its benefits to businesses.

This pilot study definitely has room for further explorations and extensions. First, we would like to extend our study for many different product categories outside of "Cell phone and PDA accessories," and to include more sophisticated baseline models adopted from the regression-based model and clustering-based model described in Section 1.1 [3, 11]. In addition, a study on the effects of the amount of historical data employed in the model's state representation will definitely provide more guidelines on improving the current model. Moreover, incorporating in the current model game-theoretic multiagent reasoning that take into account strategic reasoning among sellers and solve for the market's equilibrium states will help us to understand and capture their behavior more systematically and thoroughly. Although the performance measures employed in our study are proper scoring functions, conveying their implications using business-friendly terminologies will advance the importance of this study and in general the problem of modeling seller behavior.

Bibliography

[1] Gediminas Adomavicius and Er Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749, 2005.

[2] S.T. Anderson, D. Friedman, G. Milam, and N. Singh. Buy it now: A hybrid internet market institution. Technical report, University of California, Santa Cruz, 2004.

[3] S.T. Anderson, D. Friedman, G. Milam, and N. Singh. Seller strategies on ebay: Does size matter? *International Journal of Electronic Business*, 5(6):643–669, 2007.

[4] S. Ba and P.A. Pavlou. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *Management Information Systems Quarterly*, 26(3):243–268, 2002.

[5] S. Bellman, G.L. Lohse, and E.J. Johnson. Predictors of online buying behavior. *Communications of the ACM*, 42:32–38, 1999.

[6] United States Census Bureau. E-stats. Retrieved Jaunary 31, 2009: http://www.census.gov/econ/estats/, 2007.

[7] Y. Kovalchuk. SellerÕs strategies for predicting winning bid prices in online auctions. In *International Conference on Computational Intelligence for Modelling Control*, pages 1–6, Vienna, 2008.

[8] P.M. Lee. Behavioral model of online purchasers in e-commerce environment. *Electronic Commerce Research*, 2(1):75–85, 2002.

[9] B. Leloup and L. Deveaux. Dynamic pricing on the internet: Theory and simulations. *Electronic Commerce Research*, 1(3):265–276, 2001.

[10] P.A. Pavlou and A. Dimoka. The nature and role of feedback text comments in online marketplaces: Implications for trust building, price premiums, and seller differentiation. *Information Systems Research*, 17(4):392, 2006.

[11] A. Pereira, D. Duarte, W. Meira, Jr., and P. Góes. Assessing success factors of selling practices in electronic marketplaces. In *International Conference on Management of Emergent Digital EcoSystems*, pages 261–268, France, 2009.

[12] H. S. Shah, N. R. Joshi, A. Sureka, and P. R. Wurman. Mining ebay: Bidding strategies and shill detection. In *Fourth International Workshop on Mining Web Data for Discovering Usage Patterns and Profiles*, Edmonton, Alberta, 2003.

[13] M.P. Wellman, A. Greenwald, and P. Stone. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*. The MIT Press, 2007.

[14] Robert L. Winkler. Scoring rules and the evaluation of probability assessors. *Journal of the American Statistical Association*, 64(327):1073–1078, 1969.

SHORT PRESENTATIONS

An Automated Mechanism Design Approach for Sponsored Search Auctions with Federated Search Engines

S. Ceppi and N. Gatti

Dipartimento di Elettronica e Informazione, Politecnico di Milano
Piazza Leonardo da Vinci 32, I-20133 Milano, Italy
{ceppi,ngatti}@elet.polimi.it

Abstract. Advertising mechanisms for search engines (i.e., sponsored search auctions) have recently received a lot of attention in the scientific community. Advertisers bid on keywords and, when a user enters keywords for her search, the search engines uses an auction mechanism to select the list of sponsored links to display alongside the search results. In this paper, we make a first attempt to extend the currently available mechanisms for sponsored search auctions to the new paradigms of search computing. According to them, multiple federated domain-specific search engines are integrated by a special search engine (called integrator). The user can enter a multi-domain query that is decomposed by the integrator in single-domain queries and these are singularly addressed to the most appropriate domain-specific search engine. The integrator merges the search results. We propose a business model for this scenario and we develop an economic mechanism for it resorting to the automated mechanism design approach.

1 Introduction

Sponsored search auctions [1, 2] play a prominent role in Internet advertising, generating more than 90% of the search engines' revenues. A large number of theoretical/practical works can be found in the very recent literature. Nevertheless, this market is still largely unexplored and a number of problems are currently open. The functioning of sponsored search auctions is simple. When a user enters keywords into a search engine, several sponsored links related to the entered keywords are displayed alongside the search results, e.g., see [3]. The search engine chooses the sponsored links to display and the ranking over them by using an auction mechanism where the bidders are advertisers and the item over which they bid are keywords. The payment scheme is the *pay-per-click*, i.e., the advertiser pays the search engine only after a user has clicked on its sponsored link.

The most employed auction mechanism in sponsored search auctions is the *generalized second price* (from here on GSP) [4] that is an *ad hoc* extension of the Vickrey auction (from here on VA) [2] to the setting where a set of ranked objects is being sold. In the VA, a winner pays the second highest bid, while, in the GSP, each winner pays an amount equal to its next highest bid. However, as shown in [4], in the GSP the truth telling strategy is not (generally) optimal for the players, as instead it is in the VA. The exact generalization of the VA to the above settings, satisfying the property that the truth-telling strategy is optimal, is similar to the GSP except for the definition of the payments. Although this last mechanism is strategy-proof and could assure a higher degree of the outcome stability, it is not currently adopted in real-world applications.

The available economic mechanisms for sponsored search auctions effectively work with the major general-purpose search engines. However, the recent advancements in the search computing field lead to the definition of novel searching paradigms that rise new challenges and that require extensions of the available auction mechanisms [5, 6]. The main general-purpose search engines crawl the Web and index Web pages, finding the best pages for each specific list of keywords with excellent precision. Anyway, the so-called "deep Web" contains information that is more valuable than that contained in single Web pages and the current general-purpose search engines are not able to discover it. The development of new searching paradigms able to address more complex searches than those addressed to the current search engines and to discover deeper information is currently one of the most interesting challenges in the search computing field. In particular, the emerging paradigm is based on the integration of heterogeneous data sources. According to that, a special search engine (from here on *integrator*) integrates the results produced by multiple domain-specific search engines, e.g., see [7]. The basic idea is the following. The user's search is a multi-domain query. Each multi-domain query is automatically decomposed by the integrator in multiple single-domain queries and each of them is addressed to the most appropriate domain-specific search engine. Obviously, when a query addresses a specific domain, domain-specific search engines works better than general-purpose ones. Once the integrator has received the search results from all the domain-specific search engines, it aggregates them in a unique result. This is shown by using *ad hoc* interfaces that allow the user to explore the search results, adding/removing domains and thus refining the search itself, e.g., see [9].

The search computing field is working exclusively on the searching techniques and is neglecting the business model behind the above scenario (e.g., what kind of contracts will be drawn up between the integrator and the domain-specific search engines?). Currently the commercial use of the search results produced by a search engine is ruled by a contract between the search engine and the publisher prescribing that the publisher must display the list of sponsored links produced by the search engine, e.g., as in [10]. Once a user clicks on a sponsored link, the search engine receives the payment from the corresponding advertiser and gives part of it to the publisher. The payment ratio kept by the search engine is defined by a commercial contract and it is independent of the specific search. On the one hand, the basic idea behind this business model can be "naturally" applied to the above scenario. On the other hand, the contracts between the publisher (in our case the integrator) and the search engines (in our case the domain-specific search engines) must be reconsidered keeping into account that each search engine plays a role in the search process. Our opinion is that the contracts between the integrator and the search engines should be drawn up dynamically, depending on the specific search and, in particular, on the contribution provided by the specific search engine to the search.

In this paper we propose an economic mechanism [11] to rule the contracts between the integrator and the domain-specific search engines. In our proposal, the integrator receives the lists of sponsored links from the domain-specific search engines and merges them in a unique list. In the merging process, the integrator keeps into account the advertisers' bids and click probabilities related to the list of each domain-specific search engine in order to generate the list of sponsored links that gives the largest expected utility. Being the information on the advertisers' bids and click probabilities private for each domain-specific search engine, we must produce the appropriate incentives to the domain-specific search engines not to misreport such a information. We formulate this problem as a single-stage mechanism design problem [2] and we discuss the desired properties. We show that the domain-specific search engines present interdependent valuations due to the aggregation of their information. We study it by using the automated mechanism design approach [12]. It provides a flexible tool to design mechanisms on-the-fly and allows one to customize each problem by varying different objective functions and adding/removing possible constraints over the contracts. However, the hardness of solving an automated mechanism design problem allows us to solve in exact way only small settings with a few of search engines and advertisers. For large settings, approximate (anytime) algorithms can be developed to produce a sub-optimal solutions by a given deadline.

Finally, we remark that the possibility to integrate multiple lists of sponsored links provides, in our opinion, two advantages. First, the integrator can target at best the advertisement to the user by exploiting multiple information sources (i.e., the domain-specific search engines) and the user's feedback during her exploration of the search. Second, this paradigm allows domain-specific search engines to federate together and to be real competitors to the major general-purpose search engines. This could open new economic opportunities for online advertising.

The rest of the paper is structured as follows. In Section 2, we discuss the state of the art related to the sponsored search auctions, the multi-domain search computing, and the automated mechanism design. In Section 3, we propose a business model for the scenario we study, we formally state an economic mechanism, we discuss the desired properties, and we formulate the problem of designing the mechanism as an automated mechanism design problem. In Section 4, we discuss some examples. Section 5 concludes the paper.

2 State of the Art

We introduce a formal model of sponsored search auctions in Section 2.1, we discuss the multi-domain search computing paradigm in Section 2.2, and we survey the idea behind the automated mechanism design in Section 2.3.

2.1 Sponsored Search Auctions

The formal model of a sponsored search auction is constituted by m *items* (i.e., the ranked set of slots for sponsored links given a specific keyword) sold by the *auctioneer* (i.e., the search engine) and by a set of *bidders* (i.e., the advertisers) $N = \{1, 2, \ldots, n\}$ where $n \geq m$. Each advertiser can submit a bid constituted by a value per click on the advertisement for a keyword. The bid is unique for all the slots. Being the payment scheme pay-per-click, each advertiser pays nothing if its sponsored link is displayed but not clicked by the user. Instead, in the case the user clicks on the link, the advertiser is required to pay an amount of money that is non-larger than its bid. The exact value of the payment is carried out by the auction mechanism.

The search engine assigns to each bidder a click probability called *click-through-rate* (from here on CTR) [2]. CTR depends on various factors including the probability that users click on advertisement, the relevance of the bidders' advertisement, and so on. Formally, we denote by $\alpha_{i,j}$ the probability that the advertiser i's sponsored link is clicked when it is displayed on the j-th highest slot. Usually, these probabilities are supposed to be separable into two independent components, where the first component refers only to the advertiser and the second component refers only to the position of the slot. Formally, $\alpha_{i,j} = \alpha_{a_i} \cdot \alpha_{r_j}$ where α_{a_i} is the probability that advertiser i is clicked independently of the specific slot in which its sponsored link is displayed and α_{r_j} is the probability assigned to the j-th highest slot independently of the specific advertiser. The common assumption is that $\alpha_{r_1} > \alpha_{r_2} > \ldots > \alpha_{r_k}$. Currently, Google ranks the advertisements by using a separable CTR.

The GSP auction extends the VA as follows. We call b_i the bid submitted by advertiser i. The auctioneer ranks the bids in decreasing order in the value $\alpha_{a_i} \cdot b_i$. For the sake of simplicity suppose that, given α_{a_i} and α_{a_h}, if $i < j$, then $\alpha_{a_i} \cdot b_i > \alpha_{a_h} \cdot b_h$. The advertiser with $\alpha_{a_1} \cdot b_1$ will be displayed at the first slot, the advertiser with $\alpha_{a_2} \cdot b_2$ will be displayed in the second slot, and so on. Once the sponsored link displayed in the i-th position is clicked, the i-th advertiser pays $p_i = \frac{\alpha_{a_{i+1}} \cdot b_{i+1}}{\alpha_{a_i}}$ to the search engine (while the other search engines' payments are zero).

2.2 Multi-Domain Search Computing

The new advancements produced in the field of the search computing are directed to fill the gap between general purpose search engines and domain-specific search engines. General purpose search engines work well in finding Web pages related to the entered keywords, but are unable to find information spanning multiple topics. Domain-specific search engines work well in finding structured information spread on multiple Web pages related to a specific domain, but their expertise is clearly limited to a given domain. An expert user can perform several independent searches and then manually combine the results, but the missing aspect is the ability of joining the results of each search process so as to build a collective answer.

The emerging search computing paradigm prescribes that a user can compose her search as a multi-domain query and this is addressed to a federated search platform. Each single-domain query is addressed to the most appropriate domain-specific search engine. The search results produced by the domain-specific search engines are merged by an integrator through syntactic and/or semantic joint methods [13]. The merging of the search results is only the first feature provided by the integrator. Indeed, this allows a user to explore the search by changing modularly the dimensions of the multi-domain query. In particular, a user can add or remove dimensions, can manipulate results via composition and aggregation, and can reorder results [14].

Search results are shown through tables where the columns are the dimensions and the rows are items of the search. In the cells of the tables, information related to the specific item and dimension is reported. An example is Google Squared [9]. The definition of a business model for federated search engines and the integrator is currently an open problem.

2.3 Automated Mechanism Design

Classical mechanism design provides general mechanisms, satisfying some notion of non-manipulability and maximizing some objective. The most famous general mechanisms, VCG and dAGVA [2], only maximize social welfare. For almost all the other social choice functions, there is no known mechanism that implements them. For example, revenue-maximizing mechanisms are only known for very restricted settings, such as the Myerson's expected revenue maximizing auction for selling a single item, and the Maskin and Riley's expected revenue maximizing auction for selling multiple identical units of an item [2]. In practice, a designer often has prior information over agents' types and only needs to design a mechanism suitable for her particular context. In the automated mechanism design approach, a mechanism is designed automatically for the specific preference aggregation problem. The mechanism design problem can be formulated as an optimization problem where the input is characterized by the number of agents, the agents' possible types (preferences), and the aggregator's prior probability distributions over the agents' types and the output is a non-manipulable mechanism that is optimal with respect to some objective.

3 The Economic Mechanism

We propose a business model in Section 3.1 for multi-domain search computing scenarios in which domain-specific search engines are federated, we formulate an economic mechanism supporting our business model in Section 3.2, we discuss the desired properties of our mechanism in Section 3.3, and we state the problem of designing the social choice function and the payments in our mechanism as an automated mechanism design problem in Section 3.4.

3.1 The Proposed Business Model

The commercial use of the search results is currently ruled by a contract between the search engine and the publisher prescribing that the publisher must display the list of sponsored links alongside the search results, e.g., see [3]. If the user clicks on a sponsored link, the corresponding advertiser pays the search engine which, in its turn, gives a part of the revenue to the publisher. The ratio of revenue kept by the search engine does not depend on the specific search and is established by the contract. The values of the advertisers' bids and the corresponding click probabilities constitute private information of the search engine and are hidden to the publisher.

This business model can be easily applied to the case in which an integrator merges the search results of multiple search engines. The idea is:

- the integrator merges the lists of sponsored links returned by each single domain-specific search engine,
- the payments from the domain-specific search engines to the integrator depends on the specific search.

The crucial issue is the development of techniques that allows the integrator to produce the list that maximizes a given objective function (e.g., the expected revenue of the integrator or the expected revenue of a combination of specific-domain search engines). We propose an economic mechanism to govern the merging of the lists where:

- the domain-specific search engines communicate to the integrator their private information (values of the bids and click probabilities) concerning the sponsored links related to their own list;
- the integrator produces an estimation of the click probability for each sponsored link as a function of the received click probabilities (e.g., averaging the click probabilities of a sponsored link over the different search engines);
- the integrator selects the list of sponsored links in order to maximize a given objective function (e.g., the integrator's expected utility or the cumulative expected utility) and produces the appropriate incentives (i.e., payments) for each domain-specific search engine to make them not misreport their true values;
- the integrator keeps into account how the user explores the search results in estimating the click probabilities and produces a new list of sponsored links every time the user add/remove domains.

In what follows, we formally state the economic mechanism.

3.2 The Formal Mechanism

We consider a direct mechanism [2] $\mathcal{M}(X, S, \Theta, V, f, p)$ where the agents are the domain-specific search engines (from here on we omit "domain-specific") and the integrator acts as auctioneer. We denote by X the set of alternatives. For the sake of presentation, we formally define X below, after having defined S and

Θ. We denote by S the set of search engines. We denote by A the overall set of advertisers and by $A(s)$ with $s \in S$ the set of advertisers appearing in the list of sponsored links of search engine s. Each advertiser $a \in A(s)$ is characterized by a bid b and a click probability α that are private information for search engine s. The type $\theta_s \in \Theta_s$ of search engine s specifies a value for b and a value for α for all the advertisers $a \in A(s)$. Set Θ is composed by all the sets Θ_ss and θ denotes the profile of search engines' types. We assume that we have a probabilistic prior over Θ_s and we can represent it as a set of independent probability distributions, each over a specific advertiser $a \in A(s)$. In particular, we denote by $\Theta_{s,a}$ the set of possible types of advertiser a appearing in the list of sponsored links of search engine s and we denote by $\theta_{s,a} \in \Theta_{s,a}$ the type. We introduce the functions $b(\theta_{s,a}) : \Theta_{s,a} \to \mathbb{R}$, returning the bid submitted by advertiser a to search engine s related to type $\theta_{s,a}$, and $\alpha(\theta_{s,a}) : \Theta_{s,a} \to [0, 1]$, returning the click probability of advertiser a in the sponsored link list of search engine s related to type $\theta_{s,a}$. Generally, an advertiser a can appear in the lists of more than one search engine with different values of bid and click probability, i.e., $b(\theta_{s,a})$ can be different from $b(\theta_{s',a})$, as well as $\alpha(\theta_{s,a})$ can be different from $\alpha(\theta_{s',a})$. We denote by $\omega(\theta_{s,a})$ the probability that the actual type of advertiser a for search engine s is $\theta_{s,a}$. Therefore, the type of search engine s is a tuple specifying the type related to each advertiser $a \in A(s)$, e.g., $\theta_s = (\theta_{s,1}, \ldots, \theta_{s,|A(s)|})$. The probability $\omega(\theta_s)$ related to θ_s is defined as $\omega(\theta_s) = \prod_{\theta_{s,a} \in \theta_s} \omega(\theta_{s,a})$.

Now we focus on set X. An alternative $x \in X$ specifies a winner for each slot of the list of sponsored links displayed by the integrator. We assume that the number of available slot is fix and it is equal to k. A winner is identified by a pair (s, a), that is advertiser a related to the sponsored link list of search engine s. This is because the same advertiser $a \in A$ can appear in the sponsored link lists of different search engines. We need to specify the search engine to which the sponsored link belongs because such a search engine will be paid by the advertiser and a may have submitted different bids to different search engines. Formally, $x = \langle (s, a), \ldots, (s', a') \rangle$, where the first element of x specifies the winner of the first slot, the second element of x specifies the winner of the second slot, and so on. The unique constraint is that a sponsored link can appear only in one slot, that is, for all a and a' appearing in x in different positions, we have $a \neq a'$.

We denote by V the set $V = \{v_s : s \in S\}$ where $v_s : X \to \mathbb{R}$ denotes the valuation function of search engine s. Given x, if s does not appear in x, then $v_s(x) = 0$. Instead, if s appears in x, $v_s(x)$ returns the s's expected valuation over x defined as: for each $(s, a) \in x$, the s's expected valuation is the product between the a's click probability and the valuation that s receives when the a's sponsored link is clicked. Before formally stating $v(x)$, we focus on these two elements. First, we consider the a's click probability. It is a function of $\alpha(\theta_{s',a})$ for all s' where $\theta_{s',a}$s are the reported types. That is, the integrator produces an estimation of such click probability, denoted by $\overline{\alpha}(a)$, aggregating the click probabilities over the advertiser a of all the search engines s such that $a \in A(s)$. In estimating $\overline{\alpha}$, the integrator can exploit several parameters, e.g., it can assign different weights to different search engines or excluding search engines, once the user has removed the corresponding dimension. In our work we use a simple estimator: the average $\overline{\alpha}(a) = \frac{\sum_{s \in S: a \in A(s)} \alpha(\theta_{s,a})}{|A|}$. Second, we consider the valuation that s receives when the a's sponsored link is clicked. In this work we assume that advertiser a pays to s exactly its bid. Essentially, we assume a first-price approach. We make this assumption for simplicity because we are focusing only on the interaction between the integrator and the search engines. In future works, we shall consider also the interaction between the search engines and the advertisers. Now, we are in the position to formally state $v_s(x)$ as $v_s(x) = \sum_{(s,a) \in x} \overline{\alpha}(a) \cdot b(\theta_{s,a})$ where $\theta_{s,a}$s are the true types of s.

In mechanism $\mathcal{M}$, f and p define respectively the social choice function and the search engines' payments. More precisely, f is a function $f : \Theta \rightarrow X$ that given the type of all the search engines returns an alternative, while $p : \Theta \rightarrow \mathbb{R}^{k \cdot |S|}$, where k is the number of slots, returns the payment for each search engine for each situation in which one sponsored link is clicked. We use a quasi-linear setting where the utility of a search engine is equal to the difference between its valuation and the payment. We want to design f and p such that $\mathcal{M}$ satisfies a set of properties.

3.3 Required Properties

Before discussing the properties we require that our mechanism satisfies, we underline that, in the general case, search engines in mechanism $\mathcal{M}$ present interdependent valuations [15, 16]. This is because $v_s(x)$ depends on $\overline{\alpha}(a)$ that, in its turn, depends on $\theta_{s',a}$s for all s'. Exclusively when $A(s) \cap A(s') = \emptyset$ for all $s \neq s'$, $\overline{\alpha}(a)$ depends only on the type of the search engine s such that $a \in A(s)$ and therefore the search engines' valuations are not interdependent. We require the following properties.

(Ex-post) Individual rationality. For every x such that $f(\theta) = x$, we require that, for every realization of x, the utility of all the search engines is non-negative. This requires that, given x, whenever a sponsored link a related to search engine s is clicked, s does not pay the mechanism more than $b(\theta_{s,a})$, while the payments of all the other search engines s's are non-positive.

(Ex-post Nash and Bayesian) Incentive compatibility. We require the implementation of f either in *ex-post* Nash or in Bayes-Nash equilibrium. Therefore, we require that each search engine reporting its true type is an optimal strategy. (We use *ex-post* Nash implementation instead of dominant strategy implementation because in our problem valuations are interdependent. We recall that *ex-post* Nash and dominant strategy implementations are always the same except when the valuations are interdependent.)

(Ex-post) Weak budget balance. For every x such that $f(\theta) = x$, we require that, for every realization of x, the cumulative payments of the search engines is non-negative. This requires that, given x, whenever a sponsored link a appearing in the sponsored link list of search engine s is clicked, the sum of search engines' payments excluded s is not smaller than $-b(\theta_{s,a})$. The revenue of the integrator is equal to the cumulative payments of the search engines.

