A Decomposition Theorem for Task Systems and Bounds for Randomized Server Problems *

Avrim Blum † Howard Karloff ‡ Yuval Rabani §

Michael Saks ¶

November 8, 2000

Abstract

A lower bound of $\Omega\left(\sqrt{\log k / \log\log k}\right)$ is proved for the competitive ratio of randomized algorithms for the k-server problem against an oblivious adversary. The bound holds for arbitrary metric spaces (having at least $k+1$ points) and provides a new lower bound for the metrical task system problem as well. This improves the previous best lower bound of $\Omega(\log\log k)$ for arbitrary metric spaces [KRR] and more closely approaches the conjectured lower bound of $\Omega(\log k)$. For the server problem on $k+1$ equally-spaced points on a line, which corresponds to a natural motion-planning problem, a lower bound of $\Omega\left(\frac{\log k}{\log\log k}\right)$ is obtained.

The results are deduced from a general decomposition theorem for a simpler version of both the k-server and the metrical task system problems, called the "pursuit-evasion game." It is shown that if a metric space $\mathcal{M}$ can be decomposed into two spaces $\mathcal{M}_L$ and $\mathcal{M}_R$ such that the distance between them

*A preliminary version of this paper in the Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science, 1992.

†School of Computer Science, Carnegie Mellon University. E-mail: avrim@theory.cs.cmu.edu. This work was supported in part by an NSF Postdoctoral Fellowship.

‡College of Computing, Georgia Institute of Technology. E-mail: howard@cc.gatech.edu. This author was supported in part by NSF grant CCR 9107349 and by DIMACS.

§Work done while a graduate student at Tel Aviv University Department of Computer Science. Part of this work was done while visiting DIMACS. Present address: Computer Science Department, Technion — IIT, Haifa 32000, Israel.

¶Dept. of Mathematics, Rutgers University and Dept. of Computer Science and Engineering, UCSD. E-mail: saks@math.rutgers.edu. This work was supported in part by NSF grant CCR89-11388 and AFOSR grants 89-0512 and 90-0008.

is sufficiently large compared to their diameter, then the competitive ratio for this game on $\mathcal{M}$ can be expressed nearly exactly in terms of the ratios on each of the two subspaces. This yields a divide-and-conquer approach to bounding the competitive ratio of a space.

1 Introduction and Main Results

On-line computation is a setting in which randomization has been shown to have a provable advantage over determinism (see, e.g., [BE]). An on-line computation problem typically involves responding to a sequence of requests in order to minimize some cost function. The standard measure of success is the *competitive ratio* [ST, KMRS] which is, roughly, the maximum over all request sequences of the ratio of the cost charged to the algorithm on a request sequence, to the optimal offline cost of servicing that sequence. It is useful and customary to view the request sequence as chosen by an adversary who knows the algorithm being used and seeks to force this ratio to be large. Against a deterministic algorithm, the adversary can completely predict the responses of the algorithm and this gives it great power for forcing the algorithm to perform badly. Against a randomized algorithm the adversary knows the algorithm, but not the random choices of the algorithm. Intuitively, this can be interpreted by saying that after each successive request, the adversary "knows" only a probability distribution over states of the algorithm rather than the precise state. This restriction on the adversary provides the potential advantage of randomization. (Note that here and throughout this paper we are discussing a version of the adversary known as an "oblivious" adversary [BBKTW], [RS]. There are other, more powerful, adversaries that are less vulnerable to randomization).

In the well known k-server problem, an algorithm controls k servers, each of which occupies some point in a metric space $\mathcal{M}$. At each time step the algorithm is given a *request*, which is a point in $\mathcal{M}$, and must serve it by moving a server to that point if none is there already. The algorithm is charged a cost equal to the total distance moved. It has been shown that for any metric space having at least $k+1$ points no deterministic online algorithm can achieve a competitive ratio less than k [MMS] (note that the problem is nontrivial only if there are at least $k+1$ points). The well-known k-server conjecture [MMS] says that for any metric space, there is a deterministic online algorithm that can achieve a competitive ratio of k. In other words, if we define the competitive ratio of a metric space to be the minimum ratio achievable by any algorithm, then the conjecture is that for the k-server problem, the deterministic competitive ratio of any metric space on at least $k+1$ points is exactly k. A breakthrough result [KP] provided a deterministic algorithm with competitive ratio $2k-1$, improving on the previous exponential upper bounds ([FRR],[Gro]).

The power of randomization in this setting was first demonstrated for the *uniform* metric space on $k+1$ points, $\mathcal{U}(k+1)$, in which all pairs of distinct points are equidistant. For this space there is an $O(\log k)$-competitive algorithm, and indeed this is a lower bound:

Theorem 1.1 ([FKLMSY],[MS],[BLS]) *The k-server problem for $\mathcal{U}(k+1)$ has randomized competitive ratio exactly $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k} \sim \ln k$.*

In fact, these bounds hold for uniform metric spaces of any size greater than k. Various people have speculated on the following conjecture:

Conjecture 1.1 *For any k and any metric space $\mathcal{M}$ on more than k points, the randomized competitive ratio of the k-server problem on $\mathcal{M}$ is $\Theta(\log k)$.*

Unlike the deterministic case where the lower bound is relatively easy and only the upper bound seems difficult, neither bound has been proved in the randomized case. For the lower bound, the previous best result is the following:

Theorem 1.2 ([KRR]) *Let k be a positive integer and $\mathcal{M}$ be a metric space with at least $k+1$ points. Then the k-server problem on $\mathcal{M}$ has randomized competitive ratio $\Omega(\min\{\log k, \log\log|\mathcal{M}|\})$.*

If $\mathcal{M}$ is sufficiently large (exponential in k) then the lower bound in Conjecture 1.1 holds. For arbitrary spaces, in particular those whose size is polynomial in k, the lower bound is $\Omega(\log\log k)$. One of the main results of this paper is to improve this lower bound:

Theorem 1.3 *For any metric space $\mathcal{M}$ with at least $k+1$ points, the k-server problem on $\mathcal{M}$ has randomized competitive ratio $\Omega(\sqrt{\frac{\log k}{\log\log k}})$.*

The competitive ratio of the k-server problem for $\mathcal{M}$ is at least as large as the ratio for a subspace of $\mathcal{M}$, as the adversary can restrict its requests to that subspace. Thus, a lower bound on the competitive ratio of the k-server problem for the case that the space has exactly $k+1$ points implies the same lower bound for every metric space. One way to view this special case is to think of the algorithm as occupying a single point of the space (corresponding to the unique location where there is no server) and to think of the adversary as probing points of this space. When the adversary probes the point on which the algorithm stands, the algorithm must move to a different location. We call this the *pursuit-evasion (PE) game* and call the adversary the Pursuer and the algorithm the Evader. This paper is about this game. It should be noted that the pursuit-evasion game bears a superficial resemblance to the cat-and-mouse game of [CDRS], but that game models the case of randomized algorithms against a more powerful adversary.

The pursuit-evasion game also models a problem in robotics. Imagine a robot walking down a long hallway of some width n (e.g., if $n = 3$ then the robot may walk either down the left side, the center, or the right side of the hallway). The hallway contains rectangular obstacles, and when the robot meets an obstacle, it must go left or right around it. Any algorithm at all must travel the length of the hallway, so we will not charge for that. Instead we look at the left/right motion of the robot and compare it to the least possible left/right motion by an algorithm that knew the placement of the obstacles in advance. If the hallway has width n, then this is the pursuit-evasion game for the metric space of n equally-spaced points on the real line, a metric space we call $\mathcal{L}(n)$. The above lower bound of $\Omega(\sqrt{\frac{\log(n-1)}{\log\log(n-1)}})$ applies, of course, but for this case we have a better lower bound.

Theorem 1.4 *The pursuit-evasion game on $\mathcal{L}(n)$ has randomized competitive ratio $\Omega(\frac{\log n}{\log\log n})$. Thus, if $n > k$ the k-server problem on $\mathcal{L}(n)$ has competitive ratio $\Omega(\frac{\log k}{\log\log k})$.*

This nearly matches the conjectured bounds.

For general spaces $\mathcal{M}$, the special case of Conjecture 1.1 with $k = |\mathcal{M}| - 1$ can be stated as:

Conjecture 1.2 *For any metric space $\mathcal{M}$ on n points, the randomized competitive ratio of the pursuit-evasion game on $\mathcal{M}$ is $\Theta(\log n)$.*

As noted, the lower bounds of Conjecture 1.2 and Conjecture 1.1 are equivalent. On the other hand, an upper bound for the pursuit-evasion game does not have immediate application to the upper bound conjecture for the general k-server problem. (In fact, there is some evidence that for $k = 2$ the competitive ratio might be worse for metric spaces with more than 3 points, see [LR].) In any case, we believe that a solution to the pursuit-evasion game would be a major step towards the solution of the more general problem and would also be interesting in its own right. (Since the appearance of a preliminary version of the present work, significant progress on this problem has been made. Bartal et al. [BBBT] gave a polylog(n) algorithm for the pursuit-evasion game, and more generally for any metrical task system.)

