

Effective Variants of Max-Sum Algorithm to Radar Coordination and Scheduling

(Extended Abstract)

Yoonheui Kim
University of Massachusetts at
Amherst, MA 01003, USA
ykim@cs.umass.edu

Michael Krainin
University of Washington,
Seattle, WA 98195

Victor Lesser
University of Massachusetts at
Amherst, MA 01003, USA
lesser@cs.umass.edu

ABSTRACT

This work proposes new techniques for saving communication and computational resources when solving distributed constraint optimization problems in an environment where system hardware resources are clustered. Using a pre-computed policy and two phase propagation on Max-Sum algorithm, the system performance on Radar scheduling problem improves in terms of communication and computation.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Coherence and coordination

General Terms

Algorithms, Performance

Keywords

DCOP, Max-Sum, semi-centralized

1. INTRODUCTION

This paper focuses on utilizing semi-centralized hardware system structure to solve the agent coordination problem. It proposes modifications on message-passing algorithms in order to reduce required computation and communication resources. We consider the real-time sensor system NetRad for real-time hazardous weather phenomena detection [1]. In NetRad system, a collection of controllers responsible for multiple radars, radar coordination is essential for efficient resource utilization and accurate weather detection. We model the distributed scheduling problem as a constraint optimization problem and solve it approximately using the Max-Sum algorithm [2]. This work proposes two new extensions of the Max-Sum algorithm using a pre-computed policy and two-phase message propagation. The experimental results shows savings on 50% of communication and 5-30% of computational resources using these extensions.

2. RADAR SCHEDULING PROBLEM

Cite as: Effective Variants of Max-Sum Algorithm to Radar Coordination and Scheduling (Extended Abstract), Yoonheui Kim, Michael Krainin, Victor Lesser, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1153-1154. Copyright © 2011, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

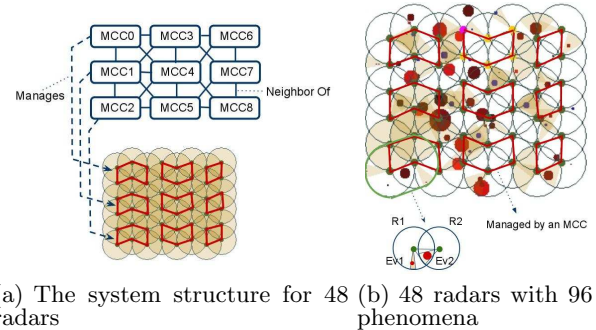


Figure 1: System structure for radars (a), Example configuration of radars with example scenario (b). All radar ranges and phenomena are assumed circular shaped. All phenomena locations and sizes are randomly selected. In (b), Radar 1 (R1) can choose to scan Event 1 (Ev1), Event 2 (Ev2) or to scan both depending on the utility. Scanning all phenomena in range with sufficient quality may not be possible given the time limit to scan.

2.1 Radar Scheduling Problem Formulation

The NetRad system simulator as in Figure 1 consists of controllers where each controller A_i controls and schedules a set of radars R_i . Given the real-time map of phenomena, each radar selects discretized scanning ranges by choosing a subset of phenomena in its range. For each phenomena p_j , the weight w_j is a constant determined by the requested user or the weather pattern. The utility (factorized local function) for each phenomenon j is defined as,

$$u_j : p_j \times \mathbf{r}^{p_j} \rightarrow c_j \quad (1)$$

where c_j denotes coverage of a scan within some range and \mathbf{r}^{p_j} denotes the scanning policy of radars which have p_j in range.

The goal of the system is to find a radar configuration r_1, \dots, r_n which maximize the sum U of the utilities for all phenomena and represented as,

$$U = \sum_j u_j(p_j, \mathbf{r}^{p_j}) \times w_j = \sum_j c_j \times w_j \quad (2)$$

Each radar can be thought as the variables with limited discrete domains and the local utility function u_j works as constraint that is involved with \mathbf{r}^{p_j} and we can solve the problem as distributed constraint optimization problem.

3. MODIFICATIONS ON MAX-SUM

3.1 Using Organization Structure: Max-Sum Alternating 2-level Hierarchy (MS2L)

We modify Max-Sum to have a two-level message propagation scheme in order to increase the algorithm efficiency in the context of clustered hardware resources. In the first propagation phase, we only send messages to nodes located within the same hardware resource. This is repeated for a number of message passing cycles. In the second phase, we send messages to nodes that are located in other hardware resources. These phases are then repeated until the termination criteria is reached. This modified Max-Sum, which we call MS2L, alternates between the cycle of global propagation cycle and local propagation so as to ensure that the utility values can also travel to other parts of the graph.

The 2-level propagation schedule

1. (Initialization) At any vertices, carry out the global flooding.