Optimality. We consider several objective functions: the maximization of the *ex-ante* expected utility of the integrator (defined as the sum of the expected payments of the search engines), the maximization of the *ex-ante* cumulative expected valuations of the search engines, the maximization of the *ex-ante* cumulative expected utilities of the search engines, and the maximization of the *ex-ante* expected utility of a specific search engine. The choice of the objective function depends on the specific contract.

We remark that in the general case (interdependent valuations) our mechanism cannot be efficient. Indeed, with interdependent valuations and multiple signals a one-stage mechanism may not be incentive compatible and efficient [17] (with two-step mechanism is instead possible to have efficient incentive compatible mechanisms [16]; we shall explore this option in future works). It can be easily shown that even in the basic case in which there are two search engines and the same advertiser for both search engines, and only the click probabilities are uncertain, a one-stage mechanism may not be incentive compatible and efficient. Indeed, in the case each player has a single signal, three conditions need to be satisfied in order to make a mechanism incentive compatible and efficient, see [17]. One of these requires that $\frac{\partial v_s}{\partial \theta_s} > \frac{\partial v_{s'}}{\partial \theta_s}$ for all s, s'. Generally this condition is not satisfied in our basic case. As a result, f cannot be easily defined as the argument maximizing the social welfare, as instead it is possible for efficient mechanism.

3.4 The Automated Mechanism Design Formulation

We formulate our mechanism as an automated mechanism design [12] problem. We represent f as a collection $f_{s,a,\theta,r} \in \{0,1\}$ where $f_{s,a,\theta,r} = 1$ means that advertiser a related to search engine s is assigned position r when type profile of the search engines is θ. Index r belongs to the range $R = \{1,\ldots,k\}$. For simplicity, for $a \in A \setminus A(s)$ we set $f_{s,a,\theta,r} = 0$, $b(\theta_{s,a}) = 0$, and $\omega(\theta_{s,a}) = 0$. We introduce the constraints in a mathematical programming fashion. Initially, we constrain every sponsored link a to appear at most in one position r:

$$\sum_{r\in R} \sum_{s\in S: a\in A(s)} f_{s,a,\theta,r} \leq 1 \qquad \forall \theta \in \Theta, \forall a \in A \tag{1}$$

We constrain that for each position r there is exactly one sponsored link:

$$\sum_{s\in S} \sum_{a\in A} f_{s,a,\theta,r} = 1 \qquad \forall \theta \in \Theta, \forall r \in R \tag{2}$$

We denote by $p_{s,\theta,r}$ the payment of the search engine s when the r-th sponsored link is clicked and the type profile is θ. To make the mathematical programming formulation easier, we divide $p_{s,\theta,r}$ in payments concerning the single advertisers, one for each of them. We denote these payments by $p_{s,a,\theta,r}$ and we define $p_{s,\theta,r} = \sum_{a\in A(s)} p_{s,a,\theta,r}$ for all $s \in S, \theta \in \Theta$, and $r \in R$. The *ex-post* individual rationality constraints make each search engine to pay no more than its valuation, formally, we have:

$$p_{s,a,\theta,r} \leq b(\theta_{s,a}) \cdot f_{s,a,\theta,r} \qquad \begin{array}{r}\forall s \in S, \forall a \in A(s),\\ \forall \theta \in \Theta, \forall r \in R\end{array} \tag{3}$$

We require further that, if the sponsored link of advertiser a is displayed at the r-th position when the type profile is θ, then its payment is non-negative, formally, we have:

$$p_{s,a,\theta,r} \geq -M \cdot \left(1 - \sum_{a' \in A(s)} f_{s,a',\theta,r}\right) \qquad \begin{array}{r}\forall s \in S, \forall a \in A(s),\\ \forall \theta \in \Theta, \forall r \in R\end{array} \tag{4}$$

where M is an arbitrarily large number. With abuse of notation we denote by $\overline{\alpha}(a,r)$ the probability that the integrator assigns to a when it is displayed at the r-th position. We represent $\theta = (\theta_s, \theta_{-s})$ where θ_s is the type profile of search engine s and θ_{-s} is the type profile θ once excluded θ_s. The *ex-post* Nash incentive compatibility constraints are:

$$\begin{array}{r}\sum_{r \in R} \sum_{a \in A} \left(b(\theta_{s,a}) \cdot f_{s,a,(\theta_s,\theta_{-s}),r} - p_{s,a,(\theta_s,\theta_{-s}),r}\right) \cdot \bar{\alpha}(a,r) \geq \\ \sum_{r \in R} \sum_{a \in A} \left(b(\theta_{s,a}) \cdot f_{s,a,(\theta'_s,\theta_{-s}),r} - p_{s,a,(\theta'_s,\theta_{-s}),r}\right) \cdot \bar{\alpha}(a,r)\end{array} \qquad \begin{array}{r}\forall s \in S,\\ \forall \theta \in \Theta,\\ \forall \theta'_s \in \Theta_s\end{array} \tag{5}$$

The Bayesian incentive compatibility constraints are:

$$\begin{array}{l}\sum_{\theta_{-s}} \sum_{r \in R} \sum_{a \in A} \Bigg(\Big(b(\theta_{s,a}) \cdot f_{s,a,(\theta_s,\theta_{-s}),r} - \\ \qquad\qquad -p_{s,a,(\theta_s,\theta_{-s}),r}) \cdot \bar{\alpha}(a,r)\Big) \cdot \prod_{s' \in S/\{s\}} \omega(\theta_{s'})\Bigg) \geq \\ \sum_{\theta_{-s}} \sum_{r \in R} \sum_{a \in A} \Bigg(\Big(b(\theta_{s,a}) \cdot f_{s,a,(\theta'_s,\theta_{-s}),r} - \\ \qquad\qquad -p_{s,a,(\theta'_s,\theta_{-s}),r}\Big) \cdot \bar{\alpha}(a,r)) \cdot \prod_{s' \in S/\{s\}} \omega(\theta_{s'})\Bigg)\end{array} \qquad \begin{array}{r}\forall s \in S,\\ \forall \theta_s \in \Theta_s,\\ \forall \theta'_s \in \Theta_s\end{array} \tag{6}$$

The *ex-ante* weak budget balance constraints are:

$$\sum_{s \in S} \sum_{a \in A} p_{s,a,\theta,r} \geq 0 \qquad \forall \theta \in \Theta, \forall r \in R \tag{7}$$

In what follow we point out the possible objective functions for our model. The maximization of the integrator's *ex-ante* expected utility is:

$$\max \quad \sum_{\theta \in \Theta} \left(\sum_{r \in R} \sum_{s \in S} \sum_{a \in A} p_{s,a,\theta,r} \cdot \overline{\alpha}(a,r)\right) \cdot \prod_{\theta_s \in \theta} \omega(\theta_{s,a}) \tag{8}$$

The maximization of the cumulative search engines' *ex-ante* expected valuations is:

$$\max \quad \sum_{\theta \in \Theta} \left(\sum_{r \in R} \sum_{s \in S} \sum_{a \in A} b(\theta_{s,a}) \cdot f_{s,a,\theta,r} \cdot \bar{\alpha}(a,r)\right) \cdot \prod_{\theta_s \in \theta} \omega(\theta_{s,a}) \tag{9}$$

The maximization of the cumulative search engines' *ex-ante* expected utility is:

$$\max \sum_{\theta \in \Theta} \left(\sum_{r \in R} \sum_{s \in S} \sum_{a \in A} \left(b(\theta_{s,a}) \cdot f_{s,a,\theta,r} - p_{s,a,\theta,r} \right) \cdot \bar{\alpha}(a,r) \right) \cdot \prod_{\theta_s \in \theta} \omega(\theta_{s,a}) \tag{10}$$

The maximization of the search engine s's *ex-ante* expected utility is:

$$\max \sum_{\theta \in \Theta} \left(\sum_{r \in R} \sum_{a \in A} \left(b(\theta_{s,a}) \cdot f_{s,a,\theta,r} - p_{s,a,\theta,r} \right) \cdot \bar{\alpha}(a,r) \right) \cdot \prod_{\theta_s \in \theta} \omega(\theta_{s,a}) \tag{11}$$

It can be easily observed that all the above constraints and objective functions are linear. That is, our formulation is linear mixed integer.

4 Some Examples

We briefly analyze in Section 4.1 the business model currently adopted by AdSense and we show an example of our mechanism in Section 4.2.

4.1 Single Search Engine Case

We consider the business model currently adopted by AdSense where a publisher displays the search engine's sponsored links alongside the search results and, if a link is clicked, the search engine pays a fix ratio of its revenue to the publisher. We study this situation with our framework by introducing a constraint over the payment of the search engine to the publisher (in our case the integrator). Formally, we need to impose that $p_{s,a,\theta,r} = \rho \cdot b(\theta_{s,a})$ with $\rho \in [0,1]$ if $f_{s,a,\theta,r} = 1$. It can be easily shown that no incentive compatible mechanism can be designed in general. Consider the following example.

We have a single search engine s and two possible advertisers. The types related to the first advertiser are $\theta_{s,1} \in \{\theta^1_{s,1}, \theta^2_{s,1}, \theta^3_{s,1}\}$ with $b(\theta^1_{s,1}) = 0.4$, $b(\theta^2_{s,1}) = 0.5$, $b(\theta^3_{s,1}) = 0.6$, and $\alpha(\theta^1_{s,1}) = \alpha(\theta^2_{s,1}) = \alpha(\theta^3_{s,1}) = 0.3$. The types related to the second advertiser are $\theta_{s,2} \in \{\theta^1_{s,2}, \theta^2_{s,2}, \theta^3_{s,2}\}$ with $b(\theta^1_{s,2}) = 0.5$, $b(\theta^2_{s,2}) = 0.6$, $b(\theta^3_{s,2}) = 0.7$, and $\alpha(\theta^1_{s,2}) = \alpha(\theta^2_{s,2}) = \alpha(\theta^3_{s,2}) = 0.2$. The probabilities $\omega(\cdot)$s can be arbitrary. It can be shown that there is no incentive compatible mechanism. Easily, when the true type of search engine s is $(\theta^3_{s,1}, \theta^3_{s,2})$, its optimal strategy is to report $(\theta^1_{s,1}, \theta^1_{s,2})$ independently of the implemented social choice function and independently of the value of ρ. (Practically, our mathematical programming formulation coding the automated mechanism design problem results to be infeasible.) In order to remove this impossibility, we need to remove the constraint on $p_{s,a,\theta,r} = \rho \cdot b(\theta_{s,a})$.

4.2 Analyzing a Case Study

We consider a scenario where an integrator aggregates two domain-specific search engines. The domain of the first search engine is music concerts, while the domain of the second search engine is hotels. A demo of integrator can be found at [7]. We report the user interface of the demo in Fig. 1. We consider a simple example

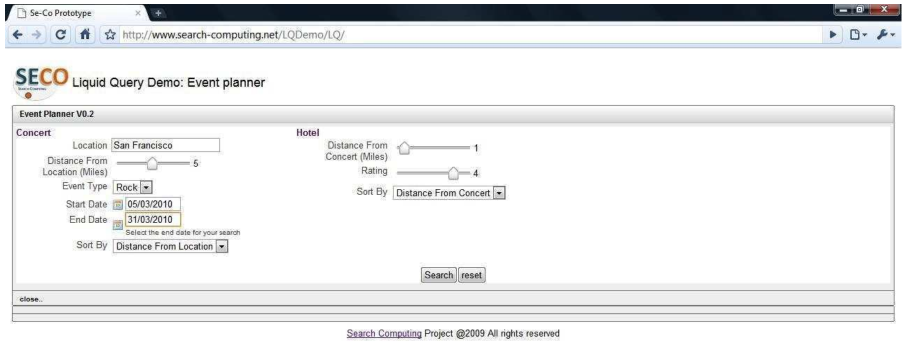


Fig. 1. An example of user interface of an integrator.

and we discuss the functioning of our mechanism (larger scenarios require long time and cannot be solved in exact way in practical applications).

We assume that the user searches for:

– (on the first domain) a concert at Toronto at May 9-15 2010,
– (on the second domain) an hotel at Toronto for the same range of days.

We assume that the first domain-specific search engine returns three sponsored links. We report the Bayesian prior over them in Tab 1. We assume that the second domain-specific search engine returns three sponsored links. We report the Bayesian prior over them in Tab 2. We assume that the number of available slots for sponsored links displayed by the integrator are two.

advertiser	taxi_service				restaurant1				restaurant2	
bid (b)	0.40 €	0.40 €	0.50 €	0.50 €	0.65 €	0.65 €	0.70 €	0.70 €	0.60 €	0.70 €
click probability (α)	0.020	0.030	0.020	0.030	0.040	0.050	0.040	0.050	0.035	0.035
type probability (ω)	0.25	0.25	0.25	0.25	0.30	0.30	0.20	0.20	0.40	0.60

Table 1. Bayesian prior over the sponsored link list returned by the first domain-specific search engine.

advertiser	restaurant1				tourist_office		taxi_service			
bid (b)	0.50 €	0.50 €	0.60 €	0.60 €	0.25 €	0.35 €	0.20 €	0.20 €	0.30 €	0.30 €
click probability (α)	0.030	0.035	0.030	0.035	0.030	0.035	0.020	0.030	0.020	0.030
type probability (ω)	0.30	0.20	0.20	0.30	0.50	0.50	0.25	0.25	0.25	0.25

Table 2. Bayesian prior over the sponsored link list returned by the second domain-specific search engine.

We solved our automated mechanism design problem with Bayes-Nash implementation with the maximization of the integrator's expected utility as objective function. We report the results only for a small number of type profiles. Exactly, we consider the type profiles reported in Tab. 3. The results are reported in Tab. 4, where f_{θ,r_1} denotes the sponsored link displayed in the first position and the search engine it belongs to, f_{θ,r_2} is the same for the second position, p_{θ,r_1} denotes the payment of the search engine whose sponsored link is in the first position once the user clicked on the link, p_{θ,r_2} is the same for the second position. All the other payments are equal to zero. It can be easily observed that the advertiser taxi_service is the one that gives both search engines, if singularly considered, the smallest expected utility. Instead, considering the search engines together, taxi_service is displayed in the second position.

	search engine 1						search engine 2					
	taxi_service		restaurant1		restaurant2		restaurant1		tourist_office		taxi_service	
θ	b	α	b	α	b	α	b	α	b	α	b	α
type profile 1	0.40 €	0.02	0.70 €	0.05	0.60 €	0.035	0.50 €	0.03	0.35 €	0.035	0.20 €	0.03
type profile 2	0.40 €	0.02	0.65 €	0.04	0.60 €	0.035	0.50 €	0.03	0.35 €	0.035	0.30 €	0.02
type profile 3	0.50 €	0.02	0.70 €	0.04	0.70 €	0.035	0.60 €	0.035	0.25 €	0.030	0.20 €	0.03

Table 3. Some type profiles.

θ	f_{θ,r_1}	f_{θ,r_2}	p_{θ,r_1}	p_{θ,r_2}
type profile 1	restaurant1, search engine 1	taxi_service, search engine 1	0.70 €	0.40 €
type profile 2	restaurant1, search engine 1	taxi_service, search engine 1	0.65 €	0.30 €
type profile 3	restaurant1, search engine 1	taxi_service, search engine 1	0.70 €	0.50 €

Table 4. Social choice function and payments.

5 Conclusions and Future Works

The recent advancements in search computing techniques lead to new search paradigms according to which multiple domain-specific search engines are integrated by a special search engine (called integrator). A user can enter a multi-domain query, this query is decomposed by the integrator in a set of single-domain query, each one of them is addressed to a specific-domain search engine. The integrator merges the search results received from each specific-domain search engine. This paradigm allows one to discover a large number of information and to produce very precise search results with respect to the currently available general purpose search engines. In this paper we made a first attempt towards the design of an advertising auction mechanism for this scenario. More precisely, we proposed a business model in which the domain-specific search engines returns, in addition to the search results, a list of sponsored links to the integrator and the integrator merges these lists in a unique list. In order to produce an effective merging, the integrator must be informed about the click probabilities and bids of the advertisers appearing in the lists. We resort to the automated mechanism design framework to design an economic mechanism for the scenario we study. We discuss its desired properties and we report some examples.

The automated mechanism design approach can be used for small problems, but it does not scale for large real-world problem. This pushes for the development of analytical mechanisms or of approximate algorithms. Furthermore, in this paper we have not posed any cooperative constraint over the revenue sharing. In future, our intention is to explore, on the one side, group strategy-proof mechanisms, such as the Moulin's mechanism and its extensions, and, on the other side, two-stage mechanisms to address interdependence valuations.

6 Acknoledgments

This research is part of the "Search Computing" (SeCo) project, funded by the European Research Council (ERC), under the 2008 Call for "IDEAS Advanced Grants", dedicated to frontier research.

References

1. Sakurai, Y., Iwasaki, A., Yokoo, M.: Keyword auction protocol for dynamically adjusting the number of advertisements. In: IAT. (2008) 410–416
2. Narahari, Y., Garg, D., Narayanam, R., Prakash, H.: Game Theoretic Problems in Network Economics and Mechanism Design Solutions. Springer, Berlin, Germany (2009)
3. Google AdWords. (http://adwords.google.com/)
4. Edelman, B., Ostrovsky, M., Schwarz, M.: Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. American Economic Review **97** (2007) 242–259
5. Baeza-Yates, R.A.: Applications of web query mining. In: ECIR. (2005) 7–22
6. White, R.W., Roth, R.A.: Exploratory Search: Beyond the Query-Response Paradigm. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers (2009)
7. SECO project. (http://www.search-computing.it/)
8. Google Fusion Tables. (http://tables.googlelabs.com/)
9. Google Squared. (http://www.google.com/squared/)
10. Google AdSense. (http://www.google.com/adsense/)
11. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations. Cambridge University Press, Cambridge, USA (2008)
12. Sandholm, T.: Automated mechanism design: A new application area for search algorithms. In: CP, Kinsale, County, Springer (2003)
13. Braga, D., Ceri, S., Daniel, F., Martinenghi, D.: Mashing up search services. IEEE Internet Computing **12** (2008) 16–23
14. Bozzon, A., Brambilla, M., Ceri, S., Fraternali, P.: Liquid query: Multi-domain exploratory search on the web. In: WWW. (2010)
15. Dash, R.K., Rogers, A., Jennings, N.R.: A mechanism for multiple goods and interdependent valuations. In: AMEC. (2004) 748–755
16. Mezzetti, C.: Mechanism design with interdependent valuations: Surplus extraction. Discussion Papers in Economics 05/1, Department of Economics, University of Leicester (2005)
17. Krishna, V.: Auction Theory. Academic Press (2006)

A Comparison of Different Automated Market-Maker Strategies

Janyl Jumadinova, Prithviraj Dasgupta

Computer Science Department
University of Nebraska, Omaha, NE 68182, USA
E-mail: {jjumadinova, pdasgupta}@mail.unomaha.edu

Abstract. Financial markets such as stock exchanges and electronic prediction markets frequently use the services of an entity called the market-maker to ensure that the market's traders can make their transactions. Recently, several strategies that can be used by market-makers to control market trading prices have been proposed by various researchers. A detailed comparison of these market maker strategies using real trading data extracted from financial markets is essential to understanding the relative merits and requirements of the different market-maker strategies. We address this aspect of market-maker strategies by empirically comparing different strategies with data obtained from the NASDAQ market. Our results show that a reinforcement learning-based strategy performs well in maintaining low spread as well as in obtaining high utilities, whereas other strategies only succeed in either maintaining low spread or outperforming others in utilities. [1]

Keywords: Market-maker strategy, electronic financial markets, market-maker simulation.

1 Introduction

Over the past few years, the rapid growth and success of automated techniques for e-commerce have resulted in their wide adoption in various domains beyond traditional B2B and B2C commodity markets. For example, in financial markets human traders are being replaced by automated agents that efficiently buy and sell financial securities. Currently many modern exchanges, such as NYSE, NASDAQ, and Toronto Stock Exchange as well as electronic prediction markets, such as tradesports.com, use such automated agents called *market-makers* to regulate prices and quantities of securities or stocks traded by the market's participants.

A market-maker holds a certain number of securities in its inventory with the purpose of being able to sell them to an interested buyer, or to buy securities from a seller selling securities in the market. A market can have either a single market-maker or multiple market-makers that compete with each other. The

[1] This research has been supported by DoD-ONR grant number N000140911174, 2009-2012.

fundamental role of a market-maker is to bring buyers and sellers together so that trading can occur in an efficient and fair manner. The main advantage of automated market-makers is its ability to maintain the liquidity [2] in the market [6]. The liquidity in turn reduces the trading cost of the market's participants [1]. Additionally, market-makers can help to smooth price fluctuations due to spurious supplies or demands [15]. Appropriately designed automated market-makers do not have an incentive to engage in the market manipulation [16].

As the role of the market-makers grows, the need for better understanding of the impact of the market-makers in the market increases as well. In this paper we use a model of a financial market with multiple market-makers to study the potential impact of widespread automated market-maker usage on market dynamics. We investigate four different strategies for automated market-making in financial markets - a myopically optimizing strategy, a reinforcement learning strategy, a market scoring rule and a utility maximizing strategy - with the goal of testing the existing strategies against each other and examining their strengths and weaknesses. Our experiments reveal that the myopically optimizing and market scoring rule-based strategies perform well in maintaining low spread and smooth market price, however they fall short in maximizing utilities as compared to other strategies. On the other hand, the utility maximizing strategy with different risk attributes performs very well in obtaining high utilities, although it fails in maintaining low and consistent spread. Consequently, the market price tends to fluctuate significantly. Finally, the reinforcement learning strategy fulfills its tasks of both controlling the spread and maximizing utility.

2 Related Work

The automation of a market-makers's functions was suggested more than three decades ago [2]. Previously, several theoretical approaches, albeit with certain simplifying assumptions, have been proposed to understand the effects of market-makers on financial markets. Garman [7] describes a model with a single, monopolistic market-maker, who sets prices, receives orders and clears trades and tries to maximize expected profit per unit time. Such market-maker fails when it runs out of inventory or cash. In [14], the authors study the optimal behavior of a single market-maker who gets a stochastic demand and tries to maximize its expected utility of final wealth, which depends on the profit it receives from trading. Glosten and Milgrom [10] investigate the market-making model with asymmetric information. Das [6] empirically studies different market-making strategies and concludes that a heuristic strategy that adds a random value to zero-profit market-makers improves the profits in the markets. Gu [11] explores changing the market-maker behavior by estimating the market-maker's profitability under different parameters. The results show that a profit-maximizing market-maker's objectives may not align with price variance minimization, which can be one of the qualities of an orderly market. Westerhoff [17] also explores the

[2] A market is said to be liquid if traders can buy or sell large quantities of a security without causing large changes in the market price. Liquidity is a valuable characteristic of a market because it enables high volume trades.

impact of inventory restrictions in a setup with an implied market-maker. The market-maker price adjustment reactions differ depending on the current inventory position along with current excess demands. The market-maker is assumed to make greater price adjustments when these two variables are of the same sign. Market-making has also been adopted as a test-bed for machine learning techniques [15] with a goal to demonstrate the general effectiveness of a learning algorithm.Also, empirical work has demonstrated the limitations of hard-coding market-making rules into an algorithm [12]. In [16], the primary goal is to optimally change the spread over the next iteration instead of finding the best model for past transactions.