Previously, Conjecture 1.2 was known to be true only for the case of uniform (or nearly uniform) spaces mentioned earlier. Here, we establish Conjecture 1.2 for a dual situation. If $C > 1$, a metric space is *C-unbalanced* if for any three distinct points, the ratio of the largest distance to the smallest nonzero distance is at least C. For example, the metric space consisting of 4 points in a rectangle with side lengths 1 and C is C-unbalanced.

Theorem 1.5 *There is a polynomial $p(n)$ such that for all n, the pursuit-evasion* *problem on any $p(n)$-unbalanced metric space with n points has randomized competitive ratio between $\ln n$ and $3 \ln n$.*

It is worth mentioning that bounds on the competitive ratio of the pursuit-evasion game carry over to the *task system* model of [BLS]. In particular, Theorems 1.3, 1.4 and 1.5 hold if we replace "the pursuit-evasion game on $\mathcal{M}$" by "the task system on $\mathcal{M}$."

1.1 Overview of the method

Theorems 1.3, 1.4 and 1.5 are proved as a consequence of a *decomposition theorem* for the competitive ratio of the pursuit-evasion game. (Henceforth, when we say "competitive ratio" we will mean the "randomized competitive ratio.") The theorem concerns metric spaces that can be split into two subspaces, where the diameter of each subspaces is small relative to the overall diameter, and it asserts that the competitive ratio of the pursuit-evasion game on the whole space can be expressed almost exactly in terms of the competitive ratios of the games of the game on the two subspaces.

To state this result, we need some notation. For a finite metric space $\mathcal{M}$, $\delta(\mathcal{M})$ denotes its diameter and $\lambda(\mathcal{M})$ denotes the competitive ratio of the associated pursuit-evasion game. We use the convention that the competitive ratio of a one point space is 0. A subspace $\mathcal{N}$ of $\mathcal{M}$ is said to be γ-*small* in $\mathcal{M}$, for $\gamma < 1$ if $\delta(\mathcal{N}) < \gamma\delta(\mathcal{M})$. We say that $\mathcal{M}$ is *bipartite* if it can be split into two subspaces $\mathcal{M}_L$ and $\mathcal{M}_R$ (called, respectively, the left space and the right space) such that the diameters of $\mathcal{M}_L$ and $\mathcal{M}_R$ are each less than $\delta(\mathcal{M})/2$. We call $(\mathcal{M}_L, \mathcal{M}_R)$ a *bipartition* of $\mathcal{M}$. It is easy to see that if a bipartition exists, then it is unique. If $\mathcal{M}$ is bipartite, then we say that it is γ-*bipartite*, for $\gamma < 1/2$ if $\delta(\mathcal{M}_L)$ and $\delta(\mathcal{M}_R)$ are each γ-small in $\mathcal{M}$, and we call $(\mathcal{M}_L, \mathcal{M}_R)$ a γ-*bipartition* of $\mathcal{M}$.

The precise statement of our result is a bit long and is given as Theorem 1.7. We begin with a simpler version, useful when $\mathcal{M}_L$ and $\mathcal{M}_R$ have nearly the same competitive ratio.

Theorem 1.6 *For any $\epsilon > 0$, there exists a polynomial p such that the following holds. Let $\mathcal{M}$ be a metric space with bipartition $(\mathcal{M}_L, \mathcal{M}_R)$. If $\lambda_{\min}$ and $\lambda_{\max}$ are real numbers such that $\lambda_{\min} \leq \lambda(\mathcal{M}_R), \lambda(\mathcal{M}_L) \leq \lambda_{\max}$ and each part is $\frac{1}{p(\lambda_{\max})}$-small in $\mathcal{M}$ then*

$$\lambda_{\min} + 1 - \epsilon \leq \lambda(\mathcal{M}) \leq \lambda_{\max} + 1 + \epsilon.$$

In words, if $\mathcal{M}_R$ and $\mathcal{M}_L$ both have competitive ratio close to λ then the competitive ratio of $\mathcal{M}$ is close to $1+\lambda$.

The decomposition theorem can be used to estimate the competitive ratio of the pursuit-evasion game on a space by partitioning it into smaller spaces and applying induction. The lower bound of the theorem can be applied to non-bipartite spaces by applying it to a bipartite subspace $\mathcal{N}$ and using $\lambda(\mathcal{N}) \leq \lambda(\mathcal{M})$. For example, Theorem 1.4 can be derived as follows. Let $\epsilon = 1/2$ and let $p()$ be the increasing polynomial whose existence is given by Theorem 1.6. Let t be the greatest integer such that $\lceil p(\log n) \rceil^t \leq n$ and let $n' = \lceil p(\log n) \rceil^t$. Note that $t = \Omega(\frac{\log n}{\log \log n})$. Let $j = n'/\lceil p(\log n) \rceil = \lceil p(\log n) \rceil^{t-1}$ and let $\mathcal{M}_L$ and $\mathcal{M}_R$ be the leftmost j points and rightmost j points of $\mathcal{M} = \mathcal{L}(n')$ respectively. By choice of j, either $\lambda(\mathcal{M}_L) > \log n$ and we are done, or else the condition on δ of Theorem 1.6 is satisfied. So, $\lambda(\mathcal{L}(n)) \geq \lambda(\mathcal{L}(n')) \geq \lambda(\mathcal{M}_L \cup \mathcal{M}_R) \geq \lambda(\mathcal{M}_L) + 1/2$. We can continue on $\mathcal{M}_L$ letting $j' = j/\lceil p(\log n) \rceil$ and so forth until after t steps we have run out of points. The competitive ratio of $\mathcal{L}(n)$ is thus at least $t/2$, which is $\Omega(\frac{\log n}{\log \log n})$.

The full decomposition theorem provides sharp bounds on $\lambda(\mathcal{M})$ even if $\mathcal{M}_L$ and $\mathcal{M}_R$ have different competitive ratios. Define the function $Z(x)$ on non-negative reals by $Z(x) = x/(e^x - 1)$ if $x > 0$ and $Z(0) = 1$. The theorem says that $\lambda(\mathcal{M})$ is well approximated by $\max\{\lambda(\mathcal{M}_L), \lambda(\mathcal{M}_R)\} + Z(|\lambda(\mathcal{M}_L) - \lambda(\mathcal{M}_R)|)$.

Theorem 1.7 *Let $\mathcal{M}$ be a metric space with at least three points, having bipartition $(\mathcal{M}_R, \mathcal{M}_L)$. Let $\alpha_R, \alpha_L \geq 0$ and set $\alpha_{\max} = \max\{\alpha_R, \alpha_L\}$ and $\alpha_{\text{diff}} = |\alpha_R - \alpha_L|$. Let $\delta = \delta(\mathcal{M})$ and $\delta_{\max} = \max\{\delta(\mathcal{M}_L), \delta(\mathcal{M}_R)\}$. Suppose that $\alpha_{\max} \geq 1$ and that each part is $1/\max\{324\alpha_{\max}, \alpha_{\max}^3\}$-small in $\mathcal{M}$. Then*

1. *if $\alpha_L \geq \lambda(\mathcal{M}_L)$ and $\alpha_R \geq \lambda(\mathcal{M}_R)$, then $\lambda \leq \alpha_{\max} + Z(\alpha_{\text{diff}})(1+\zeta)$, and*

2. *if $\alpha_L \leq \lambda(\mathcal{M}_L)$ and $\alpha_R \leq \lambda(\mathcal{M}_R)$, then $\lambda \geq \alpha_{\max} + Z(\alpha_{\text{diff}})(1-\zeta)$,*

where $\zeta = 23e^{\alpha_{\text{diff}}}\sqrt{\frac{\delta_{\max}}{\delta}\alpha_{\max}^3}$.

The proof of Theorem 1.7 consists of two parts. We introduce and completely analyze a new game, called the *Walker-Jumper* game, which abstracts the essential elements of the analysis of a decomposed problem. Then we formally demonstrate that the competitive ratio of a decomposed problem can be tightly bounded by the competitive ratio of an associated Walker-Jumper game.

Theorem 1.6 follows easily by applying the first part of Theorem 1.7 with $\alpha_L = \alpha_R = \lambda_{\max}$ and the second part of the Theorem with $\alpha_L = \alpha_R = \lambda_{\min}$.