2. (Local flooding) Both variable and function nodes sends messages only to the neighbors within the same MCC. For each local neighbor, given the newest message on each edge, compute the message values for each local neighbor and send. Let the variable node's neighbors be N_i and the nodes in MCC k m_k . In function nodes, it sends the same message to a subset of neighbors $N_i \cap m_k$. In variable nodes, it computes the message using the previous messages from neighbors outside the MCC. At cycle t , the message from the variable to function node is,

$$q_{i \rightarrow j}^t(x_i) = \alpha_{ij} + \sum_{k \in N_i \cap m_k \setminus j} r_{k \rightarrow i}^t(x_i) + \sum_{k \in N_i \setminus m_k} r_{k \rightarrow i}^{t-1}(x_i)$$

3. (Global flooding) For all neighbors, do a regular message calculation using the newest message on each edge. Function nodes computes the messages at cycle t for all neighbors using messages at $t - 1$ for neighbors $N_i \setminus m_k$. The function node does not have updated messages for all neighbors due to local propagation in the previous cycle thus it combines previous messages from neighbors outside MCC.

$$r_{j \rightarrow i}^t(x_i) = \max_{\mathbf{x}_j \setminus i} [F_j(\mathbf{x}_j) + \sum_{k \in (N_j \cap m_k \setminus i)} q_{k \rightarrow j}^t(x_k) + \sum_{k \in (N_j \setminus (i \cup m_k))} q_{k \rightarrow j}^{t-1}(x_k)]$$

4. Repeat step 2 and 3.

3.2 Starting with Known Policy

In this section, we propose to construct better initial messages incorporating global information to further optimize the efficiency of the algorithm i.e. to start the algorithm with a policy for subgraph contained in the cluster processor.

The initial message in Max-Sum has the value assuming the best-case setting of other variables and only incorporates the local preferences. Given a known policy \hat{x} , we modify the algorithm for function nodes to send the following messages which does not involve maximization to the connected variable nodes. Function node j to variable node i :

$$F_j((\hat{\mathbf{x}}_j \setminus i) \cup x_i) \quad (3)$$

After receiving these messages, if a variable node were to take on a value, it would be:

$$\tilde{x}_i = \arg \max_{x_i} \sum_{j \in N_i} F_j((\hat{\mathbf{x}}_j \setminus i) \cup x_i) \quad (4)$$

3.2.1 Using the Structure for Policy Generation

Additionally we provide a scheme which computes a policy which can be used as in Section 3.2. Instead of generating a policy for the whole problem, we tried to compute the locally optimal policy for subproblems associated with each MCC. We break the full factor graph into factor subgraphs for each MCC that contains only the radars and phenomena in each MCC and are smaller than the original factor graph. In order to accomplish this, we assign each phenomenon to one MCC to avoid redundant utilities for shared phenomena in computing the initial policy. Consequently, the domain of variable nodes and parameter values in the cost function at the function nodes are smaller than the original problem. Starting with the generated policy as prior information, Max-Sum starts with knowledge on local functions.

4. PERFORMANCE OF MAX-SUM IN A TWO-LEVEL HIERARCHY

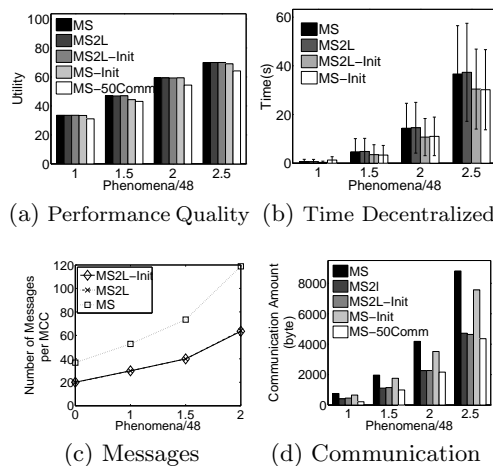


Figure 2: Performance of MS2L

We experimented with MS2L as in Section 3.1 with increasing number of phenomena and also MS2L-Init with Init-MS policy replacing the first 2 cycles for generating the policy. Detailed results and description can be found in [3].

5. REFERENCES

- [1] M. Zink et al. Meteorological Command and Control: An End-to-end Architecture for a Hazardous Weather Detection Sensor Network. In *Proc. of the ACM Workshop on End-to-End, Sense-and-Respond Systems, Applications, and Services*, pages 37–42, 2005.
- [2] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *AAMAS*, pages 639–646, 2008.
- [3] Yoonheui Kim et al. Effective variants of max-sum algorithm to radar coordination and scheduling. Technical Report UM-CS-2011-007, University of Massachusetts, Amherst, February 2011.