In the past, several market-maker strategies have been proposed and there have been a few studies on the market-maker's effect on the market. Most of these past studies focus on a market with a single market-maker or a market with multiple market-makers of the same strategy. However, there does not exist a study comparing the effect of different market-maker strategies in a market with multiple market-makers. In this paper, we attempt to address this deficit by providing experimental results using data from real security markets to examine the behavior of a market with multiple market-makers that employ different competing strategies. We also analyze the effect of each market-making strategy and the combinations of strategies on the market price dynamics.

3 Model

We have adapted the well-known Glosten and Milgrom [6, 10] model of financial markets to a multi-agent framework of a financial market with multiple electronic market-makers. In our model, each human trader is modeled as a software agent called a trading agent that embodies the behavior of a human trader.

The market consists of N traders and M market-makers who buy and sell securities or stocks, where $N >> M$. Each *trading episode* e consists of T *trading periods*. Each stock s, has a true or fundamental value $V_{s,e}$ at trading episode e. That is, there is some exogenous process that determines the value of the stock. The true price of a stock is different from the market price at which the stock gets traded. The market price of a stock is determined by the interaction between the market-makers and the traders. The stock's true price $V_{s,e}$ gets updated during each trading episode with some probability $\pi_{s,e+1}$ according to the following equation [5, **?**]:

$$V_{s,e+1} = V_{s,e} + jump, \tag{1}$$

where $jump$ is a parameter sampled from a normal distribution with mean $\mu_{V s,e}$ and variance $\delta_{V_{s,e}}$. The jump in the true value of the stock can be positive or negative and usually corresponds to some new information about the stock arriving in the market from external sources. The volatility of the stock value is influenced by the value of the standard deviation of the jump and the probability that the jump will occur.

Market-maker m's bid(buy) price of the stock s at trading period t, $p^{sell}_{m,s,t}$, is the price the market-maker charges traders for buying 1 share of stock s.

The market sell price of the stock s at trading period t, $P_{s,t}^{sell}$, is the price that the market-maker pays to traders for buying 1 share of the stock s. The market bid(buy) price $P_{s,t}^{buy}$ at trading period t for stock s is the maximum of the market-makers' bid prices. The market ask(sell) price $P_{s,t}^{sell}$ at trading period t for stock s is the minimum of the market-makers' ask prices.

The different parameters used in our financial market model to define the market characteristics and specify the market-makers and trading agents behavior are shown in Table 1 and described below.

Market	**Parameters**
e	Trading episode
t	Trading period
N	Number of traders in the market
M	Number of market-makers in the market
S	Number of stocks
$V_{s,e}$	True value of stock s during trading episode e
$\pi s, e$	Probability that the jump in the true value of stock s occurs during trading episode e
$P_{s,t}^{sell}$	Market sell price of the stock s at trading period t
$P_{s,t}^{buy}$	Market buy price of the stock s at trading period t
Market-Maker Agent	**Parameters**
$p_{m,s,t}^{sell}$	Market-maker m's sell price of the stock s at trading period t
$p_{m,s,t}^{buy}$	Market-maker m's buy price of the stock s at trading period t
θ_m	Risk coefficient of the market-maker agent m
u_m	Market-maker m's utility

Table 1. Parameters used in our model.

3.1 Trader Behavior

When a trader enters the market, it is randomly assigned to some market-maker. At each trading period t, traders place a buy or sell order, or no order at all, based on the buy or sell price of the stock given by the market-maker. Each trader n has a valuation for each stock s, $W_{n,s}$ sampled from a normal distribution. If $W_{n,s} > P_{s,t}^{sell}$, the trader buys one unit of the stock s, if $W_{n,s} < P_{s,t}^{buy}$, the trader sells one unit of the stock s, and if $P_{s,t}^{buy} \leq W_{n,s} \leq P_{s,t}^{sell}$, the trader holds the stock.

3.2 Market-maker Behavior

At each trading period t, each market-maker sets the bid and ask prices for each stock according to some algorithm. The difference between the bid and ask prices is called the stock's *spread*. Market-makers use this spread of a stock to ameliorate their risks of holding a considerable quantity of the stock. Market-makers execute the buy or sell stock orders from the traders immediately. A market-maker does not know the true value of a stock, but it receives a noisy signal about the jump in the true value of the stock, $jump + \tilde{N}(0, \delta_m)$, where $jump$ is the actual jump that has occurred and $\tilde{N}(0, \delta_m)$ represents a sample from a normal distribution with mean 0 and variance δ_m^2. Figure 1 shows the operations of the market-maker in a market.

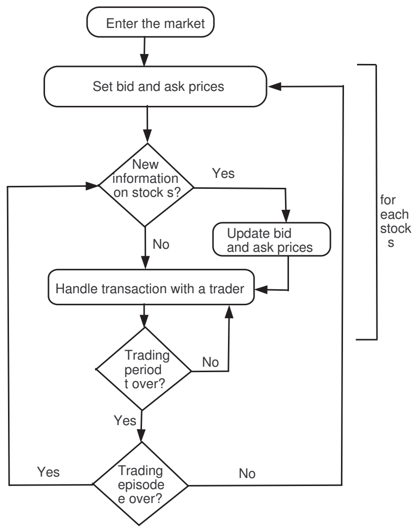


Fig. 1. A flowchart showing the operation of the market-maker agents in the market.

Before we present our experimental results, we first briefly review the market-maker algorithms that we use for our comparisons.

A Myopically Optimizing Market-Maker A Myopically Optimizing Market-Maker uses an algorithm developed by Das in [6]. The key aspect of the algorithm is that the market-maker uses the information conveyed in trades to update its beliefs about the true value of the stock, and then it sets buy/ask prices based on these beliefs. The market-maker maintains a probability density estimate over the true price of the stock. There are two key steps involved in the market-making algorithm. The first is the computation of bid and ask prices given a probability density estimate over the true price of the stock, and the second is the updating of the density estimate given the information implied in trades. This market-maker optimizes myopically, setting the prices that give the highest expected profit at each trading period. That is, the optimal buy price is the price that maximizes the expression $E(profit_s | p^{buy}_{m,s,t} = x)$, and the optimal sell price is the price that maximizes the expression $E(profit_b | p^{sell}_{m,s,t} = x)$, where

$profit_s(profit_b)$ is the profit from a marginal sell(buy) order being received. The market-maker uses the Bayesian updating method described in [5] to update its density estimates. All of the points in the density estimate are updated based on whether a buy order, sell order, or no order was received. The density estimate is initialized to be normal.

Reinforcement Learning Market-Maker Chan and Shelton [3] have modeled the market-making problem in the framework of reinforcement learning. They have used Markov decision process (MDP) to model reinforcement learning of a market-maker. A state is defined as $s_t = (inv_{m,t}, imb_t, qlt_t)$, where $inv_{m,t}$ is the market-maker m's inventory level, imb_t is the order imbalance, and qlt_t is the market quality at trading period t. The inventory level is the market-maker's current holding of the stock. The order imbalance is calculated as the sum of the buy order sizes minus the sum of the sell order sizes during a certain period of time t. Market quality measures are the bid-ask spread and price continuity (the amount of price change in subsequent trades). Given the states of the market, the market-maker reacts by adjusting the bid/ask prices and trading with incoming orders. The action vector for market-maker m is defined as $a_{m,t} = (p^{buy}_{m,t}, p^{sell}_{m,t})$.

The market-makers can obtain the optimal strategy by maximizing the profit, by minimizing the inventory risk, or by maximizing market qualities. Thus, the reward at each time step depends on the profit received, the change of inventory, and the market quality measures. This strategy assumes the risk-neutrality of the market-maker.

Utility Maximizing Market-Maker with Risk attributes Previous research [9, 19] has shown that by considering risk-taking and risk-averse behaviors of the human traders, the behavior of the market can be improved. We set out to see if the incorporating the risk behavior of the market-makers can improve the financial market performance. Following [9], we adopt a constant relative risk averse (CRRA) utility function $\widetilde{u}_{m,t}$ for market-maker m with a relative risk aversion coefficient. CRRA utility functions have been widely used to model risk behaviors. Relative risk aversion coefficient, θ_m, is used to classify market-maker m's risk levels as follows. If $\theta_m > 0$, the market-maker m is risk-averse, if $\theta_m = 0$, the market-maker m is risk-neutral, and if $\theta_m < 0$, the agent m is risk-seeking. Unless otherwise specified, the market-makers' risk coefficients are normally distributed in our simulations. Following the trading agent utility model in [9], during each trading period t market-maker m uses its instantaneous utility $\acute{u}_{m,t}$ and its risk-taking coefficient to calculate its modified instantaneous utility for that trading period, using Equation 2.

$$\widetilde{u}_{m,t}(\acute{u}_{m,t}, \theta_m) = \begin{cases} \frac{\acute{u}_{m,t}^{1-\theta_m}}{1-\theta_m} & \text{, if } \theta_m \neq 1; \\ ln(\acute{u}_{m,t}) & \text{, if } \theta_m = 1. \end{cases} \tag{2}$$

These market-maker agents are utility maximizers, that is they update prices so that their overall utility is maximized.

LMSR Market-Maker Hanson invented a market-maker for the use in prediction market applications called the logarithmic market scoring rule (LMSR) market-maker [13]. We have used Chen and Pennock's formulation of Hanson's (LMSR) market-maker [4]. Let $\overline{q} = (q_1, q_2 ... q_N)$ be the vector specifying quantities of stocks held by the different trading agents in the market. The total cost incurred by the trading agents for purchasing these stocks is calculated by the market-maker using a cost function $C(\overline{q}) = b \cdot ln(\sum_{j=0}^{|\overline{q}|} = e^{q_j/b})$. The parameter b is determined by the market-maker and it controls the maximum possible amount of money the market-maker can lose as well as the quantity of shares that agents can buy at or near the current price without causing massive price swings. If an agent purchases a quantity δ_q of the security, the market-maker determines the payment the agent has to make as $p^{buy}_{s,m,t} = C(\overline{q} + \delta_q) - C(\overline{q})$. Correspondingly, if the agent sells δ_q quantity of the security, it receives a payment of $p^{sell}_{s,m,t} = C(\overline{q}) - C(\overline{q - \delta_q})$ from the market-maker.

4 Experimental Results

We have compared the four market-maker algorithms described in the previous section through several simulations. The true value for stock s during episode s was obtained from the data of real NASDAQ stock markets. First ten stocks were randomly selected from all the stocks traded on NASDAQ. Then the real data of those ten stocks was downloaded from Yahoo! Finance [20]. We have used open prices of each day to simulate the true value of the stock at the beginning of each trading episode.

Each trading episode consists of 100 trading periods, where each trading period lasts for $0.5sec$. We simulate the financial market with 100 traders and 3 or 2 market-makers. We show the results of our simulations over 100 trading episodes.

First we want to observe the behavior of the market with market-makers that use the same strategy. After that we perform the pairwise comparison of different market-maker strategies and evaluate the performance of each one in more detail. We report the market price, the spread, and the utility earned by the each type of the market-maker used in our simulations. The market price and the spread evaluate the quality of the market, whereas the utility evaluates the profitability of the strategy employed by the market-maker. In our graphs we show the results for the Yahoo! stock. Figure 2 shows the Yahoo! stock's price obtained from [20]. In our first set of experiments there are 3 market-makers that use the same strategy in the market. Figure 3 shows the simulations of the market with myopically optimizing market-makers. We can see from the spread graph that the myopically optimizing market-maker is sensitive to the price variations in the market. The spread value has large fluctuations following the jump in the true value of the stock. Spread seems to stabilize somewhat until the next jump. Due to large jumps in the spread value, myopically optimizing market-makers are able to keep increasing their utilities. Myopically optimizing market-makers are able to avoid causing big jumps in the market price, which is one of the important functions of the market-makers.

Figure 4 shows the simulations of the market with reinforcement learning market-makers. The utility of the reinforcement learning market-makers is ex-

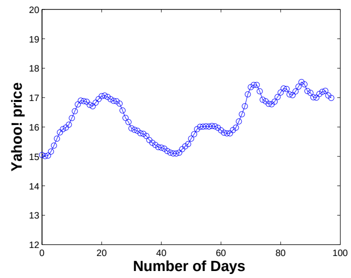


Fig. 2. Yahoo! price data used in our simulations.

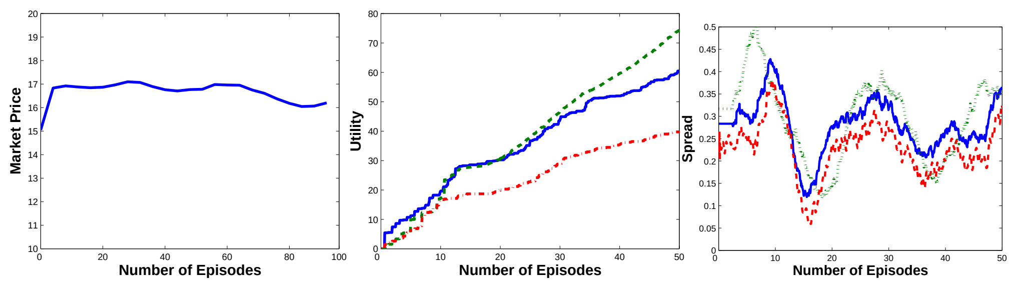


Fig. 3. Myopically Optimizing Market-Makers.

pected to improve with each trading episode. As expected, these market-makers perform very well with respect to utility-maximization. However, the spread value fluctuates somewhat throughout the trading episodes. The market price does not fluctuate a lot throughout the simulation.

Figure 5 shows the simulations of the market with logarithmic market scoring rule (LMSR) market-makers. LMSR market-makers perform very well the function of maintaining an orderly market. That is, the market price is smooth and the spread is steady and consistent. LMSR market-makers do not aggressively maximize their utility, as can be seen from the utility graph in Figure 5.

Figure 6 shows the simulations of the market with utility maximizing market-makers with different risk attributes, i.e. with one risk-taking, risk-neutral, and risk-averse market-maker. We can see that the risk-taking market-maker is able to obtain slightly higher utility than the risk-neutral and risk-averse market-makers. Risk-averse market-maker gets the least utility, but maintains the smallest spread. Risk-taking market-maker does not control the spread value well, as it fluctuates a lot and by large amounts. Also, the market price has more fluctuations with these market-makers than with other types of market-makers.

For our next set of simulations we perform pairwise comparisons of different market-maker strategies. We simulate the market with 2 market-makers, one of

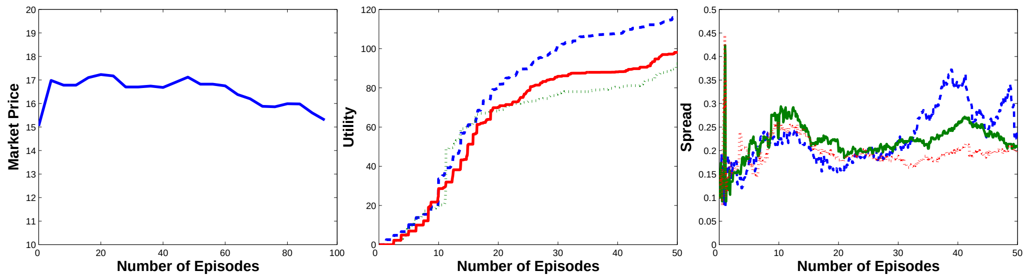


Fig. 4. Reinforcement Learning Market-Makers.

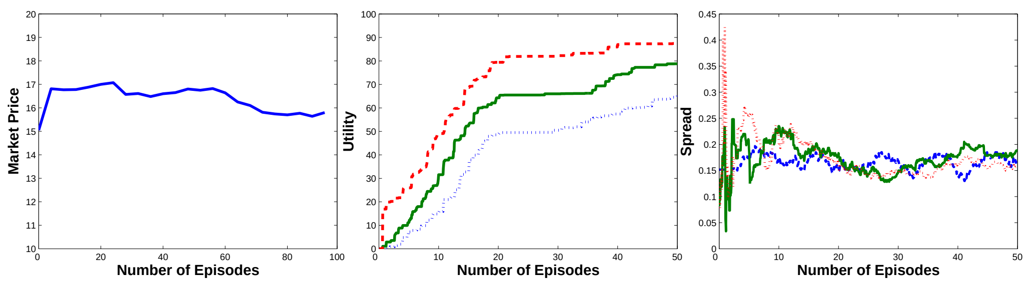


Fig. 5. LMSR Market-Makers.

each type. However, when comparing utility maximizing market-makers with 3 different risk attributes, we use 4 market-makers in the market.

First we compare myopically optimizing market-maker with 3 utility maximizing market-makers, one risk-taking, one risk-neutral, and one risk-averse market-maker. As can be seen from Figure 7, the fluctuations in the market price are pretty significant. We foresee that this is mainly due to presence of utility maximizing market-makers, since their primary function is not the control of the quality of the market, but utility maximization. Although, it is interesting to see that the risk-averse utility maximizing market-maker is able to maintain steady and low spread, and is very compatible in that regard with the myopically optimizing market-maker. Myopically optimizing market-maker also outperforms the risk-averse market-maker in overall utility.

Figure 8 illustrates the market with one myopically optimizing market-maker and one LMSR market-maker. We can see that these market-makers contribute to maintaining smooth market price and close spread values. However, myopically optimizing market-maker outperforms the LMSR market-maker by 40% on average in utility.

In Figure 9 we present the comparison of the myopically optimizing market-maker with reinforcement learning market-maker. Our results show that reinforcement learning market-maker is able to obtain 24% higher utility on average than the myopically optimizing market-maker. However, myopically opti-

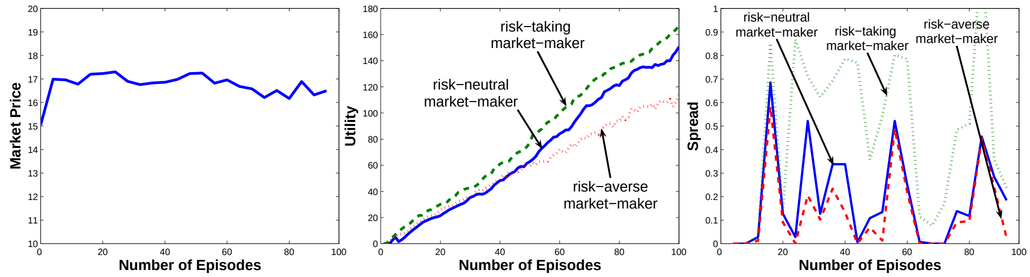


Fig. 6. Utility Maximizing Market-Makers with different risk attributes.

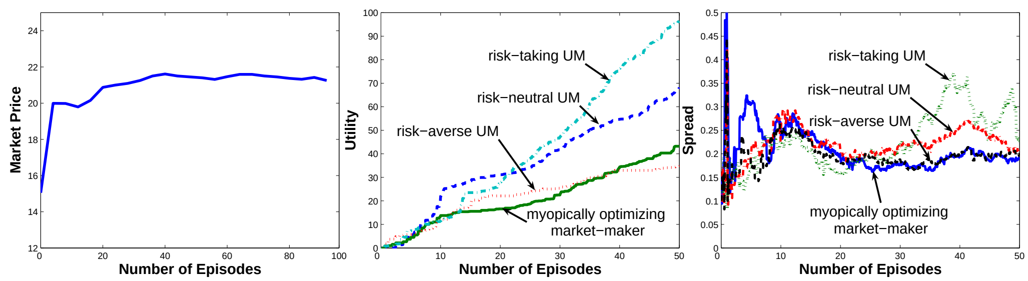


Fig. 7. Myopically Optimizing Market-Maker versus Utility Maximizing Market-Makers with different risk attributes.

mizing market-maker maintain 6.5% less spread on average than the reinforcement learning market-maker. Both market-makers do a good job in maintaining smooth market price and steady spread.

Next we compare the performance of the LMSR market-maker with 3 utility maximizing market-makers, i.e. risk-taking, risk-neutral, and risk-averse market-maker. We can see from Figure 10 that the volatility in the market is significant, with the fluctuations in the market price and large variations in the spread values. All utility maximizing market-makers outperform LMSR market-maker in utility. For example, risk-taking utility maximizing market-maker obtains 49% higher utility than LMSR market-maker. However, LMSR market-maker has 31% lower average spread than the risk-taking market-maker, which has the highest spread.

Figure 11 shows the performance of the reinforcement learning market-maker against the utility maximizing market-maker with different risk attributes. Our results indicate that the reinforcement learning market-maker has lower spread. In particular its average spread is 25%, 22%, and 13% lower than the risk-taking, risk-neutral, and risk-averse utility maximizing market-makers. Also, reinforcement learning market-maker is able to outperform risk-averse market-maker in utility by 11%, but it receives 69% less utility than risk-neutral market-maker, and over 100% less utility than the risk-taking market-maker.

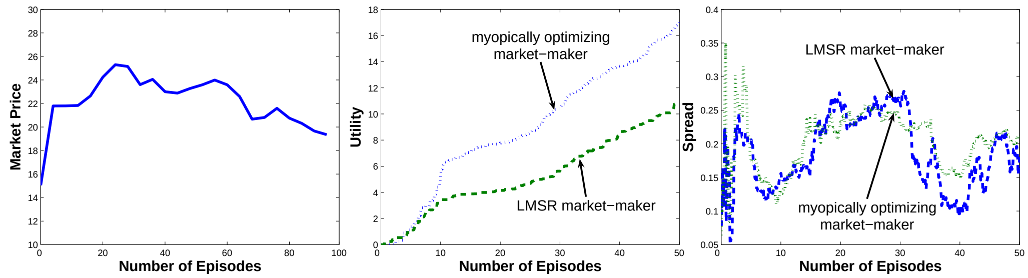


Fig. 8. Myopically Optimizing Market-Maker versus LMSR Market-Maker.

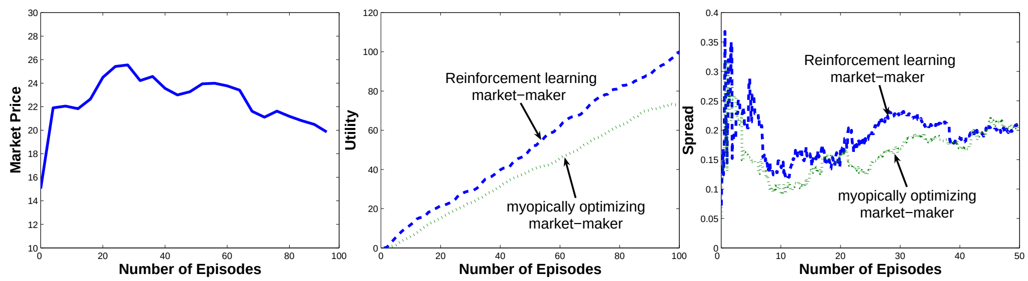


Fig. 9. Myopically Optimizing Market-Maker versus Reinforcement Learning Market-Maker.