Theorem 1.7 is combined with a Ramsey-type theorem for metric spaces to prove Theorem 1.3. The cases of Theorem 1.7 needed are the case where $\mathcal{M}_L$ and $\mathcal{M}_R$ have nearly the same competitive ratio, and the "highly unbalanced" case where $\mathcal{M}_L$ is large and $\mathcal{M}_R$ is a single point. The Ramsey type theorem, which can be viewed as an extension of a theorem from [KRR], says roughly that any metric space of n points must contain at least one of the following *three* objects: (A) a roughly uniform space of around $2^{\sqrt{\log n/\log\log n}}$ points, (B) two highly separated spaces with small diameter, each having around $n/2^{\sqrt{\log n \log\log n}}$ points, or (C) one point very far from a small diameter subspace containing nearly all the rest (around $n - n/2^{\sqrt{\log n/\log\log n}}$ points). (For other Ramsey-like theorems for metric spaces, see [Mat].)

In Section 3, we present an informal discussion of the proof to motivate the connection with the Walker-Jumper game. In Section 4 we define the Walker-Jumper game and state a theorem which gives its exact competitive ratio, and describe and prove the optimal strategies for each of the two players. The precise statement and proof of the lemma connecting the Walker-Jumper game to the decomposition theorem is given in Section 5. The applications of the decomposition theorem needed to prove Theorems 1.3 and 1.5 are given in Section 6 which can be read independently of the previous ones. It requires only the statement of the main decomposition theorem.

Section 5 is long and technical, although the underlying idea as sketched in Section 3 is fairly intuitive. Two technical lemmas stated in that section, Lemmas 5.2 and 5.3, provide quantitative bounds on the additive constant that occurs in the definition of competitive ratio, and may be of independent interest. The proofs of these lemmas are deferred to the last section.

2 Online Games: Definitions and Preliminary Results

2.1 Notation

As usual, $\mathbf{R}$ and N denote, respectively, the sets of real numbers and the set of nonnegative integers. The set $\{x \in \mathbf{R} : x \geq 0\} \cup \{\infty\}$ is denoted $\mathbf{R}_\infty$.

We will need the following notation for sequences. Let X be a set. We denote by X^n the set of sequences consisting of n terms from X and $X^* = \bigcup_{n\geq 1} X^n$. An element of X^* is denoted in boldface as $\mathbf{x} = (x_1, x_2, \ldots, x_t)$; sequences are indexed from 1 unless otherwise noted. The number t of terms is called the *term length* of

x, and is denoted $|\mathbf{x}|$. If $j \leq t$, then we use $\mathbf{x}^j$ to denote the sequence consisting of the first j terms of $\mathbf{x}$. If $\mathbf{x}$ and $\mathbf{y}$ are sequences such that $\mathbf{x}^j = \mathbf{y}$ for some integer j then we say that $\mathbf{y}$ is a *prefix* of $\mathbf{x}$ and that $\mathbf{x}$ is an *extension* of $\mathbf{y}$. If $\mathbf{x}$ and $\mathbf{y}$ are sequences, then $\mathbf{xy}$ denotes their concatenation.

If $\mathbf{x}$ is a sequence of real numbers of term length n, then $\mathbf{\Delta x}$ denotes the sequence of differences: $\Delta x_1 = x_1$ and for $2 \leq j \leq n$, $\Delta x_j = x_j - x_{j-1}$.

A metric space $\mathcal{M}$ consists of a set of points P and a symmetric nonnegative valued distance function defined on $P \times P$ that satisfies the triangle inequality and is zero only on the diagonal. We abuse notation and use $\mathcal{M}$ to denote both the metric space and the underlying set of points P. The associated metric is denoted by $d = d_{\mathcal{M}}$. $\mathcal{M}$ is assumed to be a finite set unless otherwise noted. The diameter of the space, $\delta = \delta(\mathcal{M})$, is the maximum distance between any pair of points.

A sequence of points in a metric space is referred to as a *walk* in the space. The domain of the distance function can be viewed as the set of walks of term length 2. We extend the domain of d to the set of all walks by defining $d(\mathbf{x})$ to be the sum of the distances between successive pairs of points. We refer to this as the *metric length* of the walk.

2.2 Two-Player Zero-Sum Games

The online algorithmic problems considered in this paper can be viewed as two-person zero-sum games. We recall some basic definitions and results about such games. A two-player zero-sum game G with players MAX and MIN is a triple $(S_{\mathrm{MAX}}, S_{\mathrm{MIN}}, C)$ where S_{MAX} and S_{MIN} are sets, and a *cost* function $C = C^G$, where $C : S_{\mathrm{MIN}} \times S_{\mathrm{MAX}} \to \mathbf{R}_\infty$. An element of S_{MAX} (resp. S_{MIN}) is called a *pure strategy* of MAX (resp. MIN). The game is *finite* if S_{MAX} and S_{MIN} are finite sets. It will be convenient to use an asymmetric notation: $C_q(r)$ denotes the value of this function for strategies $q \in S_{\mathrm{MIN}}$ and $r \in S_{\mathrm{MAX}}$. The value $C_q(r)$ is intuitively the cost to MIN given these two strategies.

In a *randomized instance* of the game, each player selects a probability distribution over its strategy set; such a distribution is called a *mixed strategy*. The set of mixed strategies for player X is denoted $\tilde{S_X}$. Similarly, we denote a mixed strategy by a letter with a $\sim$ over it.

The cost of two mixed strategies $\tilde{q} \in \tilde{S}_{\mathrm{MIN}}$ and $\tilde{r} \in \tilde{S}_{\mathrm{MAX}}$, denoted $C_{\tilde{q}}(\tilde{r})$, is the expected value of $C_q(r)$ with respect to the product distribution of the two strategies.

The *value* of a mixed strategy $\tilde{r} \in \tilde{S}_{\text{MAX}}$, denoted $V_{\text{MAX}}(\tilde{r})$, is the infimum of $C_{\tilde{q}'}(\tilde{r})$ over all $\tilde{q}' \in \tilde{S}_{\text{MIN}}$ (which may be 0.) Similarly, for $\tilde{q} \in \tilde{S}_{\text{MIN}}$, $V_{\text{MIN}}(\tilde{q})$ is the supremum of $C_{\tilde{q}}(\tilde{r}')$ over all $\tilde{r}' \in \tilde{S}_{\text{MAX}}$ (which may be $+\infty$.)

It is well known (and easy to show) that the value of a mixed strategy is determined by its cost with respect to pure strategies for the other player:

Lemma 2.1 *Let G be a two-player zero-sum game.*

1. *For any mixed strategy $\tilde{r}$ for MAX, $V_{\text{MAX}}(\tilde{r})$ is equal to the infimum of $C_q(\tilde{r})$ over all pure strategies $q \in S_{\text{MIN}}$.*
2. *For any mixed strategy $\tilde{q}$ for* MIN, *$V_{\text{MIN}}(\tilde{q})$ is equal to the supremum of $C_{\tilde{q}}(r)$ over all pure strategies $r \in S_{\text{MAX}}$.*

The supremum of $V_{\text{MAX}}(\tilde{r})$ over all $\tilde{r} \in \tilde{S}_{\text{MAX}}$ is called the MAX-value of game G, and is denoted $V_{\text{MAX}}(G)$. A strategy $\tilde{r}$ that attains this supremum is called an *optimal* strategy for MAX. Similarly, the infimum of $V_{\text{MIN}}(\tilde{q})$ over all $\tilde{q} \in \tilde{S}_{\text{MIN}}$ is called the MIN-value of game G, and is denoted $V_{\text{MIN}}(G)$ and a strategy $\tilde{q}$ that attains this infimum is called an *optimal* strategy for MIN. In general, optimal strategies may exist for one, both or neither player.

The following elementary result is easily proved:

Lemma 2.2 *For any game G, $V_{\text{MAX}}(G) \leq V_{\text{MIN}}(G)$.*

A game G is said to have the *min-max property* if (i) $V_{\text{MAX}}(G) = V_{\text{MIN}}(G)$ and (ii) both players have an optimal strategy. For such a game, the common value is denoted $V(G)$ and is called the *value* of the game. The following fundamental theorem of two-person games, known as the min-max theorem, was proved by von Neumann (see, e.g., [vNM]):

Theorem 2.1 *Any finite two-player zero-sum game has the min-max property.*

A *subgame* of the game $G = (S_{\text{MAX}}, S_{\text{MIN}}, C)$ is a game $G' = (T_{\text{MAX}}, T_{\text{MIN}}, C)$ where $T_{\text{MAX}} \subseteq S_{\text{MAX}}$ and $T_{\text{MIN}} \subseteq S_{\text{MIN}}$. We say that G' is *equivalent* to G if (i) $V_{\text{MAX}}(G') = V_{\text{MAX}}(G)$, (ii) $V_{\text{MIN}}(G') = V_{\text{MIN}}(G)$, and (iii) MAX (resp. MIN) has an optimal strategy in G' if and only if MAX (resp. MIN) has an optimal strategy in G.

A subset $T_{\text{MAX}} \subseteq S_{\text{MAX}}$ *dominates* S_{MAX} with respect to the game G if for any strategy $r \in S_{\text{MAX}}$, there is a strategy $r' \in T_{\text{MAX}}$ such that $C_q(r') \geq C_q(r)$ for all $q \in S_{\text{MIN}}$. Similarly, a subset $T_{\text{MIN}} \subseteq S_{\text{MIN}}$ dominates S_{MIN} with respect to G if for any strategy $q \in S_{\text{MIN}}$, there is a strategy $q' \in T_{\text{MIN}}$ such that $C_{q'}(r) \leq C_q(r)$ for all $r \in S_{\text{MAX}}$. We have:

Proposition 2.1 *Let $G = (S_{\text{MAX}}, S_{MIN}, C)$ be a game, $T_{\text{MAX}} \subseteq S_{\text{MAX}}$ and $T_{\text{MIN}} \subseteq S_{\text{MIN}}$.*

1. *If T_{MAX} dominates S_{MAX} relative to G then the subgame $(T_{\text{MAX}}, S_{\text{MIN}}, C)$ is equivalent to the game G,*

2. *If T_{MIN} dominates S_{MIN} relative to G then the subgame $(S_{\text{MAX}}, T_{\text{MIN}}, C)$ is equivalent to the game G,*

Finally, we define the notion of the *competitive ratio* of a two-person game. Let G be a game whose payoff function is nonnegative. A *base-cost* function C_{BASE} for the game G is an $\mathbf{R}_\infty$-valued function defined on S_{MAX}. We say that the mixed strategy $\tilde{q}$ for MIN is *κ-competitive with respect to* C_{BASE} if there exists a constant K such that for any strategy r for MAX:

$$C_{\tilde{q}}(r) \leq \kappa C_{\text{BASE}}(r) + K.$$

The *competitive ratio* $\lambda = \lambda(G)$, with respect to C_{BASE}, is the infimum over all κ for which there is a κ-competitive algorithm, with respect to C_{BASE}.

A natural choice for a base-cost function is the *optimal cost function*, $C_{\text{OPT}}(r)$, which is defined as the infimum of $C_q(r)$ over all MIN-strategies q. (Notice that $C_{\text{OPT}}(r) = V_{\text{MAX}}(r)$.) In the context of online algorithms, the competitive ratio with base-cost $C_{\text{OPT}}(r)$ corresponds to the standard notion of the *randomized competitive ratio* of the associated online problem. We call C_{OPT} the *standard* base-cost function. In this paper, we will also have need to refer to non-standard base-costs.

The following result gives a criterion for upper bounding the competitive ratio (which is slightly more general than the definition):

Proposition 2.2 *Let G be a game with base-cost function C_{BASE}. Let $\kappa > 0$ and $f : \mathbf{R} \longrightarrow \mathbf{R}$ be a function such that $\lim_{x\to\infty} \frac{f(x)}{x} = \kappa$. Suppose that $\tilde{q}$ is a strategy for* MIN *such that for any strategy r for* MAX*,*

$$C_{\tilde{q}}(r) \leq f(C_{\text{BASE}}(r))$$

Then $\lambda(G) \leq \kappa$.

Proof: It is easy to check that the hypothesis of the proposition implies that $\tilde{q}$ is $\kappa + \epsilon$ competitive for any positive ϵ, which implies that $\lambda \leq \kappa$. ▮

The following result gives a criterion for lower bounding the competitive ratio.

Proposition 2.3 *Let G be a game with base-cost function C_{BASE}. Let $\kappa > 0$ and $f : \mathbf{R} \longrightarrow \mathbf{R}$ be a function such that $\lim_{x\to\infty} \frac{f(x)}{x} = \kappa$. Suppose that $\{\tilde{r}_i : i \in \mathrm{N}\}$ is a sequence of mixed strategies for* MAX *and $\{u_i : i \in \mathrm{N}\}$ is a sequence of real numbers tending to ∞, such that for each i, $C_{\mathrm{BASE}}(\tilde{r}_i) \leq u_i$ and for each pure* MIN*-strategy q:*

$$C_q(\tilde{r}_i) \geq f(u_i)$$

Then $\lambda(G) \geq \kappa$.

Proof: It suffices to show that if $\epsilon > 0$, there is no $(\kappa - \epsilon)$-competitive algorithm $\tilde{q}$. Suppose, to the contrary that $\tilde{q}$ is $(\kappa - \epsilon)$-competitive. Then there is a constant K such that for any pure strategy r

$$C_{\tilde{q}}(r) \leq (\kappa - \epsilon) C_{\mathrm{BASE}}(r) + K.$$

Now, by taking expectation with respect to the distribution $\tilde{r}_i$ we get:

$$\begin{aligned} C_{\tilde{q}}(\tilde{r}_i) &\leq (\kappa - \epsilon) C_{\mathrm{BASE}}(\tilde{r}_i) + K \\ &\leq (\kappa - \epsilon) u_i + K. \end{aligned}$$

For each i, there is a deterministic strategy q_i such that $C_{q_i}(\tilde{r}_i) \leq C_{\tilde{q}}(\tilde{r}_i)$. For that strategy we have $C_{q_i}(\tilde{r}_i) \leq (\kappa - \epsilon) u_i + K$. Thus, by the hypothesis of the proposition, $f(u_i) \leq (\kappa - \epsilon) u_i + K$, for every i. This contradicts the hypothesis that that $\lim_{x\to\infty} \frac{f(x)}{x} = \kappa$. ▮

Taking $u_i = C_{\mathrm{BASE}}(\tilde{r}_i)$ in the above Proposition yields:

Corollary 2.1 *Let G be a game with base-cost function C_{BASE}. Let $\kappa > 0$ and $f : \mathbf{R} \longrightarrow \mathbf{R}$ be a function such that $\lim_{x\to\infty} \frac{f(x)}{x} = \kappa$. Suppose that $\{\tilde{r}_i : i \in \mathrm{N}\}$ is a sequence of mixed strategies for* MAX *such that $C_{\mathrm{BASE}}(\tilde{r}_i)$ tends to ∞ and such that for each i and for each pure* MIN*-strategy q:*

$$C_q(\tilde{r}_i) \geq f(C_{\mathrm{BASE}}(\tilde{r}_i))$$

Then $\lambda(G) \geq \kappa$.

2.3 The Pursuit-Evasion Game: Definitions and Preliminary Results

The Pursuit-Evasion game for a metric space $\mathcal{M}$, denoted PE($\mathcal{M}$), is a two-person zero-sum game between two players, the Pursuer (the MAX player) and the Evader (the MIN player). Intuitively, the game is played as a sequence of rounds. At all times the Evader is located at some point of the space. In each round, the Pursuer probes some point of the metric space. If she picks the point containing the Evader then the Evader must move to some other point; otherwise the Evader may stay where he is. The cost to the Evader in responding to a sequence of probes is the total distance he travels.

In the language of two-player zero-sum games, the set of pure strategies of the Pursuer is the set $\mathcal{M}^*$ of all finite sequences from the metric space. Such sequences are referred to as *probe* sequences. A sequence σ is said to be a *response sequence* for the probe sequence ρ if it has the same term length as ρ and $\sigma_i \neq \rho_i$ for all i. The point σ_i represents the location of the Evader at time i. The pure strategies for the Evader are called *deterministic response algorithms* or simply *deterministic algorithms.* A response algorithm A maps each probe sequence ρ to a response sequence $A(\rho)$ subject to the following consistency requirement: for any probe sequence ρ and point a, $A(\rho a)$ extends $A(\rho)$. The consistency requirement formalizes the intuition that the algorithm determines the response sequence of the Evader in an on-line manner, i.e., the i^{th} response depends only on the first i probes.

The cost function $C_A(\rho)$ is defined to be $d(A(\rho))$, the metric length of the response sequence generated by the algorithm A on input ρ. $C_{\text{OPT}}(\rho)$ denotes the minimum of $C_A(\rho)$ over all algorithms A. It is easy to see that this is the same as the minimum metric length of a response sequence for ρ. (Note that in our definition, the Evader is allowed to choose his own starting point σ_1 at no cost. Other authors specify a starting point σ_0 and charge the Evader an additional $d(\sigma_0, \sigma_1)$. This is a matter of convention that does not affect the results.)