Reinforcement learning market-maker performance comparison with LMSR market-maker is shown in Figure 12. The market price is smooth throughout 100 trading episodes. Although reinforcement learning market-maker obtains 54% more utility than the LMSR market-maker, the spread difference between two market-makers is not very significant (8%).

Finally, we simulate the market with all four market-makers as shown in Figure 13. The market price is smooth throughout 100 trading episodes. We observe that the utility maximizing, risk-neutral market-maker outperforms other market-makers in utility. The difference in the utility between the utility maximizing market-maker and LMSR market-maker (which got the least utility) is on average 60%. However, LMSR market-maker is able to maintain the lowest spread in the market. The difference in spread between the LMSR market-maker and the utility maximizing, risk-neutral market-maker is on average 73%. However, the spread difference between reinforcement and myopically optimizing market-makers is not very significant (11%).

5 Discussions and Lessons Learned

In this paper, we have used an agent-based financial market model to analyze the dynamics in the market with multiple market-makers. We investigated the effects

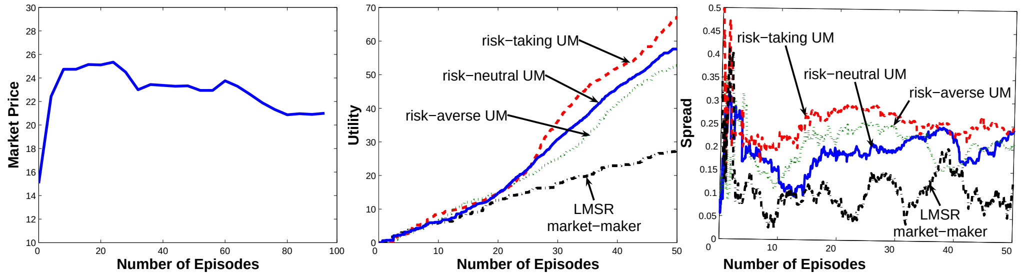


Fig. 10. LMSR Market-Maker versus Utility Maximizing Market-Makers with different risk attributes.

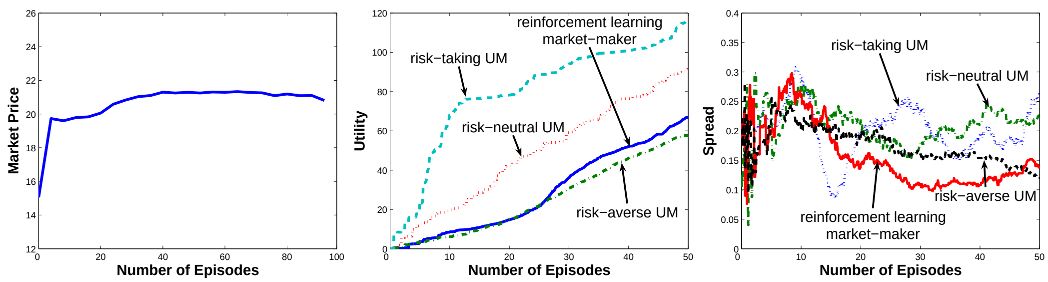


Fig. 11. Reinforcement Learning Market-Maker versus Utility Maximizing Market-Makers with different risk attributes.

of various market-making strategies on the market prices and market-makers' spread and utilities. The difficulty in constructing the market-making strategies comes from the need for the market-maker to balance conflicting objectives of maximizing utility and market quality, that is fine-tuning the tradeoff between utility and market quality.

Our simulation results show that the utility maximizing risk-taking and risk-neutral market-makers outperform all the other types of market-makers in utility, however they lack in maintaining the market quality, i.e. low and continuous spread and smooth market price. Myopically optimizing market-maker performs well with both maintaining good market quality and obtaining high utility. Reinforcement learning market-maker has comparable results when it comes to utility compared to the other market-maker strategies that are designed with a primary goal of maintaining market quality. Reinforcement learning market-makers also do their job of market control very well. LMSR market-maker does not do so good in terms of maximizing its utility, since it is not designed to do that. However, it performs well in maintaining continuously low spread.

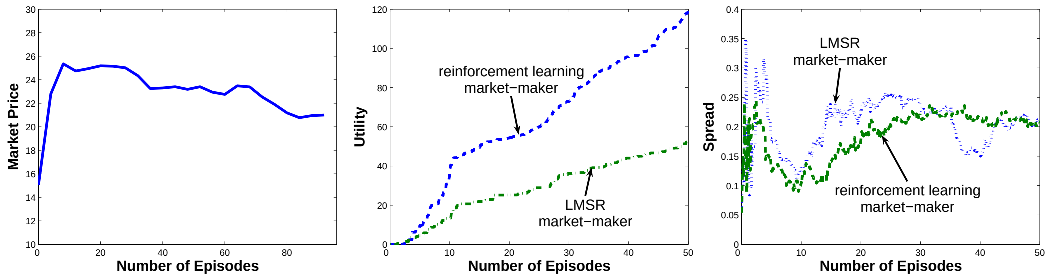


Fig. 12. Reinforcement Learning Market-Maker versus LMSR Market-Maker.

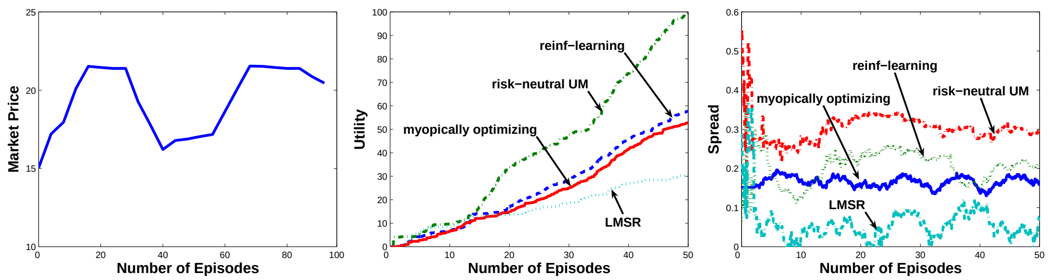


Fig. 13. Market with Reinforcement Learning, Utility Maximizing, Myopically Optimizing and LMSR market-makers.

6 Future work

This is our first step in performing a comparison of multiple market-maker strategies. In the future, we wish to explore different extensions of this work. First of all, we would like to propose and perform comparisons of other market-maker strategies such as using a minimax regret algorithm for price adjustments by the market-maker. Secondly, we would like to study the performance of the market-makers with a more complex behavior, such as dynamically switching strategies based on past performance. This way, a better balance of maintaining market quality and maximizing market-maker utilities may be obtained. And lastly, we would like to add various behavioral attributes to the market-maker model such as different risk attributes and making untruthful price revelations through bluffing for improving profits. Market-makers provide a reliable and economic technology for efficient operation of financial markets and we believe that future investigation of their strategies along the directions we explored in this paper will lead to more efficient market performance.

References

1. Yakov, A., Mendelson, H., "The Liquidity Route to a Lower Cost of Capital," Journal of Applied Corporate Finance, 12, pp. 8-25, 2000.

2. Black, F., “Toward a Fully Automated Exchange,” Financial Analysts Journal, 27, pp. 29-34, 1971.
3. Chan, N. T. and C. Shelton, ”An electronic market-maker,” Technical Report AI-MEMO 2001-005, MIT, AI Lab, 2001.
4. Chen, Y., Pennock, D., “Utility Framework for Bounded-Loss Market Maker,” Proc. of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI 2007), pp. 49-56, 2007.
5. Das. S., “A learning market-maker in the Glosten-Milgrom model,” Quantitative Finance, 5(2), pp. 169180, 2005.
6. Das, S., “The Effects of Market-making on Price Dynamics,” *Proceedings of AA-MAS 2008*, pp. 887-894, (2008).
7. Garman, M., “Market microstructure,” Journal of Financial Economics, Volume 3, Issue 3, pp. 257-275, 1976.
8. GETCO URL: http://www.getcollc.com
9. Gjerstad, S., ”Risks, Aversions and Beliefs in Predictions Markets,” *mimeo, U. of Arizona.*, 2005.
10. Glosten, L., and Milgrom, P., “Bid, Ask and Transaction Prices in a Specialist Market with Heterogeneously Informed Traders,” Journal of Financial Economics, Volume 14, Issue 1, pp. 71-100, 1985.
11. Gu, M., “Market mediating behavior: An economic analysis of the security exchange specialists,” Journal of Economic Behavior and Organization 27, 237256, 1995.
12. Hakansson, N.H., Beja, A., and Kale, J., “On the Feasibility of Automated Market Making by a Programmed Specialist,” The Journal of Finance, Vol. 40, No. 1, pp. 1-20, 1985.
13. Hanson, R., “Logarithmic Market scoring rules for Modular Combinatorial Information Aggregation,” Journal of Prediction Markets, Vol. 1, No. 1, pp. 3-15, 2007.
14. Ho, T., Stoll, H. R., “Optimal dealer pricing under transactions and return uncertainty,” Journal of Financial Economics, Vol. 9, No. 1, pp. 47-73, 1981.
15. Kim, A.J., Shelton, C.R., “Modeling Stock Order Flows and Learning Market-Making from Data,” Technical Report CBCL Paper No.217/AI Memo, 2002-009, M.I.T., Cambridge, MA, 2002.
16. Nevmyvaka,Y., Sycara, K., Seppi, D., “Electronic Market Making: Initial Investigation,” in Artificial Intelligence in Economics and Finance , World Scientific , 2005.
17. Westerhoff, F., “Market-maker, inventory control and foreign exchange dynamics,” Quantitative Finance 3, pp. 363-369, 2003.
18. Wolfers, J., Zitzewitz, E., “Prediction Markets,” *Journal of Economic Perspectives*, 18(2), pp. 107-126, 2004.
19. Wolfers, J., Zitzewitz, E., “Interpreting Prediction Markets as Probabilities,” *NBER Working Paper No. 12200.*, 2006.
20. Yahoo! Finance URL: http://www.finance.yahoo.com

On Optimal Agendas for Multi-Issue Negotiation

Shaheen Fatima1 Michael Wooldridge2 Nicholas R. Jennings3

1 Department of Computer Science
Loughborough University, Loghborough, LE11 3TU, U.K.
S.S.Fatima@lboro.ac.uk

2 Department of Computer Science
University of Liverpool, Liverpool, L69 7ZF, U.K.
mjw@liv.ac.uk

3 School of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, U.K.
nrj@ecs.soton.ac.uk

Abstract. This paper analyzes bilateral multi-issue negotiation where the issues are *divisible*, there are time constraints in the form of *deadlines* and *discount factors*, and the agents have different preferences over the issues. The issues are negotiated using the *package deal* procedure. The set of issues to be negotiated is a choice variable in that the agents can decide what issues to negotiate. This set is called the negotiation *agenda*. Since the outcome of negotiation depends on the agenda, it is important to determine what agenda maximizes an agent's utility and is therefore its *optimal agenda*. To this end, this paper presents polynomial time methods for finding an agent's optimal agenda.

1 Introduction

Negotiation has long been studied by economists and game theorists but is now receiving increasing attention from researchers in multi-agent systems [10, 13]. In the existing work, the analysis of negotiation typically begins with a given set of issues and the parties' preferences (in the form of their utilities) for different possible settlements of the issues. Within this framework, theorists have investigated a range of negotiation procedures such as the *package deal* procedure (PDP), the *simultaneous* procedure (SP), and the *sequential* procedure (SQP) [7, 8, 2].

It is well known that different procedures result in different outcomes and, therefore, give different utilities to the agents [6]. So it is important that agents choose the right procedure. Moreover, for a given procedure, it is also important that the agents choose an appropriate *agenda*. The term agenda refers to the set of issues included for negotiation [11]. The agenda is important because, irrespective of the procedure, the outcome of negotiation depends on the agenda [7, 14]. Thus, given the utility maximizing feature of agents, it is important to find what agenda maximizes their utility and is therefore *optimal* for them. For example, consider a car dealer who has five second hand cars to sell. A potential buyer may be interested in buying two of these. So he must first choose which two cars to negotiate the price for (i.e., from all possible subsets of size two, he must choose the one that maximizes his utility and so is his *optimal agenda*). Note that here, the buyer has choice over the agenda but not the seller.

Although the importance of agendas has been recognized [7, 11], most existing work has taken the agenda as given and then analyzed the outcome for different procedures [8, 5, 6]. But as the above example illustrates, the set of issues to be negotiated themselves are often choice variables (i.e., what issues to negotiate can be chosen by a negotiator) whose ultimate configuration can have decisive effects on the negotiation outcome [14]. Thus, in addition to knowing what procedure is best for an agent, it is important to know what agenda maximizes an agent's utility and is therefore its optimal agenda. To this end, this paper analyzes the problem of finding optimal agendas in the context of the PDP (future work will deal with SP and SQP). The key contribution of this paper lies in presenting, for the first time, polynomial time methods for determining optimal agendas for the PDP.

The rest of the paper is structured as follows. Section 2 provides a discussion of related literature. Section 3 describes the setting and defines 'agenda' and 'optimal agenda'. Section 4 shows how to find optimal agendas for the complete information setting. Section 5 builds on Section 4 to show how to find optimal agendas in an incomplete information setting. Section 6 concludes.

2 Related Literature

There are different procedures such as the PDP, the SP, and the SQP for multi-issue negotiation, and the term agenda has different meanings in these different contexts. For the PDP (which is the focus of this paper) and the SP, the term *agenda* refers to the set of issues to be included for negotiation. But for the SQP, the term refers to not just the set of issues to be included for negotiation but also the order in which they will be negotiated.

Although the importance of agendas has been recognized [7, 11], most existing work has taken the set of issues as given and then analyzed the equilibrium outcome for different procedures [9, 8, 6]. In the context of the SQP, [1] takes the set of issues to be negotiated as given and provides an analysis of the role of information and time preferences on the equilibrium, while [4, 5] has dealt with taking the set of issues to be negotiated as given and finding an optimal ordering for the given set. But as the example outlined in the Introduction illustrates, the set of issues to be negotiated themselves are often choice variables (i.e., what issues to negotiate can be chosen by a negotiator) whose ultimate configuration can have decisive effects on the negotiation outcome [14]. Thus, a negotiator must make three key choices: he/she must decide the following:

1. what negotiation procedure to use,
2. what set of issues to negotiate, and
3. for the SQP, what ordering to use for negotiating a given set of issues.

To our knowledge, there is no existing work that deals with finding what set of issues to negotiate. Hence, the novelty of this paper lies in showing how to find the set of issues to negotiate. Specifically, it analyzes the problem of finding optimal agendas in the context of the PDP. The key contribution of this paper lies in presenting, for the first time, polynomial time methods for determining optimal agendas for the PDP.

3 The Negotiation Setting

An agent's optimal agenda depends on its equilibrium utility from all possible agendas. So we first give an overview of the equilibrium for single and multi-issue negotiation. On the basis of this equilibrium, we will show how to find optimal agendas in Sections 4 and 5. A formal definition of optimal agenda is given toward the end of this section.

3.1 Equilibrium for Single Issue

We use the single issue model of [6] in which two agents, a and b, negotiate over a single divisible issue which is a 'pie' of size 1. The agents want to determine how to split it between themselves. Let $n \in N^+$ be the deadline and $0 < \delta \leq 1$ the discount factor for both agents. The agents use an alternating offers protocol [12], which proceeds through a series of time periods. One of the agents, say a, starts in the first time period (i.e., $t = 1$) by making an offer ($0 \leq x^a \leq 1$) to b. Agent b can accept/reject the offer. If it accepts, negotiation ends in an agreement with a getting x^a and b getting $x^b = 1 - x^a$. Otherwise, negotiation goes to the next time period, in which b makes a counter-offer. This process continues until one of the agents either accepts an offer or quits negotiation (resulting in a conflict).

Agent a's utility at time t from a share x^a is $u^a(x^a, t) = x^a \delta^{t-1}$ if $t \leq n$, otherwise it is zero. For b, u^b is analogous. For this setting, the equilibrium offers are obtained using backward induction (BI) as follows. Let $n = 1$ and let a be the first mover. If b accepts a's proposal at $t = 1$, the division occurs as agreed; if not, neither agent gets anything (since the deadline is $n = 1$). Here, a is in a powerful position and is able to keep 100% of the pie and give nothing to b[1]. Agent b accepts and agreement takes place at $t = 1$.

Now, let $n = 2$ and $\delta = 1/2$. The first mover (say a) decides what to offer at $t = 1$, by looking ahead to $t = 2$ and reasoning backwards. Agent a reasons that if negotiation goes to $t = 2$, b will take 100% of the shrunken pie by offering $[0, 1/2]$. So, at $t = 1$, if a offers b anything less than 1/2, b will reject the offer. So, at $t = 1$, a offers $[1/2, 1/2]$. Agent b accepts and an agreement occurs at $t = 1$. In this way, BI was used to obtain the equilibrium offer for $t > 2$.

3.2 Equilibrium for Multiple Issues for the PDP

Let $I = \{1, 2, \ldots, m\}$ be the set of issues. Agent a's (b's) preference for issue i is represented with a weight $w_i^a \in R_+$ ($w_i^b \in R_+$). Each issue is a pie of size 1 and can be split between the agents such that if x_i^a and x_i^b are a's and b's shares for issue i, then $x_i^a + x_i^b = 1$. Here n is the deadline for all the issues, and δ the discount factor for all of them. Agent a's cumulative utility at time $t \leq n$ is given by $U^a(I, x^a, t) = \delta^{t-1} \sum_{i=1}^{m} w_i^a x_i^a$. For b, $U^b(I, x^b, t)$ is analogous. As before, an agent's utility for $t > n$ is zero.

[1] It is possible that b may reject such a proposal. But, irrespective of whether b accepts or rejects, it gets zero utility (since the deadline is $n = 1$). So, b accepts a's offer.

For the above model, the equilibrium was obtained, using BI, as follows [6]. We give an overview of this first for complete information setting, and then explain its extension to an incomplete information setting.

The complete information setting. Let $\text{SA}(t)$ ($\text{SB}(t)$) denote a's (b's) equilibrium strategy for time t. For $t = n$, the offering agent gets a 100% of all the shrunken pies. For all previous time periods, $t < n$, the offering agent (say a) proposes a package ($[x^a, x^b]$) such that b's cumulative utility from it is what b would get from its own offer for $t + 1$. If there is more than one such package, then a must choose the one that maximizes its own cumulative utility. Thus, a must solve the following trade-off problem (called TA): maximize $\sum_{i=1}^{m} w_i^a x_i^a$, subject to $\sum_{i=1}^{m} w_i^b(1 - x_i^a) = Y$ where $0 \leq x_i^a \leq 1$. Here, Y is b's cumulative utility from its own offer SB(t+1). On the other hand, if a receives an offer $[x^a, x^b]$ at time t, then it accepts if $U^a(I, x^a, t) = Z$ where Z is a's cumulative utility from its own offer $\text{SA}(t + 1)$. The equilibrium strategy for b (in terms of TB) is analogous. Thus we have:

$$\text{SA}(n) = \begin{cases} \text{OFFER} \ \ [\mathbf{1}, \mathbf{0}] & \text{If } a\text{'s turn to offer} \\ \text{ACCEPT} & \text{If } a\text{'s turn to receive} \end{cases}$$

where **0** (**1**) denotes a vector of m zeros (ones). For all preceding time periods $t < n$, the strategies are as follows:

$$\text{SA}(t) = \begin{cases} \text{OFFER} \ \ \text{TA} & \text{If } a\text{'s turn to offer} \\ \text{If } \ (U^a(I, x^a, t) \geq Z) \ \text{ACCEPT} & \text{If } a \text{ receives } x^a \\ \text{else REJECT} & \end{cases}$$

For b, SB(t) is analogous. Both TA and TB are standard *fractional knapsack* problems [3]. The solution to TA is for a to consider issues in ascending order of w_i^a/w_i^b and allocate to b maximum possible share for the individual issues until b's cumulative utility equals Y. If the issues $1, \ldots, m$ are in ascending order of w_i^a/w_i^b, the equilibrium solution is $x^a = \{0, .., 0, x_c^a, 1, .., 1\}$, $x^b = \{1, .., 1, 1 - x_c^a, 0, .., 0\}$. The solution to TB is analogous.

An incomplete information setting. Here, the agents are uncertain about their utilities. Let T be the number of possible utility function pairs. The jth possible pair (U_j^a, U_j^b) occurs with probability γ_j. For a, the jth function is: $U_j^a(I, x^a, t) = \delta^{t-1} \sum_{i=1}^{m} w_{ij}^a x_i^a$, and its expected utility is $EU^a(I, x^a, t) = \delta^{t-1} \sum_{j=1}^{T} \gamma_j \times U_j^a(I, x^a, t)$. Agent a's expected weight for issue i is $ew_i^a = \delta^{t-1} \sum_{j=1}^{T} \gamma_j w_{ij}^a$. For b, U_j^b, $EU^b(I, x^b, t)$, and ew_i^b are analogous.

Given this, agent a's tradeoff problem at time t is to find a package $[x^a, x^b]$ that solves the following problem:

$$\begin{array}{lll} \text{TA-I} & \text{maximize} & EU^a(I, x^a, t) \\ & \text{subject to} & EU^b(I, x^b, t) = EY \qquad 0 \leq x_i^a \leq 1 \end{array}$$

Here I is fixed and EY is b's equilibrium utility for $t + 1$. Given this, a's equilibrium strategy for time t is the same as SA(t) (defined earlier) with TA replaced with TA-I, and

U^a replaced with EU^a. Likewise for agent b. The problem TA-I is also the standard fractional knapsack problem, and is solvable using a greedy approach.

On the basis of the above equilibrium, we will show how to find an optimal agenda, for the PDP, for each agent.

3.3 The Negotiation Agenda

As noted in Section 1, in many cases, the set of issues to be negotiated are choice variables. So, before negotiation begins, the agents must decide upon an *agenda* which we define as follows:

Definition 1. *In the context of the PDP, an agenda A^g of size $g \leq m$ is a set of g issues, i.e., $A^g \subseteq I$ such that $|A^g| = g$. Let AG^g denote the set of all possible agendas of size g.*

Definition 2. *For the PDP, an agenda (AA^g) of size $g \leq m$ is agent a's optimal agenda if*

$$AA^g = \text{arg max}_{X \in AG^g} EU^a(X, x^a, 1)$$

where x^a denotes a's equilibrium allocation (for agenda X, for $t = 1$). For b, AB^g is analogous. For the complete information setting, EU^a and EU^b are replaced with U^a and U^b respectively.

For the agenda I containing m issues, Section 3.2 showed how to find equilibrium outcomes (i.e., U^a, EU^a, U^b, and EU^b). Given this equilibrium, our problem now is to find each agent's optimal agenda: AA^g and AB^g for $1 \leq g \leq m$. We show how to find these agendas in Section 4.