A mixed strategy for the Pursuer is a probability distribution $\tilde{\rho}$ over probe sequences. A mixed strategy for the Evader, called a *randomized response algorithm* is a probability distribution $\tilde{A}$ over response algorithms. Since the set of pure strategies for the Evader is uncountable, a probability distribution can not be specified by simply by assigning a probability value to each strategy. The general approach to mixed strategies on infinite strategy spaces requires measure theory; in our case the measure theoretic definition can be restated in algorithmic terms: in a randomized algorithm the s^{th} move of the Evader is chosen according to some probability distribution on $\mathcal{M}$, where the probability distribution may depend on the first s moves by the Pursuer and the first $s-1$ responses of the Evader.

We can define a randomized response algorithm formally via *decision trees.* Let $T = T_{\mathcal{M}}$ denote the infinite rooted tree of degree $|\mathcal{M}|$ where the edges from each node are in one-to-one correspondence with the points of $\mathcal{M}$ and the nodes are labeled as follows: the root is labeled P (for pursuer), the children of the root are labeled E (for evader), and the remaining nodes are labeled inductively P or E so that labels alternate along each path from the root. We can represent a (randomized) on-line response algorithm $\tilde{A}$ by assigning to each E-node v a probability distribution on $\mathcal{M}$, i.e., a nonnegative function p_v on $\mathcal{M}$ whose values sum to 1. On probe sequence $\rho_1, \rho_2, \ldots, \rho_k$, the algorithm follows the branch labeled ρ_1 from the root. It chooses its response according to the probability distribution p_v for its node, and follows the corresponding branch from v to the next P node. It then processes each successive probe in the same way, following down the tree to depth $2k$.

As in Section 2.2, we extend the definition of the cost function C to randomized strategies by defining $C_{\tilde{A}}(\tilde{\rho})$ to be the expectation of $C_A(\rho)$ with respect to the product distribution of $\tilde{A}$ and $\tilde{\rho}$. Also, $C_{\text{OPT}}(\tilde{\rho})$ is the expectation of $C_{\text{OPT}}(\rho)$ with respect to the distribution $\tilde{\rho}$. (It is important here to emphasize $C_{\text{OPT}}(\tilde{\rho})$ is *not* the same as the minimum of $C_A(\tilde{\rho})$ over all A. In computing the former, we choose the best algorithm for each deterministic ρ and average the cost with respect to $\tilde{\rho}$ while in the latter we choose the one algorithm that minimizes the average cost with respect to $\tilde{\rho}$.)

For $s > 0$, an s-*block* of $\mathcal{M}$ is a prefix-minimal probe sequence whose optimal cost is at least $s-\delta$. In other words, ρ is an s-block if $C_{\text{OPT}}(\rho) \geq s-\delta$, but $C_{\text{OPT}}(\tau) < s-\delta$ for any proper prefix τ of ρ. In the nondegenerate case that $\mathcal{M}$ has at least two points, an s-block ρ satisfies $s-\delta \leq C_{\text{OPT}}(\rho) < s$, since the last step can increase C_{OPT} by at most δ. In the degenerate case that $\mathcal{M}$ consists of a single point p and we define an s-block to be the singleton sequence.

Any probe sequence ρ can be parsed uniquely into subsequences $\boldsymbol{\rho_1 \rho_2 \ldots \rho_k}$ where each successive $\boldsymbol{\rho_i}$ except possibly the last is an s-block. We refer to this as the s-*block partition* of ρ.

The competitive ratio of $PE(\mathcal{M})$ is defined in terms of the standard base-function C_{OPT} and is denoted $\lambda(\mathcal{M})$. For brevity, we often refer to this as the *competitive ratio of* $\mathcal{M}$. It is trivial that the competitive ratio of a 2 point space is 1. From the previously noted result of [MMS] (which was proved for the deterministic competitive ratio) we have:

Lemma 2.3 *For any $\mathcal{M}$, $\lambda(\mathcal{M}) \leq |\mathcal{M}| - 1$.*

The pursuit-evasion game on a 1-point space does not really make sense. However, it will be convenient to adopt the convention that the competitive ratio of a 1-point space is 0. With that definition, the main decomposition theorem will hold when one or both spaces is a one-point space.

The following fact is both well known and easy to prove:

Proposition 2.4 *For any metric space $\mathcal{M}$ and subspace $\mathcal{N}$: $\lambda(\mathcal{M}) \geq \lambda(\mathcal{N})$.*

3 An overview of the decomposition theorem

We are working in a space $\mathcal{M}$ with bipartition $(\mathcal{M}_L, \mathcal{M}_R)$, which we call the *left* space and the *right* space. In the present discussion, we assume that each space has at least two points; the degenerate case that one of the spaces consists of a single point will require special treatment, which we delay until later. The assumptions of the theorem imply that the distances within each subspace are small relative to distances between the two subspaces.

We want to express the competitive ratio of the big space in terms of the competitive ratio of each of the two subspaces. The key idea is to abstract the behaviors of the Pursuer and the Evader so as to focus on their movements *between* the spaces, treating their movements within each space as a "black box". This idea leads to the formulation of a new game, called the Walker-Jumper game, which abstracts the pursuit-evasion game for such a partitioned space. This game is defined and analyzed in the next section, and the proof of the decomposition theorem is then completed in the following section.

The proof is technical, but the underlying idea is natural. In this section we provide intuition for the proof with an informal discussion that leads naturally to the definition of the Walker-Jumper game. Throughout the section we make various plausible but unjustified assumptions and approximations, which will be cleaned up in the proof.

At each point in the game, the Evader is either "on the left" or "on the right". While the Pursuer probes the opposite space the Evader need do nothing. While the Pursuer probes the subspace occupied by the Evader, it seems apparent that either the Evader should follow his optimal randomized response algorithm for that space (achieving, over that interval of moves, a competitive ratio equal to that for that space) or he should move to the other space. By randomizing his choice of when to switch between spaces, he can hope to "fool" the Pursuer as to his location.

We view the probe sequence of the Pursuer as a sequence of left phases and right phases, where a left (resp. right) phase consists of moves in the left (resp. right) space. When the Evader uses a randomized strategy, the Pursuer will only have a probability distribution on the location of the Evader. In order to maximize the competitive ratio, the Pursuer wants to construct a probe sequence that (i) has a good chance of catching the Evader often, and (ii) has a low offline cost. For the first goal, it would seem that she should always probe on the side with higher probability of containing the Evader, while for the second goal, it would seem that she would do well to avoid switching between spaces too often (this will make it easier for an offline algorithm to "hide" safely in one space for long intervals of moves) and thus should tend to make each phase long.

For $s > 0$, we defined an s-block to be a prefix-minimal probe sequence of cost at least $s - \delta$, and we observed that its cost is at most s. For some large integer D (to be specified later) we define $s = \delta/D$ where δ is the diameter of $\mathcal{M}$ and define a *left block* (resp., *right block*) to be a probe sequence which is an s-block with respect to the space $\mathcal{M}_L$ (resp., $\mathcal{M}_R$). Note that an s-block for $\mathcal{M}_L$ or $\mathcal{M}_R$ is not an s-block with respect to the entire space; indeed any such block has 0 optimal cost with respect to the entire space since an offline Evader will respond by staying at one location on the opposite space.

Recall that the s-block partition provides a canonical way to parse every probe sequence from a space as a concatenation of sequences each of which, except possibly the last, is an s-block. For a given probe sequence of $\mathcal{M}$, parse each left phase according to its s-block partition with respect to $\mathcal{M}_L$ and parse each right phase similarly. It is reasonable to expect that if s is small relative to the typical cost of a phase then we may ignore the "remainder" block of the phase and simply assume that each left (resp., right) phase is a concatenation of left (resp., right) blocks.

With this assumption, we view the entire probe sequence as a sequence of left and right blocks. If the Pursuer chooses to add a right block, it would seem natural that she select the right block so that, assuming the Evader is on the right, the expected cost to the Evader is maximized. This expected cost can not exceed $s\lambda_R$ (by much), since the Evader can follow his λ_R-competitive strategy for the space $\mathcal{M}_R$. On the other hand, the Pursuer can force an Evader who stays on the right to incur nearly this much (in expectation) by selecting her right block according to her optimal randomized s-block strategy $\tilde{\rho}_{\mathrm{s}} = \tilde{\rho}_{\mathrm{s}}(\mathcal{M}_R)$. Similarly when the Pursuer picks a left block, she can force an Evader who stays on the left to pay roughly $s\lambda_L$.

To summarize, when the Pursuer adds a block, if the Evader is on the opposite side he pays nothing. If the Evader is on the same side, he either moves to the other side immediately, paying roughly δ or he stays on the original side and incurs an expected cost of $s\lambda_R$ or $s\lambda_L$ depending on the side. We assume that δ is much larger than both $s\lambda_R$ and $s\lambda_L$ so that it would not pay for him to move to the opposite side at the beginning and back at the end the block.

Thus, we have a good approximation to the cost of each block to the Evader, depending only on (i) the side from which each block is chosen by the Pursuer (ii) the side on which the Evader finishes responding to each block.

We would like to get a similar estimate for the offline cost. For each probe sequence ρ, define $C_L(\rho)$ to be the minimum cost of a response sequence σ whose last point is on the left and $C_R(\rho)$ in the analogous way. We refer to these respectively as the left-optimal and right-optimal cost of ρ. The optimal cost of ρ is just the minimum of these. Since they clearly differ by at most the diameter δ of $\mathcal{M}$, we may take $C_R(\rho)$ as a good estimate of $C_{\mathrm{OPT}}(\rho)$. We want to understand how C_L and C_R change when the Pursuer adds a left block or a right block. So, let ρ be the sequence constructed so far and consider adding a right block β. It is easy to see that $C_L(\rho\beta) = C_L(\rho)$, as $C_L(\rho\beta) = \min\{C_L(\rho), C_R(\rho)+\delta\}$ and $C_L(\rho) \leq C_R(\rho)+\delta$. On the other hand, we can estimate: $C_R(\rho\beta) \approx \min\{C_L(\rho)+\delta, C_R(\rho)+s\}$, where the two terms correspond to the two choices of the offline Evader: either finish the previous block on the left and move right only at the end of β or finish the previous block on the right and respond to β on the right.

Let us summarize this discussion, by considering the evolution of the parameter $w = (C_R - C_L)/s$. Note that w is always between $-D$ and D. Each right block increases w by (roughly) 1, subject to $w \leq D$ and, similarly, each left block decreases w by 1, subject to $w \geq -D$. It is useful to visualize the evolution of w as a walk on the integer points between $-D$ and D. A right step corresponds to w being increased by 1, and C_R being increased by s. Similarly, a left step corresponds to w being decreased by 1, and C_L being decreased by s.

Notice that if the Pursuer adds a right block that causes w to reach D then the Pursuer may add an arbitrary number of right blocks without affecting C_L, C_R or w. Let us assume that she does this, i.e., a right step from $D-1$ to D corresponds to a huge number of right blocks. The effect of this on the Evader is clear: on such a step he must move to the left space (if he is not there already) or incur a huge cost. (Note that the Evader can compute C_R, C_L and w online and thus recognize this situation).

Similarly we may assume that a left step from $1 - D$ to $-D$ corresponds to a huge number of left blocks and the Evader must move to the right space.

Having associated the Pursuer's probe sequence to a walk on the integer line we now can make the following estimates of the offline cost and the Evader's cost. The offline cost is estimated by s times the number of steps to the right. The Evader's cost is 0 on any step that is taken in the direction opposite the space he occupies. On a step taken in the direction of the space that he occupies, his cost is sD if he chooses to move to the other space and is $s\lambda_R$ or $s\lambda_L$ (depending on his space) if he chooses to stay in his space. Whenever a right (resp., left) step is taken that reaches D (resp.,$-D$), the Evader must move to the left (resp., right) space.

This idealization suggests that we can model our problem by a game between two players, the *Walker* who walks on the line (and corresponds to the Pursuer) and the *Jumper* who jumps between the left and right side (and corresponds to the Evader). In the next section we define this game precisely and analyze it. In the succeeding section, we then formally show how to use the analysis of this game to establish the decomposition theorem.

4 Walker-Jumper games

The Walker-Jumper game $WJ[D, \alpha_R, \alpha_L]$ has parameters D, a positive integer, and nonnegative real numbers α_R and α_L. The players are referred to as the Walker (Wendy) and the Jumper (Jack). The game "board" is the set $I_D = \{-D, -D + 1, \ldots, D-1, D\}$. At each integer time $t \geq 0$, the position of the game is the ordered pair (w_t, j_t), where $w_t \in I_D$ is Wendy's position and $j_t \in \{-D, D\}$ is Jack's location. The initial position for Wendy is $w_0 = 0$ and Jack can choose either $j_0 = -D$ or $j_0 = D$.

At each time t, a legal move for Wendy consists of one step either to the left ($w_t = w_{t-1} - 1$) or to the right ($w_t = w_{t-1} + 1$). If $|w_t| = D$, Wendy has only one legal move. Thus, the sequence $\mathbf{\Delta w}$ defined by $(\Delta w)_i = w_i - w_{i-1}$ has entries in the set $\{-1, 1\}$. We refer to Δw_i as the *direction* of move i. A move of -1 (resp. $+1$) will be referred to as a left move (resp. right move). Also, for convenience of notation, we sometimes use α_{-1}, α_1 in place of α_L, α_R.

Jack's answer to request w_t is either to stay where he is ($j_t = j_{t-1}$) or to *jump* to his other allowed position ($j_t = -j_{t-1}$). Jack's moves are subject to constraint that $j_t \neq w_t$, i.e., if Wendy arrives at Jack's location ($w_t = j_{t-1}$), then Jack must jump

$(j_t = -w_t)$.

If $j_t = (\Delta w)_t D$, i.e., Wendy's move at time t brought her closer to the location that Jack occupies *after* his move, then we say that Wendy *hit* Jack; it is a *left hit* or a *right hit* depending on the direction of Wendy's move.

Formally, a (pure) strategy for Wendy (a request sequence) is given by a sequence $\mathbf{w} = (w_0 = 0, w_1, w_2, \ldots)$ having entries in I_D and satisfying $|w_t - w_{t-1}| = 1$ for each $t > 0$. A pure strategy for Jack is a function (algorithm) A that maps each finite request sequence $\mathbf{w} = (0, w_1, \ldots, w_s)$ to a sequence $(j_0, j_1, j_2, \ldots j_s)$ in $\{-D, D\}^{s+1}$. The map satisfies the constraint $A(\mathbf{w})_i \neq w_i$ for each i, and it also satisfies the consistency constraint that if $\mathbf{w}$ is an extension of $\mathbf{v}$ then $A(\mathbf{w})$ is an extension of $A(\mathbf{v})$; this constraint means that A is an on-line algorithm. As usual, we also consider randomized strategies for both players. A randomized strategy for Wendy is a probability distribution $\tilde{\mathbf{w}}$ on request sequences and a randomized strategy for Jack is a probability distribution $\tilde{A}$ on algorithms.

The cost function for algorithm A on request sequence $\mathbf{w}$, $C_A(\mathbf{w})$ (representing the cost to Jack), is given as follows.

1. Each jump by Jack costs him D,
2. Each right hit by Wendy costs Jack α_R,
3. Each left hit by Wendy costs Jack α_L.

Thus the cost of step t to Jack, $(\Delta C_A(\mathbf{w}))_t = C_A(\mathbf{w^t}) - C_A(\mathbf{w^{t-1}})$, can be written:

$$(\Delta C_A(\mathbf{w}))_t = D\chi(j_t = -j_{t-1}) + \alpha_{((\Delta w)_t)}\chi(j_t = (\Delta w)_t D),$$

where, for the predicate P, $\chi(P) = 1$ if P is true and 0 otherwise.

As usual, if $\tilde{A}$ and $\tilde{\mathbf{w}}$ are randomized strategies then $C_{\tilde{A}}(\tilde{\mathbf{w}})$ is defined to be the expectation of $C_A(\mathbf{w})$ with respect to the distributions.

We are interested in the competitive ratio $\lambda = \lambda(WJ[D, \alpha_R, \alpha_L])$ of this game with respect to a nonstandard base-cost function: $C_{\text{BASE}}(\mathbf{w})$ is the total number of steps to the right. Note that $C_{\text{BASE}}(\mathbf{w})$ is within $\pm D$ of $|\mathbf{w}|/2$. Applying Proposition 2.2, we obtain the following criterion for upper bounding λ:

Corollary 4.1 *Let $\tilde{A}$ be an algorithm for Jack. Suppose that b is a positive real such that for all $j \in \mathrm{N}$ and sequences $\mathbf{w}$ of term length j:*

$$C_{\tilde{A}}(\mathbf{w}) \leq j(b + g(j)),$$

where $g(j)$ is a function that tends to 0 as j tends to ∞. Then $\lambda \leq 2b$.