Below, we focus on the case where one of the agents, say a, prefers different issues differently but b prefers all the issues equally[2]. So, for a, different issues have different weights but b has the same[3] weight for all the issues. We let the issues $\{1, \ldots, m\}$ be such that $w_i^a \leq w_{i+1}^a$ for $1 \leq i \leq n - 1$. Also, for agenda $X \in AG^g$, we let $X = \{1, \ldots, g\}$ be such that $w_{X_i}^a \leq w_{X_{i+1}}^a$ for $1 \leq i < g$. Note that, although we are viewing X as a list, as per its definition, an agenda is a set because the equilibrium for the PDP is independent of the ordering of issues. However, we view X as a list because it makes it easier to refer to the individual elements of X.

4 Optimal Agendas: Complete Information

We first show how to find optimal agenda for the complete information setting and then for the incomplete information setting described in Section 3.2. For the complete information setting, Theorem 1 (Theorem 2) shows how to find a's (b's) optimal agenda.

[2] This is situation occurs often: for the previous car dealer example, a buyer may have different preferences over the seat color and the car color, but the dealer may be indifferent between colors.

[3] Future work will deal with those situations where, for both a and b, the weights are different for different issues.

Then, for the incomplete information setting described in Section 3.2, Theorem 3 (Theorem 4) gives a's (b's) optimal agenda. Theorem 5 gives the time complexity of computing the optimal agendas.

Theorem 1. *In a setting where agent a has different weights for different issues and b has the same weight for all m issues, a's optimal agenda of size g is a set of g issues that are associated with the g highest weights in w^a (i.e., $AA^g = \{m-g+1, \ldots, m\}$).*

Proof. *Here AA^g is obtained using BI. If an agenda is optimal for a for the last time period $t = n$, then by BI, it will be optimal for all previous ones. Consider the last time period ($t = n$) for which two possibilities can arise: either a or b could be the last mover. Consider first the case where a is the last mover. As per the equilibrium for the PDP, irrespective of the agenda, a gets a 100% of all the shrunken pies. So, from among all possible agendas, its cumulative utility is maximized for the agenda $AA^g = \{m - g + 1, \ldots, m\}$.*

Now, consider the case where b is the last mover. Here, as per the equilibrium for PDP, b gets a 100% of all the shrunken pies and a gets nothing (i.e., a's utility for $t = n$ is zero). Also, since b has equal weights for all the issues, its cumulative utility for $t = n$ is independent of the agenda. We therefore consider the previous time period ($t = n-1$) for which it is a's turn to offer. As per the equilibrium for $t = n - 1$, b's utility must be equal to its equilibrium utility for $t = n$. If U^b denotes b's equilibrium utility for $t = n$, then in the equilibrium for $t = n - 1$, b must get U^b. Note that since b has equal weights, U^b is independent of the agenda. But a's utility depends on the agenda. Given U^b, a's equilibrium utility for $t = n - 1$ (for an agenda $X \in AG^g$) will be $U^a_X = (\sum_{i=1}^{g} w^a_{X_i}) - U^b$. Then, a's optimal agenda is the one that maximizes U^a_X. Since all weights are positive, and the issues in $X = \{1, \ldots, g\}$ are in ascending order of w^a_i, U^a_X is maximized when $X = \{m - g + 1, \ldots, m\}$. Thus, $AA^g = \{m - g + 1, \ldots, m\}$. □

Regarding b's optimal agenda, one might think that AB^g will be the set of g issues that correspond to the g lowest weights for a. But this is not so, because an agent's cumulative utility from an agenda depends not just on its weights but also on its equilibrium shares for the g issues. The following example clarifies this point.

Example 1. Let $m = 4$, $I = \{1, 2, 3, 4\}$, $g = 3$, $\delta = 0.5$, $n = 2$, $w^a = \{1, 2, 3, 4\}$, $w^b = \{1, 1, 1, 1\}$, and b be first mover. There are four possible agendas of size $g = 3$: $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, and $\{2, 3, 4\}$. For each of them, the agents' equilibrium utilities for $t = 1$ (i.e., U^a_1 and U^b_1) are as given in Table 1. Agent b's utility U^b_1 is highest for the agenda $\{1, 2, 4\}$, and so $AB^g = \{1, 2, 4\}$ is b's optimal agenda. Likewise, $AA^g = \{2, 3, 4\}$.

As Example 1 shows, the optimal agenda may be different for different agents. But in many practical cases, only one of the two agents has a choice over the agenda. This is also the case in the car dealer example (outlined in Section 1) where the buyer can choose an agenda but not the seller.

We will now show how to find b's optimal agenda AB^g using the following method. Initially, the optimal agenda is empty. Then, we add issues to it one by one using a greedy approach. For $1 \leq k \leq g$, let AB^g_k denote the issue that is b's optimal choice at step k. Then, b's optimal agenda AB^k of size k is $AB^k = \cup_{i=1}^{k} AB^g_i$.

Agenda	U_1^a	U_1^b	b's Optimal Agenda?
$\{2,3,4\}$	4.5	1.833	No
$\{1,2,3\}$	3	2	No
$\mathbf{\{1,2,4\}}$	$\mathbf{3.5}$	$\mathbf{2.125}$	**Yes**
$\{1,3,4\}$	4	2	No

Table 1. Agents' utilities (for $t = 1$) for possible agendas.

Theorem 2. *In a setting where agent a has different weights for different issues and b has the same weight for all of them, b's optimal agenda AB^g (for $g \geq 2$) is obtained as follows. Agent b's optimal choice for the first two issues is:*

$$AB_1^g = 1 \qquad \textit{and} \qquad AB_2^g = m.$$

Then AB_k^g ($3 \leq k \leq g$) is given by the following rule. If we let $\cup_{i=1}^{k-1} AB_i^g = \{1, \ldots, p, q, \ldots, m\}$ where ($1 \leq p < q \leq m$) and ($p + m - q + 1 = k - 1$), then if

$$\sum_{i=q}^{m} w_i^a \geq \delta \sum_{i=1}^{k-1} w_{AB_i^g}^a + \delta w_{p+1}^a \tag{1}$$

then $AB_k^g = p + 1$. Otherwise, if $\exists Z \in I - \cup_{i=1}^{k-1} AB_i^g$ such that

$$w_Z^a + \sum_{i=q}^{m} w_i^a \geq \delta \sum_{i=1}^{k-1} w_{AB_i^g}^a + \delta w_Z^a \tag{2}$$

then if

$$\delta \sum_{i=1}^{k-1} w_{AB_i^g}^a \geq \sum_{r=q}^{m} w_{AB_r^g}^a \tag{3}$$

then $AB_k^g = p + 1$. Otherwise (i.e., Eq. 2 or 3 is false), $AB_k^g = q - 1$.

Proof. *An agent's optimal agenda problem exhibits the 'greedy choice property' and the 'optimal sub-structure property'. So a globally optimal solution can be found by making locally optimal choices. A problem has 'optimal substructure' if an optimal solution can be constructed from optimal solutions to its subproblems. An agent's optimal agenda problem has both these properties since all weights are positive and so an agent's cumulative utility is maximized if its utility from the individual issues is maximized. Given that b has equal weights for all the issues, b's cumulative utility is maximized if the issues are chosen such that its cumulative equilibrium share for them is the maximum over all possible agendas. So if x_X^b denotes b's equilibrium share for agenda X, then $AB^g = \arg\max_{X \in AG^g} \sum_{i=1}^{g} x_{X_i}^b$ or $AB^g = \arg\min_{X \in AG^g} \sum_{i=1}^{g} x_{X_i}^a$ (since, $x_i^a = 1 - x_i^b$). To find AB^g, we will choose one issue at a time. At step k ($1 \leq k \leq g$) we will choose AB_k^g such that a's shares for the issues chosen thus far are minimized (relative to all possible agendas of size k). In more detail, this is done as follows.*

As before, if an agenda is optimal for b for $t = n$, then by BI, it will be optimal for all previous time periods. Now, either a or b could be the last mover. If it is a, it gets a 100% of all the shrunken pies and b gets zero utility. But if b is last mover, then it gets a 100% of all shrunken pies and a gets nothing. So irrespective of who the last mover is, b's utility for $t = n$ is independent of the agenda. Hence, we look at the previous time period $t = n - 1$ to find AB^g. Consider the case where b is the offering agent for $t = n - 1$. If x^a_X denotes a's equilibrium share for $t = n - 1$, then b's optimal agenda for this time period is $AB^g = arg\ min_{X \in AG^g} \sum_{i=1}^{g} x^a_{X_i}$.

To begin, we show how to find the first and second issues (i.e., AB^g_1 and AB^g_2) that will be included in AB^g. Then, we will show how to find issues one by one (using a greedy approach) to include in AB^g until the number of issues in it is g. Let the m issues $\{1, \ldots, m\}$ be such that $w^a_i \leq w^a_{i+1}$ for $1 \leq i < m - 1$. We decide what AB^g_1 and AB^g_2 should be on the basis of the relation between $w^a_{AB^g_1}$ and $w^a_{AB^g_2}$. Irrespective of what these two issues are, there are two[4] possible relations between their weights: $w^a_{AB^g_1} < w^a_{AB^g_1}$ or else $w^a_{AB^g_1} > w^a_{AB^g_1}$. We consider each of these below.

***For the case** $w^a_{AB^g_1} < w^a_{AB^g_2}$: At this stage, we know the relation between $w^a_{AB^g_1}$ and $w^a_{AB^g_2}$ but not the actual weights. Given this relation, as per the equilibrium for $t = n-1$ (see Section 3.2), b will first allocate a share to a for the issue AB^g_2 and then for AB^g_1. This will be done such that a's cumulative utility (say Y) from AB^g_1 and AB^g_2 is equal to its cumulative utility for $t = n$ (i.e., $Y = \delta^{n-1} \sum_{i=1}^{2} w^a_{AB^g_i}$). When doing this allocation for $t = n - 1$, one of two possible cases (C1.1 or C1.2) can arise:*

C1.1 *The entire utility Y can be given to a from the issue AB^g_1 alone so a's share for AB^g_2 is zero, so we have:*

$$w^a_{AB^g_2} \delta^{n-2} \geq (w^a_{AB^g_1} + w^a_{AB^g_2}) \delta^{n-1} \tag{4}$$

If $x^a_{AB^g_2}$ denotes a's equilibrium share for AB^g_2 for $t = n - 1$, then as per Section 3.2, we have:

$$x^a_{AB^g_2} = (w^a_{AB^g_1} + w^a_{AB^g_2}) \delta^{n-1} / w^a_{AB^g_2} \delta^{n-2} \tag{5}$$

Clearly, it is optimal for b to choose AB^g_1 and AB^g_2 such that $x^a_{AB^g_2}$ is minimized. In Equation 5, since δ is a constant, $x^a_{AB^g_2}$ is minimized when $w^a_{AB^g_1}$ is minimized and $w^a_{AB^g_2}$ is maximized. Thus, AB^g_1 is the issue with lowest weight in w^a, and AB^g_2, is the one with highest weight in w^a. So $AB^g_1 = 1$ and $AB^g_2 = m$.

C1.2 *The utility Y cannot be given just from the issue AB^g_2 so a's share for AB^g_2 is a 100% of it and its share for AB^g_1 is non-zero, so Equation 4 is false. and if $x^a_{AB^g_1}$ denotes a's equilibrium share for AB^g_1 for $t = n - 1$, then as per Section 3.2, we have:*

$$(w^a_{AB^g_1} x^a_{AB^g_1} + w^a_{AB^g_2}) \delta^{n-2} = (w^a_{AB^g_1} + w^a_{AB^g_2}) \delta^{n-1}, \quad \textit{or}$$

[4] For the case $w^a_{B^1} = w^a_{B^2}$, the agents' equilibrium shares are independent of $w^a_{AB^g_1}$ and $w^a_{AB^g_1}$. This can be verified by substituting $w^a_{AB^g_1} = w^a_{AB^g_1}$ in Equations 5 and 6.

$$x^a_{AB^g_1} = \delta - \frac{w^a_{AB^g_2}}{w^a_{AB^g_1}}(1-\delta) \tag{6}$$

Here, $x^a_{AB^g_1}$ is minimized when $w^a_{AB^g_2}$ is maximized and $w^a_{AB^g_1}$ is minimized. So $AB^g_1 = 1$ and $AB^g_2 = m$.

Thus, if $w^a_{AB^g_1} < w^a_{AB^g_2}$, then $AB^g_1 = 1$ and $AB^g_2 = m$ for both cases (C1.1 and C1.2).

***For the case** $w^a_{AB^g_1} > w^a_{AB^g_2}$: This case is the same as the previous one, with $w^a_{AB^g_1}$ and $w^a_{AB^g_2}$ swapped.*

Now, we will choose AB^g_3. Since $AB^g_1 = 1$ and $AB^g_2 = m$, we have $w^a_1 < w^a_{AB^g_3} < w^a_m$. So, in the equilibrium for $t = n-1$, b will first allocate to a, a share for the issue AB^g_2, then for AB^g_3, and finally for AB^g_1. In the equilibrium allocation for $t = n-1$, a's utility is $Y = \delta^{n-1}\sum_{i=1}^{3} w^a_{AB^g_i}$. We find a's equilibrium shares in this allocation by considering the following three possible cases (C3.1 that corresponds to Eqn. 1 being true, C3.2 that corresponds to Eqn. 1 being false and Eqn. 2 being true, or C3.3 that corresponds to Eqns. 1 and 2 being false) that can arise:

C3.1 *The entire utility Y can be given to a from the issue AB^g_2 alone so a's share for AB^g_1 and AB^g_3 is each zero. So we have:*

$$w^a_{AB^g_2}\delta^{n-2} \geq \sum_{i=1}^{3} w^a_{AB^g_i}\delta^{n-1} \tag{7}$$

If $x^a_{AB^g_2}$ denotes a's equilibrium share for AB^g_2 for $t = n-1$, then as per Section 3.2, we have:

$$x^a_{AB^g_2} = (\sum_{i=1}^{3} w^a_{AB^g_i})\delta^{n-1}/w^a_{AB^g_2}\delta^{n-2} \tag{8}$$

Here, $x^a_{AB^g_2}$ is minimized by choosing as AB^g_3 the issue with lowest weight from the remaining issues in $I - \{AB^g_1 \cup AB^g_2\}$. This gives $AB^g_3 = 2$. Moreover, Equation 7 is true iff $AB^g_3 = 2$ because, in $I - \{AB^g_1 \cup AB^g_2\}$, 2 is the issue with least weight.

C3.2 *The utility Y cannot be given just from AB^g_2 but can be given from AB^g_2 and AB^g_3. So a's share for AB^g_2 is a hundred percent of it and its share for AB^g_3 is non-zero, i.e., Equation 7 is false $\forall AB^g_3 \in I - \{AB^g_1 \cup AB^g_2\}$ and $\exists AB^g_3 \in I - \{AB^g_1 \cup AB^g_2\}$ s.t.:*

$$(w^a_{AB^g_2} + w^a_{AB^g_3})\delta^{n-2} \geq \sum_{i=1}^{3} w^a_{AB^g_i}\delta^{n-1} \tag{9}$$

If $x^a_{AB^g_3}$ denotes a's equilibrium share for AB^g_3 for $t = n-1$, then as per Section 3.2, we have:

$$(w^a_{AB^g_3}x^a_{AB^g_3} + w^a_{AB^g_2})\delta^{n-2} = \sum_{i=1}^{3} w^a_{AB^g_i}\delta^{n-1} \tag{10}$$

Solving the above equation for $x^a_{AB^g_3}$, we get:

$$x^a_{AB^g_3} = \frac{\sum_{i=1}^{2} \delta w^a_{AB^g_i} - w^a_{AB^g_2}}{w^a_{AB^g_3}} + \delta$$

Here, if Eqn. 3 is true, then $x^a_{AB^g_3}$ is minimized if $w^a_{AB^g_3}$ is maximized. So AB^g_3 must be the issue with lowest weight from the remaining issues, or $AB^g_3 = 2$. But if Equation 3 is false, $x^a_{AB^g_3}$ is minimized when $AB^g_3 = m - 1$.

C3.3 *The entire utility Y cannot be given from AB^g_2 and AB^g_3 so a's share for each of AB^g_2 and AB^g_3 is a hundred percent and its share for AB^g_1 is non-zero, i.e., Equations 7 and 9 are false $\forall AB^g_3 \in I - \{AB^g_1 \cup AB^g_2\}$, so:*

$$(\sum_{i=1}^{3} w^a_{AB^g_i})\delta^{n-2} \geq \sum_{i=1}^{3} w^a_{AB^g_i}\delta^{n-1}$$

If $x^a_{AB^g_1}$ denotes a's equilibrium share for AB^g_1 for $t = n - 1$, then as per Section 3.2, we have:

$$(w^a_{AB^g_1}x^a_{AB^g_1} + w^a_{AB^g_2} + w^a_{AB^g_3})\delta^{n-2} = \sum_{i=1}^{3} w^a_{AB^g_i}\delta^{n-1}$$

Solving the above equation for $x^a_{AB^g_1}$, we get:

$$x^a_{AB^g_1} = (\delta w^a_{AB^g_1} + \delta w^a_{AB^g_2} - w^a_{AB^g_2} - w^a_{AB^g_3}(1 - \delta))/w^a_{AB^g_1}$$

Here, $x^a_{AB^g_1}$ is minimized when $AB^g_3 = m - 1$.

For C3.1, $AB^g_3 = 2$, for C3.2 $AB^g_3 = 2$ or $AB^g_3 = m - 1$ (depending on Equation 3), and for C3.3 $AB^g_3 = m - 1$.

In general, to decide what to choose as AB^g_k ($4 \leq k \leq g$), we find a's equilibrium allocation for each of the k issues (chosen thus far) for $t = n-1$. At stage k,we already have the optimal agenda of size $k - 1$, i.e., $\cup_{i=1}^{k-1} AB^g_i$. Also, we know that for stage i, ($1 \leq i \leq k - 1$) b's optimal choice is either the issue with highest weight for a or else the one with the lowest weight from the remaining issues $I - \{\cup_{j=1}^{i-1} AB^g_j\}$. Hence, as mentioned in the statement of this theorem, we let $\cup_{i=1}^{k-1} AB^g_i = \{1, \ldots, p, q, \ldots, m\}$ where ($1 \leq p < q \leq m$) and ($p+m-q+1 = k-1$). This implies that $w^a_p < w^a_{AB^g_k} < w^a_q$. Also, as before, the equilibrium allocation for k issues for $t = n - 1$ must give a a utility of $Y = \delta^{n-1} \sum_{i=1}^{k} w^a_{AB^g_i}$. We find a's equilibrium shares in this allocation by considering the following three possible cases (CK.1 that corresponds to Equation 1 being true, CK.2 that corresponds to Equation 1 being false and Equation 2 being true, or CK.3 that corresponds to Equations 1 and 2 being false) that can arise:

CK.1 *The cumulative utility Y can be given to a just from the pies in $\{j, \ldots, m\}$ (where $q \leq j \leq m$) so we have*[5]*:*

$$\delta^{n-2} \sum_{r=j}^{m} w^a_{\alpha_r} \geq \delta^{n-1} \sum_{i=1}^{k} w^a_{AB^g_i} \tag{11}$$

If $x^a_{AB^g_k}$ denotes a's equilibrium share for AB^g_k for $t = n-1$, then as per Section 3.2, we have:

$$x^a_j = \big(\delta \sum_{i=1}^{k} w^a_{AB^g_i} - \sum_{r=j+1}^{m} w^a_r\big)/w^a_j \tag{12}$$

Here, x^a_j is minimized if $w^a_{AB^g_k}$ is minimized. So b's optimal choice for AB^g_k is the issue with lowest weight from the remaining issues $I - \{\cup_{i=1}^{k-1} AB^g_i\}$, i.e., $AB^g_k = p+1$. Moreover Equation 11 is true iff $AB^g_k = p+1$ because, in $I - \{\cup_{i=1}^{k-1} AB^g_i\}$, $p+1$ is the issue with least weight.

CK.2 *The entire utility Y cannot be given just from the pies in $\{q, \ldots, m\}$ but can be given from $\{q, \ldots, m\}$ together with AB^g_k, i.e., $\exists AB^g_k \in I - \{\cup_{i=1}^{k-1} AB^g_i\}$ such that:*

$$\delta^{n-2} \sum_{r=q}^{m} w^a_r < \delta^{n-1} \sum_{i=1}^{k} w^a_{AB^g_i}, \quad \textit{and} \tag{13}$$

$$\delta^{n-1}\big(w^a_{AB^g_k} x^a_{AB^g_k} + \sum_{r=q}^{m} w^a_r\big) \geq \delta^{n-2} \sum_{i=1}^{k} w^a_{AB^g_i} \tag{14}$$

If $x^a_{AB^g_k}$ denotes a's equilibrium share for AB^g_3 for $t = n-1$, then as per Section 3.2, we have:

$$x^a_{AB^g_k} = \frac{\delta \sum_{i=1}^{k-1} w^a_{AB^g_i} - \sum_{r=q}^{m} w^a_r}{w^a_{AB^g_k}} + \delta \tag{15}$$

Here, if Eqn. 3 is true, then $x^a_{AB^g_k}$ is minimized if $x^a_{AB^g_k}$ is maximized, i.e., $AB^g_k = p+1$. But if Eqn. 3 is false, $x^a_{AB^g_k}$ is minimized if $w^a_{AB^g_k}$ is minimized, i.e., $AB^g_k = q-1$.

CK.3 *The cumulative utility Y cannot be given just from the pies in $\{AB^g_k, q, \ldots, m\}$ but can be given from $\{j, \ldots, r, AB^g_k, q, \ldots, m\}$ where $1 \leq j \leq p$, so we have:*

$$\delta^{n-2}(w^a_{AB^g_k} + \sum_{r=q}^{m} w^a_r) < \delta^{n-1} \sum_{i=1}^{k} w^a_{AB^g_i}, \quad \textit{and}$$

[5] Here, a's share for each of the pies $j+1, \ldots, m$ is one, for pie j it is x^a_j, and for each of the pies $1, \ldots, j-1$, is zero.

$$\delta^{n-1}\big(\sum_{r=j}^{p} w_r^a + w^a_{AB_k^g} + \sum_{r=q}^{m} w_r^a\big) \geq \delta^{n-2}\sum_{i=1}^{k} w^a_{AB_i^g}$$

If $x^a_{AB_k^g}$ denotes a's equilibrium share for AB_k^g for $t = n-1$, then as per Section 3.2, we have:

$$x_j^a = \big(\delta\sum_{i=1}^{k} w^a_{AB_i^g} - \sum_{r=q}^{m} w_r^a - w^a_{AB_k^g} - \sum_{r=j+1}^{p} w_r^a\big)/w_j^a$$

Here x_j^a is minimized when $w^a_{AB_k^g}$ is maximized. So AB_k^g must be the issue with highest weight from the remaining issues $I - \{\cup_{i=1}^{k-1} AB_i^g\}$, i.e., $AB_k^g = q-1$.