Proof: Let $g'(x) = \max_{\{d: -D \leq d \leq D\}} g(x + d)$. The hypothesis implies that for any $\mathbf{w}$, $C_{\tilde{A}}(\mathbf{w}) \leq (C_{\mathrm{BASE}}(\mathbf{w}) + D/2)(b + g'(C_{\mathrm{BASE}}(\mathbf{w})))$. So the corollary follows if we take $f(x) = (x + D/2)(b + g'(x))$, and $c = 2b$ in Proposition 2.2. □

Similarly, we get a criterion for lower bounding λ from Corollary 2.1. Recall that $\mathbf{w^j}$ denotes the prefix of $\mathbf{w}$ up to w_j:

Corollary 4.2 *Let $g(j) \to 0$ as $j \to \infty$. Let $\tilde{\mathbf{w}}$ be a distribution over infinite sequences for Wendy. Suppose that b is a positive real such that for any algorithm A for Jack and $j \in \mathrm{N}$:*

$$C_A(\tilde{\mathbf{w}}^{\mathbf{j}}) \geq j(b - g(j)),$$

then $\lambda \geq 2b$.

An algorithm A for Jack is called *lazy* if Jack never jumps when Wendy moves away from him, i.e., if $(\Delta w)_t \neq j_{t-1}/D$ then $j_t = j_{t-1}$. It is easy to show that any nonlazy strategy is dominated by some lazy one in the sense that the lazy strategy performs at least as well on any request sequence by Wendy. Thus we may assume that Jack is restricted to lazy strategies.

We consider only the case that D is at least $\alpha_{\max} = \max\{\alpha_R, \alpha_L\}$. To develop some intuition for this game, let us first show that $\alpha_{\max} \leq \lambda \leq \alpha_L + \alpha_R + 1$.

To see the lower bound assume, without loss of generality, that $\alpha_L \geq \alpha_R$ and consider the following pure strategy for Wendy: move right D times to position D and then alternately move between $D-1$ and D. Each time Wendy moves from D to $D-1$, Jack must start at $-D$ and pays $\alpha_{\max}$ (if he doesn't move) or D (if he does). After j steps by Wendy, Jack's cost is at least $\alpha\frac{j-D}{2} = j(\frac{\alpha_{\max}}{2} - \frac{D\alpha_{\max}}{2j})$. Applying Corollary 4.2 yields a lower bound of $\alpha_{\max}$ on the competitive ratio.

To see the upper bound of $\alpha_L + \alpha_R + 1$, consider the following pure strategy for Jack: never jump unless a jump is required (because Wendy arrives at the same location). On any j step sequence, Wendy takes at most $j/2 + D$ steps in each direction (since the number of left steps differs from the number of right steps by at most D) and Jack jumps at most $j/(2D)$ times (since he jumps at most once every $2D$ steps). Thus Jack's cost can be bounded above by $(\alpha_R + \alpha_L)(j/2 + D) + j/2$.

Applying Corollary 4.1 implies an upper bound of $\alpha_R + \alpha_L + 1$ on the competitive ratio.

As we shall see, the trivial lower bound above is much closer to the truth than the trivial upper bound. The main result of this section is an exact expression for the competitive ratio $\lambda(WJ[D, \alpha_R, \alpha_L])$. Define $\beta_R = 1 - \alpha_R/2D$ and $\beta_L = 1 - \alpha_L/2D$.

Theorem 4.1 *For nonnegative real numbers α_R and α_L and positive integer $D \geq max\{\alpha_L, \alpha_R\}$, the competitive ratio λ of the game $WJ[D, \alpha_R, \alpha_L]$ is given by:*

$$\lambda = \begin{cases} \alpha_R + \beta_R & \text{if } \alpha_R = \alpha_L, \\ \frac{\alpha_R \beta_L^{2D} - \alpha_L \beta_R^{2D}}{\beta_L^{2D} - \beta_R^{2D}} & \text{if } \alpha_R \neq \alpha_L \end{cases}$$

Using elementary estimates and recalling that we defined $Z(x) = x/(e^x - 1)$ for $x > 0$ and $Z(0) = 1$ we obtain:

Corollary 4.3 *For nonnegative real numbers α_R and α_L, define $\alpha_{\max} = max\{\alpha_R, \alpha_L\}$ and $\alpha_{\text{diff}} = |\alpha_R - \alpha_L|$. Suppose that $\alpha_{\max} \geq 1$. Let $D \geq \alpha_{\max}^2$ be a positive integer. Then the competitive ratio λ of $WJ[D, \alpha_R, \alpha_L]$ satisfies:*

$$\alpha_{\max} + Z(\alpha_{\text{diff}})(1 - \epsilon) \leq \lambda \leq \alpha_{\max} + Z(\alpha_{\text{diff}})(1 + \epsilon)$$

where $\epsilon = \frac{4\alpha_{\max}^2}{D}$.

We first deduce the Corollary from the Theorem.

Proof: (of Corollary 4.3).

The case $\alpha_R = \alpha_L$ is trivial since $Z(0) = 1$. So assume $\alpha_R \neq \alpha_L$. From the Theorem and simple algebraic manipulation we get:

$$\lambda = \frac{\alpha_R \beta_L^{2D} - \alpha_L \beta_R^{2D}}{\beta_L^{2D} - \beta_R^{2D}} = \alpha_{\max} + \frac{\alpha_{\text{diff}}}{(1 + \frac{\alpha_{\text{diff}}}{2D - \alpha_{\max}})^{2D} - 1}.$$

To prove the Corollary, we need to show that for $\epsilon = 4\alpha_{\max}^2/D$:

$$\left| \frac{1}{(1 + \frac{\alpha_{\text{diff}}}{2D - \alpha_{\max}})^{2D} - 1} - \frac{1}{e^{\alpha_{\text{diff}}} - 1} \right| \leq \frac{\epsilon}{e^{\alpha_{\text{diff}}} - 1},$$

or equivalently:

$$\left| \frac{(1 + \frac{\alpha_{\text{diff}}}{2D - \alpha_{\max}})^{2D} - e^{\alpha_{\text{diff}}}}{(1 + \frac{\alpha_{\text{diff}}}{2D - \alpha_{\max}})^{2D} - 1} \right| \leq \epsilon.$$

Note that $(1+\frac{\alpha_{\rm diff}}{2D-\alpha_{\rm max}})^{2D} \geq e^{\alpha_{\rm diff}}$, by taking $x = (2D-\alpha_{\rm diff})/\alpha_{\rm diff}$ in the inequality $(1+1/x)^{x+1} \geq e$, which holds for all positive x. Thus we may remove the absolute value from the inequality to be proved. Replacing the denominator by a smaller quantity it now suffices to show:

$$\frac{(1+\frac{\alpha_{\rm diff}}{2D-\alpha_{\rm max}})^{2D} - e^{\alpha_{\rm diff}}}{e^{\alpha_{\rm diff}}-1} \leq \epsilon.$$

Next we upper bound $(1+\frac{\alpha_{\rm diff}}{2D-\alpha_{\rm max}})^{2D}$:

$$\begin{aligned}(1+\frac{\alpha_{\rm diff}}{2D-\alpha_{\rm max}})^{2D} &= e^{2D\ln(1+\frac{\alpha_{\rm diff}}{2D-\alpha_{\rm max}})} \leq e^{\frac{2D\alpha_{\rm diff}}{2D-\alpha_{\rm max}}} \leq e^{\alpha_{\rm diff}} e^{\frac{\alpha_{\rm diff}\alpha_{\rm max}}{2D-\alpha_{\rm max}}} \\ &\leq e^{\alpha_{\rm diff}} e^{\frac{\alpha_{\rm diff}\alpha_{\rm max}}{D}} \leq e^{\alpha_{\rm diff}}(1+\frac{2\alpha_{\rm diff}\alpha_{\rm max}}{D})\end{aligned}$$

(The last inequality uses the assumption $\alpha_{\rm max}^2 \leq D$.) Thus it suffices to show:

$$\frac{e^{\alpha_{\rm diff}} 2\alpha_{\rm diff}\alpha_{\rm max}/D}{e^{\alpha_{\rm diff}}-1} \leq \epsilon.$$

If $e^{\alpha_{\rm diff}} \leq 2$ then, since the denominator is at least $\alpha_{\rm diff}$, the expression on the left is at most $4\alpha_{\rm max}/D$. If $e^{\alpha_{\rm diff}} > 2$ then $e^{\alpha_{\rm diff}}/(e^{\alpha_{\rm diff}}-1) < 2$, and the expression is at most $4\alpha_{\rm diff}\alpha_{\rm max}/D \leq 4\alpha_{\rm max}^2/D$.