For CK.1, $AB_k^g = p+1$, for CK.2 $AB_k^g = p+1$ or $AB_k^g = q-1$ (depending on Eqn. 3), and for CK.3 $AB_k^g = q-1$.

In the same way, we can obtain b's optimal agenda for the case where a is the offering agent at $t = n-1$. □

5 Optimal Agendas: Incomplete Information

We will now show how to find optimal agendas for the incomplete information setting of Section 3.2. To this end, Theorem 3 gives a's optimal agenda and Theorem 4 that for b. Finally, Theorem 5 gives the time complexity of computing these optimal agendas. Let the m issues $\{1,\ldots,m\}$ be such that: $ew_1^a \leq ew_2^a \leq \ldots \leq ew_m^a$. Here, agent a has different expected weights for different issues, but b has the same expected weight for all the m issues.

Theorem 3. *In a setting where agent a has different expected weights for different issues but, for b, the set of possible weights for an issue and the associated probabilities are the same for all the m issues, agent a's optimal agenda of size g is a set of g issues that are associated with the g highest expected weights (i.e., $AA^g = \{m-g+1,\ldots,m\}$).*

Proof. *As Theorem 1, with weights replaced with corresponding expected weights and utilities with corresponding expected utilities.* □

Theorem 4. *In a setting where agent a has different expected weights for different issues but, for b, the set of possible weights for an issue and the associated probabilities are the same for all the m issues, b's optimal agenda AB^g (for $g \geq 2$) is obtained as in Theorem 2 with agent's weights replaced with corresponding expected weights, and utilities replaced with corresponding expected utilities.*

Proof. *As Theorem 2, with weights replaced with corresponding expected weights and utilities with corresponding expected utilities.* □

Theorem 5. *For the complete information setting, the time taken to compute AA^g is $\mathcal{O}(g)$ and to compute AB^g is $\mathcal{O}(mg^2)$. For the incomplete information setting, the time to compute AA^g is $\mathcal{O}(Tg)$ and to compute AB^g is $\mathcal{O}(Tmg^2)$.*

Proof. *For the complete information setting, as per Theorem 1, AA^g is the set of g issues with highest weight in w^a. We therefore need to choose the last g issues from α which takes time $\mathcal{O}(g)$. For AB^g, as per Theorem 2, determining AB^g_k ($1 \leq k \leq g$) requires evaluating the conditions in Eqns. 1, 2, and 3. Consider Equation 2 which requires one comparison between $w^a_Z + \sum_{i=q}^{m} w^a_i$ and $\delta \sum_{i=1}^{k-1} w^a_{AB^g_i} + \delta w^a_Z$. Computation of these two terms to compare requires no more than k additions because the summation variable i varies between 1 and k. Also, Z can vary at most between 1 and m. So finding AB^g_k takes $\mathcal{O}(mk)$ time, and to find AB^g_k for $1 \leq k \leq g$, it takes $\sum_{k=1}^{g} \mathcal{O}(mk) = \mathcal{O}(mg^2)$ time. Note that we considered only Eqn. 2 because the time to evaluate the condition in Eqns. 1 or 3 is no more than the time taken to evaluate the condition in Eqn. 2.*

It follows that, for the incomplete information setting, if there are T possible utility function pairs, the time to compute AA^g is $\mathcal{O}(Tg)$ and to compute AB^g is $\mathcal{O}(Tmg^2)$. □

6 Conclusions and Future Work

This paper presented polynomial time methods for finding each agent's optimal agenda for the PDP. The polynomial time complexity of our methods makes it easy for automating the process of choosing an agent's optimal agenda, and thereby reducing human involvement during negotiation.

Possible avenues for future research include extending the current analysis to scenarios where both agents have different weights for different issues. Also, this paper focussed on one specific incomplete information setting. In future, we will extend this analysis to other possible incomplete information settings.

Bibliography

[1] M. Bac and H. Raff. Issue-by-issue negotiations: the role of information and time preference. *Games and Economic Behavior*, 13:125–134, 1996.

[2] L. A. Busch and I. J. Horstman. A comment on issue-by-issue negotiations. *Games and Economic Behavior*, 19:144–148, 1997.

[3] T. H. Cormen, C. E. Leiserson, R. L Rivest, and C. Stein. *An introduction to algorithms*. The MIT Press, Cambridge, Massachusetts, 2003.

[4] S. S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal agendas for multi-issue negotiation. In *Proceedings of the Second International Conference on Autonomous Agents and Multi-Agent Systems*, pages 129–136, Melbourne, Australia, July 2003.

[5] S. S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda based framework for multi-issue negotiation. *Artificial Intelligence Journal*, 152(1):1–45, 2004.

[6] S. S. Fatima, M. Wooldridge, and N. R. Jennings. Multi-issue negotiation with deadlines. *Journal of Artificial Intelligence Research*, 27:381–417, 2006.

[7] C. Fershtman. The importance of the agenda in bargaining. *Games and Economic Behavior*, 2:224–238, 1990.

[8] C. Fershtman. A note on multi-issue two-sided bargaining: bilateral procedures. *Games and Economic Behavior*, 30:216–227, 2000.

[9] D. Fudenberg and J. Tirole. Sequential bargaining with incomplete information. *Review of Economic Studies*, 50:221–247, 1983.

[10] S. Kraus. *Strategic negotiation in multi-agent environments*. The MIT Press, Cambridge, Massachusetts, 2001.

[11] D. A. Lax and J. K. Sebenius. Dealcrafting: The substance of three dimensional negotiations. *Negotiation Journal*, 18(1):5–28, 2002.

[12] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[13] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[14] J. K. Sebenius. Negotiation arithmetic: Adding and subtracting issues and parties. *International Organization*, 37(2):281–316, 1983.

A Practical Multiagent Model for Resilience in Commercial Supply Networks

Andrew Smith[1] and José M. Vidal[1]

University of South Carolina, Columbia, SC 29208, USA

Abstract. As commercial supply chains grow into complex global supply networks, more and greater risks are introduced for cooperating and competing companies alike. These networks can be affected by events such as natural disasters, terrorism, and of late, economic downturn. Supply industry leaders, such as IBM, have announced a need for methods to identify and prevent risks in these ever-growing complex networks. Multiagent-based simulation lends itself perfectly to supply network modeling due to its autonomous nature. Our research illustrates a multiagent supply network formation technique using greedy supply agents and limited resource allocation. Using these formations, the resilience of each network is compared with others and assessed so that we may ascertain the characteristics of risky supply network structure. Our results show that an increase in relationship resources results in a more resilient network; however, as the amount of available resources increases, the risk of the most vulnerable agent in the network decreases by a smaller margin.

1 Introduction

The need for resilience in supply networks is a concern for many. During combat, the military is concerned because convoys and supply stations are highly susceptible to enemy attack and disruption [12, 9]. More recently, with the rough global economy, we are seeing businesses declaring bankruptcy and going out of business at an alarming rate; from the end of the year 2007 to the end of 2008, business bankruptcies rose 54%, and are continuing to rise[1]). These closures cause disturbances and possible breaks in supply networks, especially as the world's leading manufacturers, like in the automotive industry, are starting to fail. Other global factors such as terrorism and severe weather conditions also have been known to cause commercial supply networks to come to a halt [4, 9]. As the globalization of supply networks becomes more common, these networks also become more complex and thus increase the chances of global factors affecting larger number of businesses [4].

Methods such as just-in-time (JIT) inventory to improve efficiency in supply chain management have been geared towards making a company the most profit with the least amount of inventory on hand. This approach unfortunately creates

[1] `www.bankruptcyaction.com`

weaknesses in the supply network's reliability [9]. Since the terrorist attacks of September 11th, 2001 and the current downturn in the economy, efforts have shifted to try and improve reliability in supply networks [4].

Our long-term goal is to build dynamic agent-based model of these supply networks so that we may study how they handle disturbances: missing nodes, broken routes, order delays, etc. We hope to model both the current human-made supply networks, where all the decisions about who to buy from and what to buy are made by humans, as well as the emerging agent/human networks where some of these decisions are made by automated agents. This paper presents our first steps towards that goal. We use an agent-based model of human-formed supply networks, based on [6], to generate supply networks and analyze these networks to determine their resilience to various types of attacks. Our test results provide quantitive measures of the resilience of networks formed by humans given different capacities to form social ties.

1.1 Previous Research

IBM and several other sources have noted the importance of maintaining a reliable supply network [2, 9]. Several examples exist of major losses of profit and business due to supply chain disruptions. Because of this, there is a need to reconsider how supply networks are setup, as well as the processes involved in these networks.

Several examples exist of multiagent supply network formations as simulating the day to day operations of supply networks [13, 10], but neither assess the reliability of the network as a whole. As recent research has shown, however, complex networks need to cooperate with other agents in the network, even with competing agents to some extent [5]. Using the customer lifetime value equation developed by V. Kumar, Chuan and Yun developed a multiagent market that models how consumers interact with suppliers [6, 3]. In this paper we will extend this method to an entire supply network.

A topological method for developing a reliable supply network for military settings was developed to ensure that suppliers could get goods to troops, even in the event of random or planned attacks [12]. This model resulted in a high level of redundancy between suppliers and their consumers. However, as the authors of [12] noted, such large amounts of interconnectivity are not practical in a commercial market. In order to accrue decent profit and maintain a competitive market, partnerships must be selective. Making such a large amount of connections and trade agreements takes a considerable amount of management time, and often decreases the quality of business relationships and profit. Reducing the similar property of interaction costs in auctions was recently studied [15]. Agents in our model consider these costs in network formation, and the reduction of these costs will undoubtedly be an incentive in our future work. This behavior of resource management can also be related to personal social management [2][13] [10], and will be compared throughout the paper. This paper presents a topological approach similar to that of [12], but geared towards identifying resilient network structures in a competitive, commercial market.

Resilience, as opposed to reliability, refers to a network's ability to respond to attacks or disabled nodes.

The multiagent community has studied the problem of how automated agents should make decisions in is supply chain, especially in the context of the Trading Agent Competition Supply Chain Management (TAC SCM) game[2][1, 8]. However, we are interested in studying the reliability of human-formed networks which we assume to be static, for now. In our future work we will expand the model to include dynamically trading agents as well as allow agents to dynamically find new partners when necessary.

2 The Model

We are interested in investigating the resilience of supply networks that are formed by selfish agents. We start by identifying the various types of agents in a supply chain (section 2.1). We then explain the various types of ties that can exist between these agents (section 2.2) and use proven models of supply-chain tie formation to create our networks (section 2.3). Finally, we formally describe how we measure resilience in a supply chain network.

2.1 Agent Composition

We model five different types of business agents: suppliers, manufacturers, distributors, retailers, and consumers. Each agent's identity determines who they interact and exchange product with. At any time, an agent has one of two roles: a local supplier or a local customer. A local supplier is one who is supplying a product while a local customer is one who is purchasing a product. All agents, except for the suppliers and the consumer, can take on both of these roles. The suppliers can only take on the supplier role, and the consumer can only take on the customer role. A very simple supply chain is shown in figure 1. Along with their roles, each agent also has a number of properties that describe its capabilities. Table 1 shows these properties along with their descriptions.

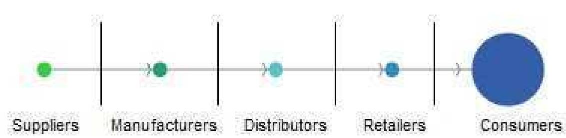


Fig. 1. Organization of a supply network.

Once an agent (except for suppliers) is initialized, their price, profit, reliability, quality, supply and demand are set to zero. The values stay at zero until

[2] `http://www.sics.se/tac/page.php?id=13`

Table 1. Properties of Business Agents

Price	The selling price of this agent's product
Quality	Quality of this agent's product
Reliability	Reliability of this agent's product
Profit	The amount of the selling price that goes towards profit
Customer List	A CLV-ordered list of this agent's customers
Score List	An ordered list of desired suppliers
Supply	The amount of product coming in to this agent
Demand	The amount of product desired by this agent
Relationship Resources	Amount of relationship resources available to this agent (discussed in section 2.2)

relationships are formed. Suppliers' price is set to the raw good price, and their profit, quality, reliability, supply and demand are set randomly. Their customer and score list is set to empty until negotiations begin, which we discuss later in section 2.3. Relationship resources are set by the user before the model is initialized. Notice that since supplier and consumer agents only take on one of the two roles available, they only have the properties necessary to satisfy these roles. That is, suppliers do not have a score list and consumers do not have a customer list, price or profit.

The number of agents can vary from run to run, to provide varying network structures. We only model one consumer agent. The reason for this is that supply networks usually make relationships based on contracts, whereas consumers often make one-time purchases. The interaction between an individual consumer and a market is discussed in [3], and our work could be extended to include that research. However, the focus of this paper is on resilience of the supply network so we opted not to model individual customer behavior.

Relationships are formed based on maximization of each agent's utility. This utility function differs depending on the current role the agent is playing. If an agent is in the supplier role the we use Equation (2) as the utility function. If an agent is in the customer role, the agent with the highest combination of price, reliability and quality wins. The determination of how strong the relationship will be, and what this strength implies, is discussed in the next section.

2.2 The Importance of Ties

Managing relationships between business agents can get costly depending on the number and strength of each tie. In social relationships, on average, the human brain can manage 150 strong relationships determined by age, frequency of interaction, emotional attachment, reciprocity, and kinship [11]. This limited amount of relationship resources can be reflected to supply network management in different units, like time and money [7]. In commercial networks the proper distribution of these limited relationship resources among an agent's strong and weak ties is critical for maintaining high profitability and reliability.

The definitions of strong and weak ties are very similar for social and supply networks [7]. In commercial supply networks, strong ties indicate a day-to-day relationship. Businesses linked by strong ties engage in frequent orders and shipments, thus undergoing more reliable and predictable transactions as a result. They can also show a parallel in business practices and ideas. Businesses linked by weak ties acknowledge each others' product needs, but are not regularly involved in transactions. These ties exist as bridges for possible future needs. If a weak tie is made, these transactions will not be as reliable, predictable, or cheap in price as those between two businesses with a strong tie due to the unfamiliarity and higher cost in planning.

In our model, business agent relationship resources are a user-controlled variable which allows us to set up different kinds of networks. An agent establishes relationships with other agents by using these resources. Relationships in the simulation can run from a range of 0 to 10, where 0 is a nonexistent relationship and 10 is the strongest relationship. Relationships in the range of $0-4$ are considered weak ties and are represented by a thin gray directed arrow. Relationships from $5-10$ are considered strong and are represented by a thick black directed arrow, as shown in Figure 2. For example, if an agent has 5 available relationship resources, it may form one strong relationship of 5 with a customer, or it could form 5 weak relationships of 1 with 5 customers. These relationships form during the negotiation stages of network formation.

2.3 Communication and Negotiation

Once the agents are initialized, communication between the different tiers of the network begins. A weak tie is temporarily established between all the agents to exchange product and company information, so each agent playing the customer role can evaluate their potential suppliers. Customers assign each supplier a score using the formula

$$\text{quality} + \text{reliability} + \left(\frac{\text{mean price} - \text{price}}{\text{price}} \times 100\right), \tag{1}$$

where quality, reliability and price are as defined in Table 1. This formula was established to ensure that if a price is below the mean price of all suppliers, a negative score is produced, unless quality and reliability are enough to offset it. These scores are then translated to desired relationship strengths, in the range of 1– 10. These desired relationships are stored in the score list agent property.

After the communication stage, negotiations begin. The first stage involves the customer agents sending their supplier assessment to each supplier. Generally, this information is not broadcast like this, but since there are no past interactions to evaluate, suppliers need a way to know how customers assess their service.

Suppliers then assess the customers using V. Kumar's [6] customer lifetime value formula

$$CLV = \sum_{y=1}^{T_i} \frac{CM_{i,y}}{(1+r)^{\frac{y}{f_i}}} + \sum_{i=1}^{n} \frac{\sum_m c_{i,m,l} \times x_{i,m,l}}{(1+r)^{i-1}}, \tag{2}$$

where CLV stands for the customer lifetime value of the current agent, T_i is the predicted number of purchases for customer i in a given time interval, $CM_{i,y}$ is the contribution margin of customer i during purchase y, r is the discount rate, f_i is the predicted purchase frequency for customer i, n is the number of years predicted for the relationship, $c_{i,m,l}$ is the marketing cost for customer i in market m during year l and $x_{i,m,l}$ is the number of other suppliers customer i is in a relationship with in market m during year l.

This formula is reduced in our experiments, since there is no past purchase history, the discount rate (r) is zero. Also, since there is only one consumer base, there is only one channel. The reduced CLV formula is

$$CLV = \sum_{y=1}^{T_i} CM_{i,y} - (c_i \times x_i). \tag{3}$$

The number of predicted purchases a customer makes (T_i) is determined by their desired relationship with the supplier. The marketing cost (c_i) is the maximum possible relationship value (10 for this model) minus the desired relationship with the supplier. The current number of other suppliers (x_i) is the count of all those with higher desired relationships. The corresponding CLVs for each customer are stored in the supplier's customer list. Only those with positive CLVs are kept in this list. Since negative values indicate a predicted profit loss, they are not beneficial to the supplier. These CLVs are then translated to desired relationship values based on the mean CLV and the number of resources available (similar to the customer's relationship conversion described above).

Once suppliers calculate their desired relationship, they ask their customers for this relationship. Customers receive the request, and first see if there are enough relationship resources available to accept the proposed relationship. If there are, the relationship is accepted and formed. However, if the relationship would cause the customer to exceed their available relationship resources, a pruning process ensues. The inquiring suppliers score is compared with the score of those suppliers who are currently in a relationship with the customer. Those suppliers who are ranked lower have their relationship reduced until there are enough relationship resources available for the inquiring supplier, or until the relationship dies. If there are still not enough relationship resources to include the inquiring supplier, the proposed relationship is reduced until the relationship can be made. The weight of this relationship determines how much product, by percent, is sent to the customer. For example if 50% of a suppliers resources are allocated to a customer, that customer will get 50% of the supplier's product.

When the negotiation process has ended, the customers calculate their price, reliability and quality based on the weighted average of all its suppliers. Their demand and initial supply are set to the total number of incoming product from their suppliers. This negotiation process is repeated down to the consumer agent.

Once the consumer agent is reached, the supply/demand ratio is set to 1 initially, since the supply is equal to demand. The importance of this value is discussed section 2.4. The entire negotiation process described is highlighted in Table 2.

Table 2. Steps for Agent Negotiation

Step	Task
1	Customers evaluate suppliers
2	Customers send evaluations to suppliers
3	Suppliers evaluate customers using CLV
4	Suppliers request relationship with customers
5	Customers accept, deny, or reduce relationships accordingly

We note that this negotiation process sometimes results in isolated agents who have no suppliers or customers. This is the result of a saturated market, and the isolated agents are those unfit for competition. Figure 2 shows the result of a completely formed network using the methods described above.

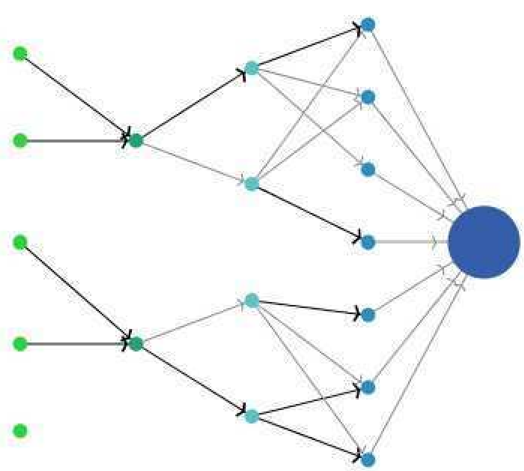


Fig. 2. Complete network formation after agent negotiations.

2.4 Resilience

After the network is established, it is tested for resilience. Testing for resilience simulates the attack or disabling of a single agent. We then measure the effect of its removal on the entire supply network. The method begins by eliminating an agent and its relationships. The effects of this elimination are then propagated down the network until the consumer is reached. The resulting supply is compared to the consumer's demand and measured as the supply/demand ratio. That is, the **supply/demand ratio** of agent i in network N is given by

$$r_i(N) = \frac{\text{amount of product arriving to the consumers in } N - i}{\text{amount of product demanded by the consumers in } N}. \tag{4}$$

The lower the ratio, the harder it will be for the network to recover from the attack. The agent is then placed back in the network, and the network is returned to normal. Once all agents have been tested for resilience, the agent with the lowest $r_i(N)$ is saved along with its corresponding ratio value. We define the **resilience** of a network as the supply/demand ratio of the agent with the lowest supply/demand ratio in the network. That is, the resilience of network N is given by

$$r(N) = \min_{i \in N} r_i(N). \tag{5}$$

We also save the the **variance of the supply/demand** ratio across all agents, which we denote as $\sigma^2(r_i(N))$. A higher variance in the supply/demand ratio means that there are some nodes which are much more important to the well-being of the supply network than others. Thus, these nodes might be more at risk for an attack by an enemy.

3 Results

The results gathered for this experiment are from over 400,000 different network structures with varying number of agents, relationship resources, and raw good price. Agents were varied (1 – 15 of each) to see how different relationships and market competition would affect resilience. Relationship resources were varied (5, 10, 20, and 50) to see how selectivity and the number of relationships each agent has affect resilience. Raw good price was varied to see if more expensive products, where the cost of relative distance is relatively small compared to product cost, affect network formation and resilience. Figures 3–5 show the difference in network formation and resource allocation with different variable settings.

Figure 3 shows two networks with varying numbers of agents. Though the total number of agents in the network is equal, the networks formed are very different, and undoubtedly result in different resilience factors. Figure 4 illustrates the critical difference between resource availabilities. The network with 50 resources clearly results in a more connected network, and, in this particular case, a more resilient network. Figure 5 shows how the price of the product does not greatly impact the network structure. This example also suggests that locality of customers may matter less as product price increases, but it is not significant.

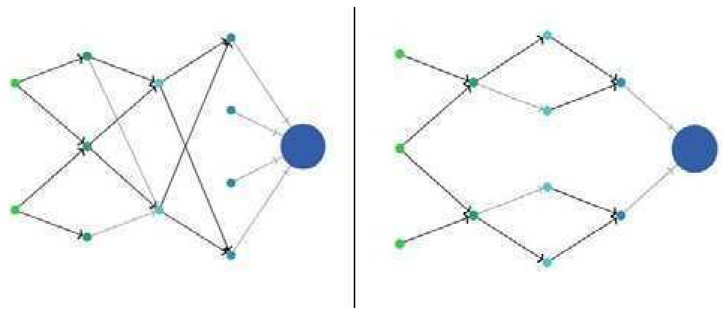


Fig. 3. Varying the number of agents in supply networks

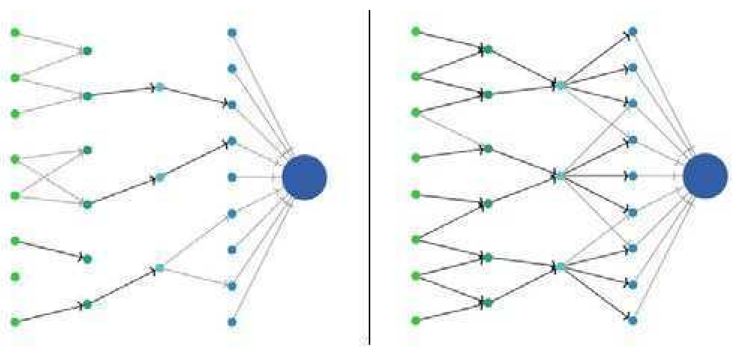


Fig. 4. 5 Relationship Resources vs. 50 Relationship Resources

Our experiments focus on overall network resilience, on the variance of the individual risks of each agent in a supply network, and on the lowest of the individual risks of each agent in a supply network. We pay special attention to the structure with least available resources to each agent since we are interested in resilience in limited-resource commercial domains.

3.1 Model Validation

Our test results show a clear distinction between the most and least resilient network structures, which validates the supply network formation methods implemented by our model. The network structure that produced the least resilient

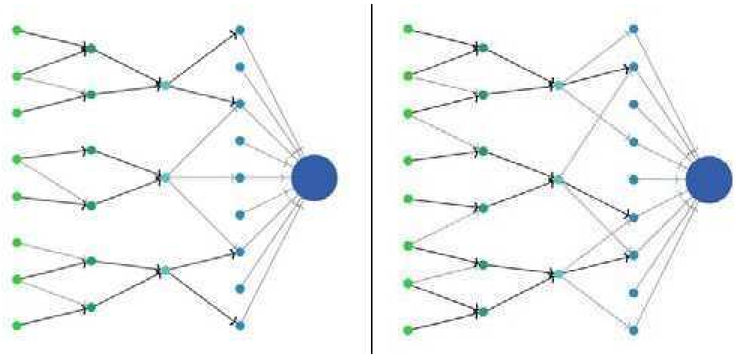


Fig. 5. Raw good price of 5 vs. 1000

supply network was that of a single, simple supply chain. This is true for all resources and raw good values. Since removing any of the agents from a simple supply chain results in complete disruption of the network, it is easy to understand why this is the least resilient network. Intuitively, the most resilient network is practically 15 separate supply chains directed to one consumer. If an agent is removed, only one of the supply chains is disrupted, leaving 14 other paths for goods to flow to the consumer.

Another result that validates our model with the results of [12] is shown in Figure 6. This chart also confirms that our model complies with the statement that redundancy increases network resilience [9]. This chart will be further discussed in section 3.2.

3.2 Analysis of Results

The first measure we look at is the variance in supply/demand ratio $\sigma^2(r_i(N))$ for a given network N. A high variance indicates that some agents are far safer from risk or far more at risk. High variance could leave a supply network more susceptible to planned attacks. The most imbalanced networks in our test results occurs when there is a large number of suppliers and only one of each other agent. In it, the agents that incur the least amount of risk are the suppliers, while removing any agent from the other tiers would result in a complete disruption. The least imbalanced network we found is also the most resilient network of 15 supply chains directed to one customer. It is the least imbalanced for the same reasons that give it high resilience: there are several supply chains, each of which has close to equal resilience.

We also learned that varying the number of relationships resources affects network resiliency. Figure 6 shows how as we increase the number of relationship resources the supply/demand ratio of the most vulnerable agent generally increases. Figure 7 shows how the variance of the supply/demand ratio $\sigma^2(r_i(N))$ decreases as we increase the amount of relationship resources. We note that a

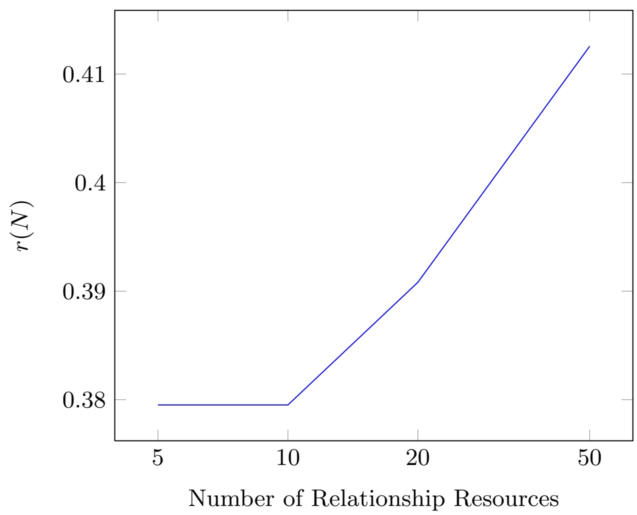


Fig. 6. Chart illustrating the average resiliences $r(N)$ of the networks formed using varying relationship resources.

slight anomaly occurs in Figure 6 when agents have 10 available resources. This could be due to the semi-random generation of the market when the agents are initialized. In general, however, these charts show how the resilience of a network increases as available relationship resources increases, as expected. This is because each agent has more opportunity to divide its product and create fewer dependencies in the supply network. It is also important to note that the most vulnerable node of a network with 5 relationship resources is only slightly lower than that of one with 50 relationship resources. Thus demonstrating that while having unlimited relationship resources may help slightly; it would probably not be worth the cost of having to manage all of the relationships.

Figure 8 shows that the number of agents involved in a single consumer based network with only 5 relationship resources available affects the network up to a certain point as well. Specifically, we see that the variance in supply/demand ratio is high for small numbers of agents (up to 10) but then drops after that and stays at nearly the same value for more than 10 agents. This happens because as more agents are involved in a single network, then more sub-networks are formed, thus increasing resilience. Another anomaly occurs when the total number of agents is 10. The variance is considerably higher. This is probably due to the type of structures that can be formed with 10 agents, or again, the semi-randomness of the agent initialization.

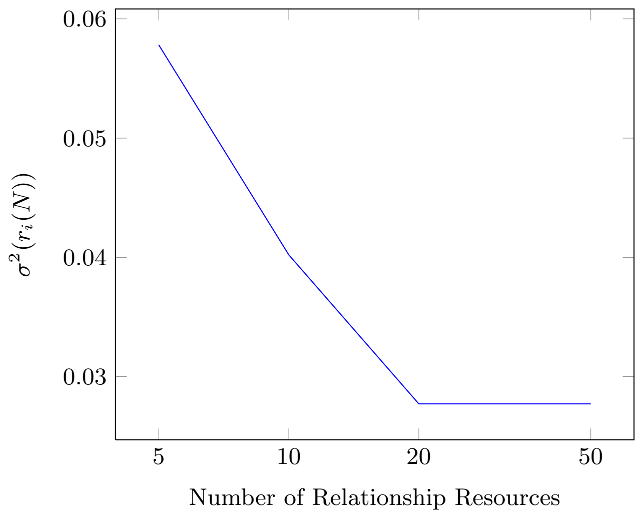


Fig. 7. Chart illustrating the average variances in the supply/demand ratio $\sigma^2(r_i(N))$ of the networks N formed as relationship resources vary.

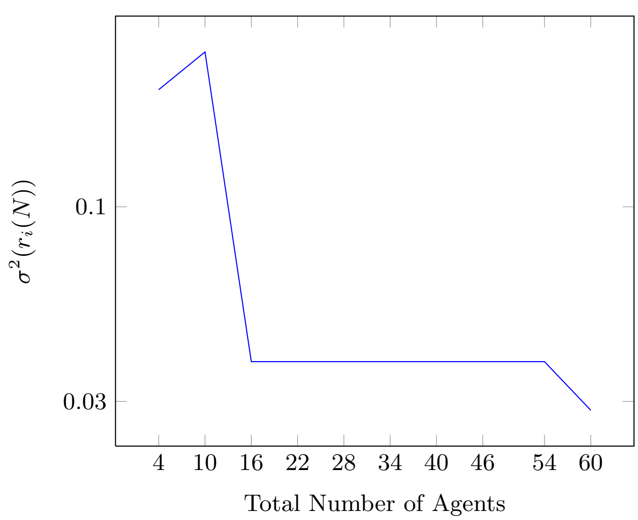


Fig. 8. Chart illustrating the average variances of the supply/demand ratio as the size of the network increases.

4 Conclusion and Future Work

The selfish-agent limited-resources supply network formation implemented by our model has produced results that are confirmed by previous research [12]. We have shown that the number of relationship resources available to business agents directly affects the resilience of the network. We have also shown that as relationship resources increase, the risk of the most vulnerable agents shows diminishing returns. Though the relationship resources in this experiment can be interpreted as any combination of time, money or any other overhead needed to maintain a contractual relationship, there is no doubt these additional relationship resources increase the cost to agents in a commercial supply network. Our results show that it is not cost effective to increase these relationship resources past a certain point (in Figure 6, this point is 20 relationship resources). These factors must be considered when constructing a commercial supply network.

The analysis of all the network topologies identify networks with low and high resilience. Although available resources seem to have the biggest impact on resilience, different structures also have a substantial effect on resilience. More specifically, the more opportunities agents have to form relationships, the more resilient the network. These observations will be useful in our future work.

While our current model does not fulfill the need of making supply networks more resilient, it identifies possible weaknesses in commercial supply networks and illustrates the reasons for these weaknesses. This is a significant step towards a solution. Our model also implements a new method for forming multiagent supply networks using current supply network management techniques [9][6][3]. The network formation and results from testing resilience will be improved and expanded in a number of different ways.

The greedy agent limited resource network formation method implemented in this paper will be used to create a dynamic market in which agents are actively trading goods. We will combine this model with our previous study of robustness, responsiveness and dynamism of supply networks during attacks [14]. This addition will give more intuition as to how the resilience of a particular network structure affects the way agents trade with each other and how trading agents will respond to the disruption of their supply network. The next step will be to offer certain incentives to agents and see if they cause the agents to form resilient supply networks.

Our research and results presented in this paper are a good foundation for developing methods that cause competing agents to be cooperative, but still selfish, as suggested by supply network management research [5].

References

1. James Andrews, Michael Benisch, Alberto Sardinha, and Norman Sadeh. Using information gain to analyze and fine tune the performance of supply chain trading agents. In *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*, 2009.

2. Gautam Basu, Mondher Ben-Hamida, Karen Butner, Eric Cope, Henry Dao, Lea Deleris, Jin Dong, Mary Helander, Kaan Katircioglu, Bonnie Ray, and Jason Torpy. Supply chain risk management: A delicate balancing act. Technical report, IBM Whitepaper, 2008.
3. Fu Chuan and Chen Yun. The research on customer lifetime value in multi-agent artificial market. In *Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, pages 489–494, 2008.
4. Trevor Hale and Christopher R. Moberg. Improving supply chain disaster preparedness: A decision process for secure site location. *International Journal of Physical Distribution and Logistics Management*, 35(3):195–207, 2005.
5. David J. Ketchen Jr. and G. Tomas M. Hult. Bridging organization theory and supply chain management: The case of best value supply chains. *Journal of Operations Management*, 25(2):573–580, 2007.
6. V. Kumar and Morris George. Measuring and maximizing customer equity: a critical analysis. *Journal of the Academy of Marketing Science*, 35(2):157–171, 2007.
7. Rolf A. E. Mueller, Doreen Buergelt, and Linda Seidel-lass. Supply chains and social network analysis. In *110th Seminar for the European Association of Agricultural Economomists*, 2008.
8. David Pardoe and Peter Stone. Adapting price predictions in tac scm. In *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*, 2009.
9. Yossi Sheffi. Building a resilient supply chain. *Harvard Business Review*, pages 1–4, October 2005.
10. Jayashankar M. Swaminathan, Stephen F. Smith, and Norman M. Sadeh. Modeling supply chain dynamics: A multiagent approach. *Decision Sciences*, 29(3):607–632, 2007.
11. David Teten. What distinguishes a strong tie from a weak tie? `http://www.thevirtualhandshake.com/blog/2004/05/25/what-distinguishes-a-strong-tie-from-a-weak-tie`, 2004.
12. Hari Prasad Thadakamalla, Usha Nandini Raghavan, Soundar Kumara, and Réka Albert. Survivability of multiagent-based supply networks: A topological perspective. *IEEE Intelligent Systems*, 19(5):24–31, 2004.
13. Jiang Tian and Huaglory Tianfield. Multi-agent based dynamic supply chain formation in semi-monopolized circumstance. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, pages 179–189. Springer, 2007.
14. José M. Vidal and Anand Nair. Examining the relationship between topology and performance of supply networks in the presence of random and targeted disruptions. In *38th Annual Meeting Decision Sciences Institute*, 2007.
15. Yunqi Zhang and Kate Larson. Reducing interaction costs for self-interested agents. In *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*, 2009.

A Resource Bounded Multi-Agent Approach to Deciding Advertisement Display*

Yue Zhang[1‡] and Nadeem Jamali[2]

[1] cfactor Works, 250 2nd Avenue S., Saskatoon, SK, S7K 1K9, Canada
[2] Agents Lab, Department of Computer Science, Univ. of Saskatchewan
110 Science Place Saskatoon, SK, S7N 5C9, Canada

Abstract. As popularity of online multimedia content grows, a number of models of advertising have emerged. Typically, brokers maintain the balance between content and advertising, their decisions informed by a significant body of work in psychology and marketing. However, existing approaches focus primarily on personalizing advertisements for viewer segments essentially identified by the content they view, with minimal decision-making capacity for individual viewers.

We take a resource bounded multi-agent view on the problem with an explicit treatment of viewer attention and its ownership. Particularly, we treat a multimedia consumer's attention space as a precious resource owned by the viewer. Viewers pay for the content they wish to view in dollars, as well as in terms of their attention. Advertisers pay for viewers' attention by subsidizing the cost of their content viewing.

Our approach specializes the CyberOrgs model for the attention resource; CyberOrgs encapsulate distributed computations and owned resources available for their execution. Particularly, advertisers can trade in viewers' attention just as viewers can buy multimedia content in a market of content. Key mechanisms are developed to give viewers flexible control over the display of advertisements in real time through personal agents. Pluggable customizable policies specify negotiation preferences of different parties, scalably automating typical negotiations and relieving the parties and their agents from actively engaging in explicit negotiations. This paper presents the rationale, design, implementation, and preliminary evaluation of our solution, FlexAdSense.

1 Introduction

According to a recent ad spending report from eMarketer [5], spending on Internet advertising will surpass $25 billion in 2009, and will reach up to $42 billion by 2013, as more money moves away from traditional media like television and newspapers. For reasons of the rapid expansion of video content availability on the web, analysts predict a rise in spending on video advertising on the web. TV ad spending is on the decline because features such as skip-over capability

*Support from NSERC, CFI, and Government of Saskatchewan is acknowledged.
‡This research was carried out as part of an M.Sc. thesis at the Agents Lab.

of recorders makes it difficult for advertisers to reach audiences. As traditional business models are threatened, advertisers and publishers are looking for alternatives. Brokers such as cable operators, satellite companies and multimedia web sites are the intermediaries between content publishers and content consumers.

Currently, there are primarily two models for multimedia brokers. One delivers paid high-quality content to viewers without any advertisement, and the other provides free content viewing but at the cost of embedded advertisements. Some approaches try to mix these two. In all these cases, the broker makes the decisions about whether or not to display advertisements, how many and which types of advertisements to display and when to display them. Viewers only have limited coarse-grained control. If the TV program or website contains too much unwanted commercial advertising, the viewer can switch to another channel or provider. In Figure 1, we compare these various approaches according to the granularity of control over multimedia content as well as advertising that they offer viewers. Although fine-grained control over content is often available, a similar control over advertising is not.

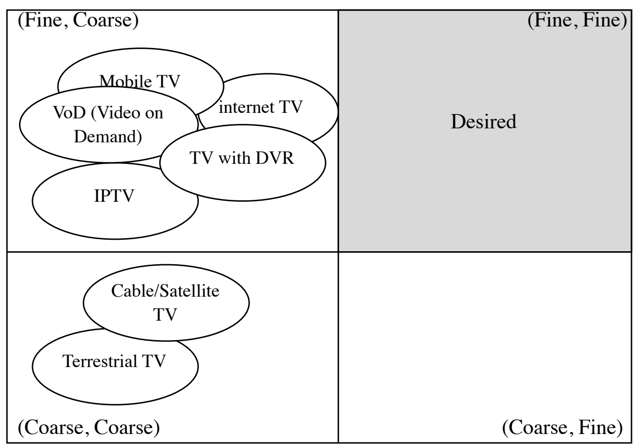


Fig. 1. Granularity of control for existing platforms. Values in the parentheses refer to the granularities of control over multimedia content and advertising respectively.

A number of advertising models have emerged for supporting digital content to consumers over the Internet; however, their primary focus is on personalizing advertisements for viewer segments, with minimal decision making capacity for individual viewers. Interactive TV mechanisms allow users to voluntarily interact. Currently, the interaction includes directly incorporated polls, questions, comments, and other forms of audience response back into the show. iMEDIA [15] business model applies technologies exploring viewers' interactive data and empowered viewers with control on their personal information. Yoon et al. [18] have proposed a system based on the TV-Anytime standard, which provides both media library services and targeted advertisement services. Unlike DVR, TV-Anytime enables remote recording on a personal or network DVR. In the web domain, personalization techniques are proposed for effective advertising. They can be roughly classified into four categories [14] according to the mathematical techniques adopted: *data mining* extracts consumers' behaviour patterns by observing and collecting interaction manners (e.g., [10], [14]); *decision trees* allow users to define target variables so as to generate rules for advertisement selection [11]; *linear programming* makes advertising decisions to maximize click-through rate and for satisfying advertisers' incentives in the meantime [12]; *nearest-neighbour collaborative filtering algorithms* recommend products based on predictions about consumers' preferences (e.g., [16], [13]). Viewers typically have limited control over the advertisements displayed.

The study of *Attention* has been pioneered by psychologist William James in 1890 [7]. He identified the two characteristics of attention – focalization and concentration – which continue to be studied today. Focalization means focusing on some interest or activity. Concentration means devoting mental effort to understand the information we receive. *Attention* was first introduced to Computer Science by Simon in 1971 [17]. He highlighted the imbalance between the "wealth of information" and the "poverty of attention", which has inspired a variety of personalization techniques and recommender systems. Bagozzi [3] proposed a model of complex circular exchange, in which attention is exchanged for entertainment or product information. Among more recent advances, Sander et al. [4] proposed a distributed Competitive Attention-space System (CASy), which treats a consumer's attention space as a scarce resource and makes use of adaptive software agents to allocate that resource in an electronic shopping mall. Suppliers compete with each other in an auction by bidding for the limited attention space of the consumer.

The remainder of the paper is structured as follows. Section 2 describes our approach for enabling fine-grained resource trade in viewers' attention space. In section 3, our prototype design and implementation are presented. In section 4, experimental results are presented. Finally, section 5 concludes our work.

2 Trade in Attention Resource

We take a resource ownership view on this problem. We view consumers' attention spaces – abstracted as a display screen for an engaged viewer – as precious resources owned by the viewers. Consumers pay for the content they wish to view in cash, as well as in terms of their attention. Specifically, advertisers may make partial payment for a viewer's content, in return for receiving the viewer's attention to their advertising. We build a market of viewers' attention spaces in which advertisers can trade, just as viewers can trade in a content market. We have developed key mechanisms to provide consumers with flexible realtime control over the advertising embedded in the multimedia content appearing on their screens. This approach relaxes the exclusivity of the relationship between advertisers and brokers, and empowers consumers to participate in decision making about advertising. Our approach is modular and allows reusability: specific policies needed for automated negotiations can be plugged-in.

In order to illustrate the features of our approach, let us consider the following scenarios involving a fictional user Jack:

Scenario 1: Jack turns on his TV after a long day of work. He browses through the 100 or so channels, and finally settles on his favorite detective show. During the show, advertisements interrupt his viewing at intervals. Jack wishes he could view his favorite content on his own schedule and wishes he could eliminate ads from this favourite content even if it requires paying extra money. Additionally, he wishes that he did not have to go through the ever larger number of channels; he wishes there were better searching and recommendation functionalities.

Scenario 2: Jack prefers high quality videos. However, these expensive tastes sometimes lead him to a tight budget at the end of the month. He wants to be able to set a monthly budget for himself, so that the system can automatically schedule content and advertisements for him: relatively more ads when the budget is tight and fewer ads when the budget is freer.

Scenario 3: Jack does not mind viewing advertisements if the ads fit his interests. Additionally, he dislikes ads that pop up and overlay at the bottom of the screen. He wishes to be able to select the types of ads.

Scenario 4: Jack has a 6-year-old daughter. He wants the screen free of adult-only content/ads, such as tobacco, alcohol and violence, when his daughter is watching television with him.

It turns out that none of the existing mechanisms satisfy the requirements identified in these scenarios. Our approach to the problem of making decisions about embedded advertisements is to enable fine-grained resource trade – in real time – between the owners of attention resource and the parties interested in acquiring them. Furthermore, because active participation in trade negotiation can place significant additional demands on viewer attention – taking away from the viewing experience – we have separated preferences of the different parties as pluggable policies, which – in *normal* situations – enable automated negotiations on behalf of the parties.

2.1 FlexAdSense

The name of our system, FlexAdSense, is inspired by Google's *AdSense*, in the sense that the system "senses" the most appropriate advertisements for viewers on the basis of preference data, both explicitly provided by viewers and implicitly extracted by exploring viewers' historical behaviours. "Flex" refers to the fine-grained flexible control available to viewers over the content and advertisements.

Consider the four parties involved in multimedia delivery: content publisher, content viewer (consumer), broker and advertiser. Each party owns certain resources and seeks to obtain certain resources. Multimedia publishers own multimedia content, which is a type of information resource. They wish to make profit by selling content resource using brokers' intermediary service. Viewers own cash and attention resource, and want to be entertained by viewing multimedia content. Brokers own delivery resources such as cables, network bandwidth and storage servers, and provide services such as the payment mechanisms and customer support modules. Advertisers own cash and are interested in viewers' attention. The resources of the greatest interest to us are *viewer attention* and *multimedia content*.

We treat the display screen as an abstraction of a viewer's attention space and view attention as a type of resource owned by the viewers, having both spatial extension and temporal duration. Specifically, we specialize the CyberOrgs model [6] for trade in these resources. Cyberorgs are distributed resource encapsulations which use *eCash* to buy and sell resources from/to each other. *eCash* is replaced by *real cash*. Viewer attention becomes an owned resource; anyone who wants this resource has to negotiate a contract with its owner. Contracts are negotiated between cyberorgs; these *contracts* stipulate the types, quantities and costs at which resources will be available to a buyer by a seller.

As illustrated in Figure 2, to be entertained, viewers pay for content resource in cash, as well as in terms of their attention. Brokers earn money by providing intermediary services: the trading and delivering platform. Publishers make profits by selling content. Screen real estate is owned by viewers, which means advertisers may not interrupt programming for displaying advertising without the viewer's permission. Advertisers have to purchase viewers' attention resources by proposing to partially pay for the viewers' content.

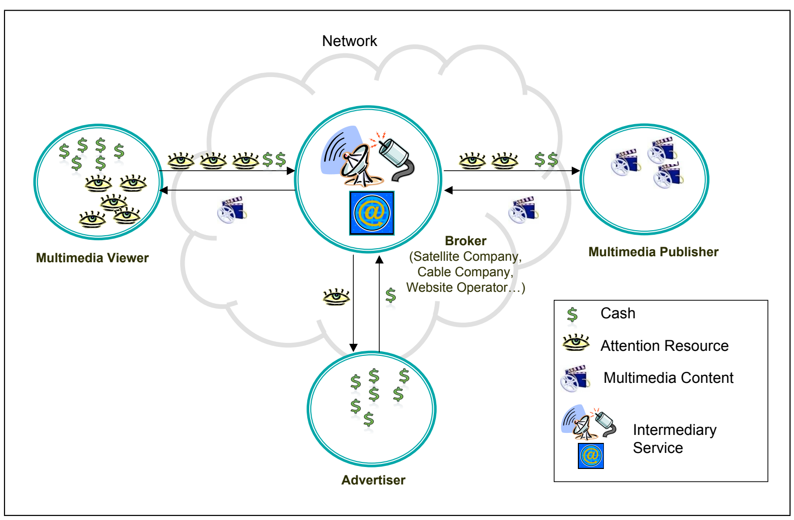


Fig. 2. Resource trade analysis.

As owners of their attention space, viewers are empowered to decide what is displayed in their attention space by managing access admission. For example, if a viewer does not desire paying any attention to advertisers, all attention resources go to the publisher and the viewer has a screen free of ads, at the cost of paying the full price for the viewed multimedia content. At the same time, because advertisements may also contain entertainment and useful information, such as upcoming promotions and new products, some viewers may not mind or may even prefer to view some ads. In such a case, viewers may agree to subtract the value received from the compensation expected for the ad.

Our approach allows viewers to choose advertisers as well as the schedule of access to their precious attention. The selected advertisers negotiate with viewers on how they may consume their attention resource, such as at what time to display which ads, occupying which parts of the screen and for how long.

Contracts FlexAdSense uses a number of parameters to describe availability of any type of resources; these descriptions serve as specifications of resources in contracts. For example, for multimedia content, the quality (DVD quality, fine quality or poor quality), playback capabilities (fast forward, fast reverse, or rewind), content category (action, strategy, sports or romance), etc. can be specified. Viewers can also negotiate the payment mechanism to be used, such as pay-per-view, periodic payments, or a flat monthly fee for unlimited viewing. For advertisements, viewers can specify the type (banner, video, text-in), category (beauty, apparel, travel, sports), and the specific format for each type of ad, such as font, position, duration and insertion time for text-in ads.

Once the multimedia content to be watched has been selected, a list of ads is generated. This process can either be carried out automatically after adoption of viewer preferences as policies, or manually by the viewers themselves; viewers have the ability to configure ads at a fine grain. Ads generated by this process become part of a contract.

2.2 Policies

Fine-grained negotiation can require significant user interactions, which can lead to unwanted additional demands on viewer attention. Therefore, we introduce policies for automated negotiations that can be used in predictable situations. Specifically, users may create their own policies, or adopt or customize policies available in a repository of typical policies. There are default policies in place for users who have not created customized ones. Policies are also composeable.

We have implemented three types of policies: preference policies, payment policies and privacy policies, which respectively specify policies for content/ad selection, payment and privacy.

Preference Policies Preference policies reduce explicit user interaction by specifying rules for filtering out unwanted or unrelated ad/content. Viewers, advertisers and publishers can all have preference policies. Viewers' preference policies are used to personalize the programming choices displayed on their screens – both content and ads. Viewer preference policies provide constraints which are used to create choices of display sequences with advertisements embedded in multimedia content streams. Policies can be as simple as "no advertisement" or as complex as "only sports ads of video clip type, inserted at the beginning of content." Recall scenario 1 and 3 involving our viewer Jack: he can now specify the multimedia content category of his interests, as well as the ad types he dislikes. Jack can specify constraints about price, category, date, type, language, etc., which are used for selecting ads. The price constraint, for example, can be set so that a price higher, lower or equal to an amount is accepted. The category and type constraints can be set to specify preference or otherwise for categories and types. The date constraint specifies before, on or after a given date. Similarly, for his content preference policy, Jack can specify the category, language, video quality, and price constraints, in much the same way as he did for ad preference policies.

Advertisers and publishers can also define preference policies, specifying ad/content attributes, by which they can target their audience and ease the ad/content creation process. For example, a lingerie advertiser not interested in wasting advertising money on men, can set the gender to female.[3] At the same time, these policies can be used to respect viewers' preferences, such as Jack's Scenario 4 preference to exclude adult–only content/ads in his daughter's presence. Similarly, local businesses can select to target viewers in specific geographic locations to avoid expenditure on people who are impossible to purchase their products. These policies can be used to simplify publishing of ads information: an advertiser/publisher who deals with only one type of product does not have to specify the common attribute shared by each product being advertised or multimedia content being sold. For instance, for a lingerie advertiser whose ads typically fall under the apparel category, the category can be automatically set to "apparel" by default, and exceptions can be identified as necessary.

Payment Policies FlexAdSense supports policies to support automated payment decisions by advertisers and viewers. An advertiser can specify payment policies by specifying pricing models or setting monthly or daily budgets. Similarly, a viewer can set long-term, budget-based or content-based policies which impact the amount of subsidy received from advertisers, and the amount payable for the multimedia content viewed.

- *Pricing Model*: Advertisers can specify static or dynamic pricing. Static pricing is independent of the context in which ads appear; dynamic prices depend on attributes defining the context. For example, an advertiser may want to vary the price to be paid to a viewer depending on the point of insertion in the content, as follows, to give preference to ads shown close to middle of the content being viewed. Price P paid by advertiser would be:

$$P = P_{max} - \left| \frac{(T_{insert} - D_{con}/2) * (P_{max} - P_{min})}{D_{con}} \right| \tag{1}$$

  where P_{max} and P_{min} are the maximum and minimum prices for the ad, T_insert is the point of insertion, and D_{con} is the duration of the content.

[3]This would require gender information about viewers, with privacy implications. Privacy policies are discussed later.

This linear model can also be used for other attributes: font size of text-in ads and image size of banner ads.

– *Budget*: Advertisers can set a daily or monthly budget for advertising spending. For example, an advertiser can set a daily budget of 20 dollars, so that when the spending reaches 20 dollars for the day, the ads are no longer selected for showing to viewers.

Viewers can choose the balance between viewing preferences and the price paid by specifying policies. Here are some examples:

– *Coverage Percentage*: A viewer can set a value p between 0 and 100 to specify the percentage of the content's price $P_{content}$ that they want to be paid by watching advertising; 100 would be equivalent to free programming, and 0 to no ads. Note that the percentage can also be automatically computed from a dollar amount for a selected content. Price P for watching content is:

$$P = P_{content} * (1 - p/100) \tag{2}$$

– *Monthly Budget*: Recall Scenario 2 in which Jack wishes to view high quality content within constraints of a monthly budget. This could be represented using the following policy for determining the current day's budget:

$$P_i = \frac{B_i}{f * (30 - i)} \tag{3}$$

where P_i is the budget for the i^{th} day of the month, B_i is the remaining balance in the monthly budget as of the i^{th} day, and f is the frequency (between 0 and 1, representing percentage of days) that the viewer watches content on television. Note that the policy tries to distribute budget evenly over the month.

– *Number of Ads Per Content*: A viewer can specify the number of ads to be viewed during the course of viewing specific content. If a viewer specifies the number of ads per content, n_{ads}, the price paid can be computed as follows:

$$P = P_{content} - \sum_{i=1}^{n_{ads}} P_{ad_i} \tag{4}$$

Privacy Policies Privacy issues have been addressed in the literature [9]. Unlike the way in which viewer information is typically guessed in existing systems based on viewer actions, in our approach, personal information is directly controlled by the viewers themselves. Viewers specify privacy policies determining how their data can be used. Particularly, viewers' personal information is treated as a type of resource with commercial value. In a manner similar to how owned attention resource could be traded, viewers can sell their personal information to advertisers in a market at a fine grain, rather than have them guess and then use it unacknowledged or arbitrarily. For example, viewers can set their information to be "totally private," "only used for ad targeting," "used for automatic subscription to newsletters," or "for sale to specific advertisers under particular conditions and terms," with a price specified for each type of access.

3 Implementation

The prototype is implemented over Actor Architecture (AA) [8], which implements primitive agents called actors [1]. The AA platform provides the asynchronous communication primitive **send** as well as synchronous communication primitive **call** (which is built upon asynchronous communication). Each type of agent can extend the base class *Actor* by defining methods which can be invoked as a result of receiving messages from other agents. Messages received by an actor are stored in its message queue until it is ready to process them. Actors are identified by globally unique names called Universal Actor Names (*UAN*), such as uan://128.233.104.144:3.

Each party is implemented as an agent. The broker agent acts as a server maintaining the database, which stores information including users' profiles, policies and information of all published content and ads. One special agent named *Directory Manager* (DM), offers a *Yellow Pages* service. Each agent is implemented with the functionality to negotiate resource trade with other parties at a fine grain. Several example policies are implemented and user interfaces are designed implementing key mechanisms as required for a functioning system.

3.1 Architecture

The structure of our system is as shown in Figure 3. Agents in FlexAdSense are designed according to their roles. Brokers are servers providing service, such as processing database queries and communicating with requesting users. Advertisers, publishers and viewers are represented by client agents responsible for interacting with users and sending users' requests to brokers, trading resources, displaying response results and so on.

The Directory Manager (DM) maintains the names of all brokers. When a new client agent is created, DM is responsible for arranging a rendezvous with a server, which subsequently takes care of requests from the agent. Because each broker maintains a replication of the database, DM also takes charge of synchronizing databases on brokers. Specifically, when the database in any broker is modified, DM has to instruct all brokers to update the database.

A Broker maintains the database which stores information about all users, including users' account information, viewers' profiles, various customized policies, advertisement information and content information. There can be more than one brokers. We use a load balancing based scheduling policy: the broker with the lightest load balance is selected to serve a new client agent. New brokers have to register with DM and replicate the database before going online.

Advertiser agents represent advertisers in interacting with brokers and DM, and manages their advertisements, policies and accounts, as well as investigates viewers' informations if applicable and necessary. Publisher agents represent publishers in managing their multimedia content resources and accounts. Viewer agents represent viewers in managing their attention space by searching for and selecting content resources, selling attention resources at a fine grain, as well as managing their profiles, policies and accounts.

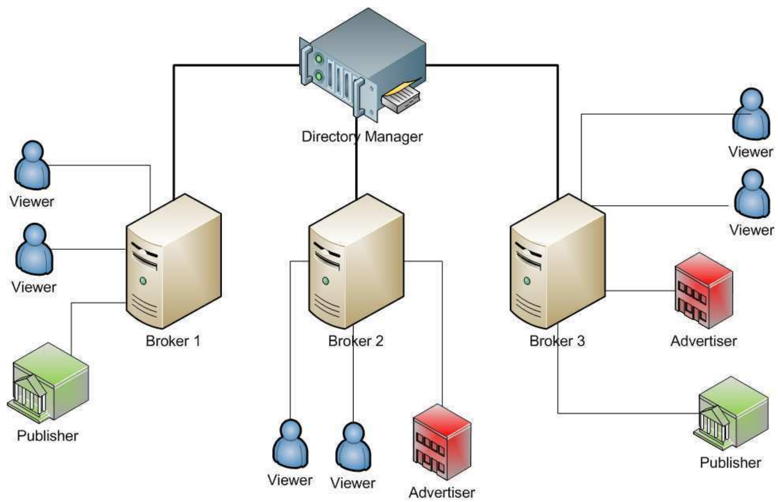


Fig. 3. The distributed structure of FlexAdSense.

3.2 Communication

There are two types of communications between agents: query requests and modification requests. Query requests refer to requests between users and their designated brokers. These requests can either be for searching or for subscribing to content/ad, or retrieving users' personal information. These types of requests are implemented by sending a message to the relevant broker. Messages can either be synchronous or asynchronous. The following examples show how a viewer's profile can be obtained using a synchronous message, and content can be searched for using an asynchronous message.

```
viewerForm = (ViewerForm)call(myBroker,  "searchViewerFormByID", loginID);

send(myBroker, "searchContentByName", conName, getActorName());
```

Note that *searchViewerFormByID* and *searchContentByName* are two methods defined in the broker actor class and are used to retrieve data from the database. The former requires a ViewerForm return value; the latter does not require a return value and sends results back by sending a message, along with its own name (*UAN*) obtained through a call to *getActorName()*.

Modification requests refer to requests that change the database. Because each broker maintains a copy of the database, modification requests trigger an update procedure on all brokers. These requests include requests to publish a new ad/content, to register a new user or to change preference information. For this type of requests, we use attribute–based communication [2] between agents, which means that an agent does not need to know the recipient agent's name, just some characteristics. To enable this, a Directory Manager maintains a public tuple space, which stores tuples with specific patterns, and offers the "deliverAll" service, using which a viewer agent can tell DM to send a request to all brokers.

If there exists a corresponding method at which the receiver matches the given arguments, the method is invoked. The following program illustrates registering of viewer's profile with brokers. Viewers do not have to know brokers' *UANs* or communicate with them individually. Instead, they simply send messages to "brokers."

```
//Extract all brokers in a tuple.
ActorTuple tuple = new ActorTuple(null, "broker");
send(anDM, "deliverAll", tuple, "addViewer", viewerform);
```

The user interface is encapsulated into an agent in a way similar to how a unix shell enables a user to act like a process in interacting with other processes. Except for the login user interface, all other GUI components are embedded inside user agents.

4 Evaluation

FlexAdSense can be evaluated along multiple dimensions. Key among these is the level of flexible fine-grained control afforded all involved parties. Section 4.1 provides this comparison, which is admittedly subjective. Two other interesting metrics of evaluation are server scalability and the additional demand on viewer attention resulting from interactions with the system. Among these, we have focused primarily on the former in Section 4.2; [19] briefly addresses the latter.

4.1 Granularity of Control

Existing multimedia delivery mechanisms afford viewers limited decision making capability over the advertisements they watch. Table 1 illustrates how FlexAdSense compares with existing approaches.

The flexibility of control available to advertisers and brokers/publishers is similarly high; more details can be found in [19].

4.2 Server Scalability

FlexAdSense offers mechanisms for viewers to be involved in making advertising decisions at a fine grain, which existing mechanisms do not. We experimentally evaluated these mechanisms for server scalability. Experiments were carried out using six Mac OS X Servers each running an actor platform of AA. Servers had 2×2.8 GHz Quad-Core Intel Xeon CPUs with 8 GB memory each; they were connected using a Gigabit network switch.

Our experiments used simulated load, with a number of viewers concurrently sending information requests at pre-set rates. These are the more frequent types of requests in the system; requests to modify schedules would be orders of magnitude less frequent, and thus have relatively insignificant impact on scalability. The rates at which simulated viewers sent requests (10 per sec) is also orders of magnitude higher than what would happen in practice: not all viewers would

Control over		Content Selection		Content Viewing		Control Granularity of content	Ad Selection	Ad Viewing	**Control Granularity of ads**
Mechanisms		# of Channels	Personalized Schedule	FF/FR/RW	Quality Control		Attributes Configuration	Skip-over ability	
TV	Terrestrial TV	Small	No	No	No	Coarse	**No**	No	**Coarse**
	Cable / Satellite	Large	No	No	No	Coarse	**No**	No	**Coarse**
	DVR	Large	Yes	Yes	No	Fine	**No**	Yes	**Coarse**
	interactive TV	Large	Yes	Depends	Yes	Fine	**No**	Depends	**Coarse**
Web	Video on Demand	Extra large	Yes	Depends	Yes	Fine	**No**	Depends	**Coarse**
	IPTV	Extra large	Yes	Depends	No	Fine	**No**	Depends	**Coarse**
Mobile Device	Mobile TV, etc.	Depends	Yes	Depends	No	Fine	**No**	Depends	**Coarse**
FlexAdSense		Extra large	Yes	Depends	Yes	Fine	**Yes**	Yes	**Fine**

Table 1. Comparison with existing mechanisms on control granularity over content and advertising.

be actively searching at the same time, nor would they search at as high a rate. In fact, the cumulative rate of generation of requests, not the number of viewers, turn out to be the significant determiner of performance. Consequently, our results about highly demanding viewers are equally applicable to orders of magnitude larger numbers (we expect 10^6) of actual viewers with typical demands.

Number of Viewers vs. Execution Time A set of experiments measured the total amount of wall-clock time required to complete serving requests as the number of viewers (distributed over three machines) grows from 1 to 3000. The requests were generated by viewers at the rate of 10 per second; the time required to process a request was set to 10 ms. All brokers were located on the same (multi-core) server. As Figure 4(a) shows, for the one broker case, as the number of viewers grows, the execution time increases significantly before becoming linear. However, there are orders of magnitude improvements when the load is divided between 2 or 3 brokers (note the logarithmic scale on y-axis). This shows that relatively few brokers executing in parallel can sufficiently improve performance.

Number of Brokers and Servers vs. Execution Time Another set of experiments examined the effect of number of brokers and servers on the total execution time. We simulated 1000 viewers; each viewer sent 100 requests in total; a request was sent every 100ms and took 100ms of computation. Brokers were evenly divided among up to 3 servers. As Figure 4(b) shows, when there is only one server, the system performs the best with close to 10 brokers, following which the execution time stays the same until it begins to grow linearly. This is expected because each servers had 8 cores (2 quad-core processors). The overhead of using multiple servers becomes evident when using two servers: although 16 brokers evenly divided between the two servers resulted in the best perfor-

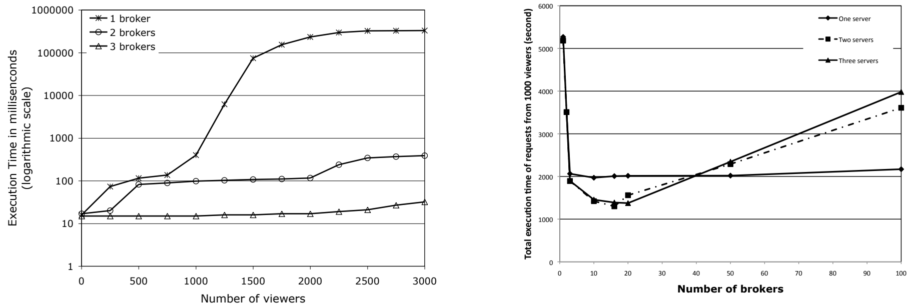


Fig. 4. a) Total execution time for different numbers of viewers; b) Total execution time for 1000 viewers with increasing number of brokers

mance, it was less than twice as good as for one server hosting half as many brokers. Significantly, when the number of servers was increased to 3, no perceivable improvement can be seen beyond the 2 server case, except that the best performance is shifted slightly to the right. On the one hand this suggests that a small number of servers is sufficient for obtaining the best performance; on the other hand, it is difficult to improve performance by simply adding additional distributed servers.

5 Conclusion and Future Work

We have presented an owned attention resource based approach to making decisions about advertising embedded in multimedia content, where viewers can sell their attention (and personal information) resource to advertisers. Different parties can install pluggable policies which enable their participation in negotiations without explicit interaction requiring attention resource. A distributed prototype has been designed and implemented. Preliminary evaluation shows greater flexibility than existing approaches at an acceptable cost. Experimental results indicate that the approach is scalable to large numbers of viewers and requests. An obvious opportunity for improving efficiency of our implementation lies in making updates to the replicated databases lazy.

An interesting direction of future exploration would be to extend the idea for dynamically changing groups of viewers. Imagine viewers entering a room with their personal agents carrying their preferences on personal mobile devices. Viewers' agents would then negotiate with each other before the group's preferences are negotiated with advertisers, not only to select content and ads agreeable to all parties, but also to allow fair distribution of any costs and leverage between the parties.

Finally, some broader questions require more thought. For example, advertisers may be interested in targeting new market segments; however, matchmaking based on stated preferences alone may exclude such matches.

References

1. G. Agha. *Actors: a model of concurrent computation in distributed systems*. MIT Press, Cambridge, MA, USA, 1986.
2. G. Agha and C. J. Callsen. Actorspace: An open distributed programming paradigm. In *Proceedings 4th ACM Conference on Principles and Practice of Parallel Programming, ACM SIGPLAN Notices*, pages 23–323, 1993.
3. R. P. Bagozzi. Marketing as exchange. *Journal of Marketing*, 39(4):32–39, 1975.
4. S. M. Bohte, E. Gerding, and H. L. Poutré. Competitive market-based allocation of consumer attention space. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 202–205, 2001.
5. eMarketer. emarketer's predictions for 2009. Technical report. Last retrieved: 11-June-2009.
6. N. Jamali and X. Zhao. A scalable approach to multi-agent resource acquisition and control. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)*, pages 868–875, Utrecht, Netherlands, July 2005. ACM Press.
7. W. James. *The Principles of Psychology*. Dover Publications, June 1950.
8. M.-W. Jang. *The Actor Architecture Manual*. Department of Computer Science. University of Illinois at Urbana-Champaign, 2004. Last retrieved: 5-August-2009.
9. A. Juels. Targeted advertising ... and privacy too. In *Proceedings of the Conference on Topics in Cryptology*, pages 408–424, London, UK, 2001. Springer-Verlag.
10. P. Kazienko and M. Adamski. Adrosa-adaptive personalization of web advertising. *Inf. Sci.*, 177(11):2269–2295, 2007.
11. H. Kiyomitsu, A. Takeuchi, and K. Tanaka. Web reconfiguration by spatio-temporal page personalization rules based on access histories. In *Applications and the Internet, 2001. Proceedings. 2001 Symposium on*, pages 75–82, 2001.
12. M. Langheinrich, A. Nakamura, N. Abe, T. Kamba, and Y. Koseki. Unintrusive customization techniques for web advertising. *Proceeding of the eighth international conference on World Wide Web*, pages 1259 – 1272, 1999.
13. M. R. McLaughlin and J. L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the International ACM SIGIR conference on Research and development in information retrieval*, pages 329–336, New York, NY, USA, 2004. ACM.
14. T. Oshiba, Y. Koike, M. Tabuchi, and T. Kamba. Personalized advertisement-duration control for streaming delivery. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 21–28, 2002.
15. K. Pramataris, D. Papakyriakopoulos, G. Lekakos, and N. Mylonopoulos. Personalized interactive tv advertising: The imedia business model. *Journal of Electronic Markets*, 11, 2001.
16. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.
17. H. A. Simon. Designing organizations for an information-rich world. *Computers, Communication, and the Public Interest*, 1971.
18. J. H. Yoon, H. K. Lee, J. M. Seok, H. Y. Kim, and H.-K. Lee. Tv-anytime based personalized bi-directional metadata service system. In *Consumer Electronics, 2006. ISCE '06.*, pages 1–6, 2006.
19. Y. Zhang. Screen real estate ownership based mechanism for negotiating advertisement display. Master's thesis, University of Saskatchewan, October 2009.