∎

We now return to the proof of Theorem 4.1. This is proved in two parts:

1. We analyze a specific randomized algorithm for Jack and use Corollary 4.1 to obtain an upper bound on λ.
2. We analyze a specific randomized strategy for Wendy (a probability distribution on request sequences) and use Corollary 4.2 to obtain a lower bound on λ.

4.1 The upper bound: a strategy for the Jumper

We begin with an explicit description of a randomized strategy for Jack. The strategy is simple but not particularly intuitive; we will motivate it as we analyze it.

Jumper Strategy.

1. The strategy is defined in terms of $2D+1$ parameters $0 = p_{-D} < p_{-D+1} < p_{-D+2} < \ldots < p_{D-1} < p_D = 1$ which are specified below. Initially, Jack chooses his initial position to be $-D$ with probability p_0. At round $t+1$, if Wendy moves in the direction away from j_t, Jack does not move. If $j_t = D$ and Wendy moves rightward from $w_t = j-1$ to $w_{t+1} = j$ then Jack moves to $-D$ with probability $\frac{p_j - p_{j-1}}{1-p_{j-1}}$. The probability of jumping from D to $-D$ is chosen so that if Jack occupied $-D$ with probability p_{j-1}, then Jack is now at $-D$ with probability $p_{j-1} + (1-p_{j-1})\frac{p_j - p_{j-1}}{1-p_{j-1}} = p_j$. If $j_t = -D$ and Wendy moves leftward from $w_t = j+1$ to $w_{t+1} = j$ then Jack moves to D with probability $\frac{p_{j+1}-p_j}{p_{j+1}}$. Here, the probabiliity of jumping from $-D$ to D is chosen so that if Jack occupied $-D$ with probability p_{j+1}, then Jack is now at $-D$ with probability $p_{j+1}\left(1 - \frac{p_{j+1}-p_j}{p_{j+1}}\right) = p_j$.

   Notice that this strategy ensures that at all times t, Jack is at $-D$ with probability p_{w_t}: at the start, Wendy is at location 0 and Jack is at $-D$ with probability p_0; and, as noted above, this property is maintained inductively.

2. The parameters p_i that are used are given by:

$$p_i = \begin{cases} \frac{1}{2} + \frac{i}{2D} & \text{if } \alpha_R = \alpha_L, \\ \frac{\beta_L^i \beta_R^{2D-i} - \beta_R^{2D}}{\beta_L^{2D} - \beta_R^{2D}} & \text{if } \alpha_R \neq \alpha_L \end{cases}$$

For a given sequence $\mathbf{w}$ for Wendy, let $N_{j,j'} = N_{j,j'}(\mathbf{w})$ be the total number of steps that Wendy takes from j to j' (which can be nonzero only if $j' \in \{j-1, j+1\}$). Recall that each right hit costs Jack α_R, each left hit costs Jack α_L, and each jump costs Jack D. So, each time Wendy takes a step from j to $j+1$, Jack pays α_R with probability $1 - p_{j+1}$ (the probability Jack is at D after the step is taken), and pays D with probability $p_{j+1} - p_j$. Similarly, each time Wendy moves from $j+1$ to j, Jack's expected cost is $p_j \alpha_L + (p_{j+1} - p_j)D$. Therefore, we can express the expected cost to Jack of the sequence $\mathbf{w}$ as:

$$\begin{aligned} C_{\tilde{A}}(\mathbf{w}) &= \sum_{j=-D}^{D-1} N_{j,j+1}((1-p_{j+1})\alpha_R + (p_{j+1}-p_j)D) + N_{j+1,j}(p_j\alpha_L + (p_{j+1}-p_j)D) \\ &\leq \sum_{j=-D}^{D-1} (N_{j,j+1}+1)((1-p_{j+1})\alpha_R + p_j\alpha_L + 2(p_{j+1}-p_j)D). \end{aligned}$$

The inequality follows from the fact that $N_{j,j+1} + 1 \geq N_{j+1,j}$.

The expression $r_j = (1 - p_{j+1})\alpha_R + p_j\alpha_L + 2(p_{j+1} - p_j)D$ that multiplies $N_{j,j+1}$ can be interpreted as the expected cost to Jack if Wendy makes a "round trip" from j to $j+1$. The specific definition of p_j given in Jack's strategy was chosen so that r_j is the same for all j. These values can be determined by introducing a parameter K, setting the $r_j = K$ for each j, solving the resulting linear recurrence relation for p_j, and then using $\sum p_j = 1$ to determine K. As is easily verified, this yields:

$$K = \begin{cases} \alpha_R + \beta_R & \text{if } \alpha_R = \alpha_L, \\ \frac{\alpha_R \beta_L^{2D} - \alpha_L \beta_R^{2D}}{\beta_L^{2D} - \beta_R^{2D}} & \text{if } \alpha_R \neq \alpha_L \end{cases}$$

As noted, $\sum_{j=-D}^{D-1} N_{j,j+1} \leq \frac{1}{2}|\mathbf{w}| + D$. This leads to:

$$\begin{aligned} C_{\tilde{A}}(\mathbf{w}) &\leq \sum_{j=-D}^{D-1} (N_{j,j+1} + 1) r_j \\ &\leq K \sum_{j=-D}^{D-1} (N_{j,j+1} + 1) \\ &\leq \frac{1}{2}|\mathbf{w}|K + 3DK. \end{aligned}$$

Applying Corollary 4.1, yields $\lambda \leq K$ as desired.

4.2 The lower bound: a strategy for the Walker

We now prove a matching lower bound by describing and analyzing a randomized strategy for Wendy. As for the upper bound, the strategy is simple; we describe it first and motivate it as we analyze it. Essentially, Wendy's strategy is to follow a biased random walk on the line, where the bias to the left or right depends on the direction taken in the previous step.

Walker Strategy

1. The strategy is defined in terms of two parameters σ_R and σ_L, which are real numbers between 0 and 1. If $w_t = D$ or $w_t = -D$ then w_{t+1} is forced. Otherwise $|w_t| < D$ and Wendy moves as follows. If move t was to the right ($w_t = w_{t-1}+1$) then Wendy goes left at step $t + 1$ with probability σ_L and to the right with probability $1-\sigma_L$. Similarly if move t was to the left ($w_t = w_{t-1}-1$) then Wendy goes right at step $t+1$ with probability σ_R and to the left with probability $1-\sigma_R$.

2. The parameters σ_R and σ_L are defined by $\sigma_R = \alpha_L/2D = 1 - \beta_L$ and $\sigma_L = \alpha_R/2D = 1 - \beta_R$.

The intuition behind this strategy is the following: We want to choose a strategy for Wendy that guarantees that the ratio of Jack's cost to the base-cost is at least a certain value, regardless of what Jack's algorithm is. This suggests that we seek a strategy that has the property that the expected cost to Jack is essentially independent of what Jack does. This was the guiding principle in designing this strategy.

Fix a lazy deterministic strategy A for Jack and define $C(\mathbf{w}) = C_A(\mathbf{w})$. We are interested in lower bounding the expectation of $C(\mathbf{w})$ with respect to the above distribution for Wendy. It will be convenient to introduce a modified cost function, $C^*(\mathbf{w}) = C(\mathbf{w}) + \psi(\mathbf{w})$, where the correction ψ depends only on the final step: if the last step by Wendy scored a hit on Jack (precisely, $j_s = D(w_s - w_{s-1})$) then $\psi(\mathbf{w}) = D - \alpha_R$ if the hit was to the right and $D - \alpha_L$ if the hit was to the left. Otherwise $\psi(\mathbf{w}) = 0$. Thus, $C(\mathbf{w}) \geq C^*(\mathbf{w}) - D$.

The purpose of introducing this modified cost is that with respect to this cost measure, the cost to Jack of any given step does not depend on what Jack does at that step. More precisely, define Jack's *modified cost at step t* to be $(\Delta C^*)_t(\mathbf{w}) = C^*(\mathbf{w^t}) - C^*(\mathbf{w^{t-1}})$. Since we are assuming that Jack is following a lazy strategy, Jack has an option to move only if $j_{t-1} = (\Delta w)_t D$. In this case, if he does not jump then his true cost increases by $\alpha_{(\Delta w)_t}$ while his modified cost increases by $\alpha_{(\Delta w)_t} + \psi(\mathbf{w^t}) - \psi(\mathbf{w^{t-1}}) = D - \psi(\mathbf{w^{t-1}})$. If he jumps at time t, then his true cost goes up by D but his modified cost goes up by $D + \psi(\mathbf{w^t}) - \psi(\mathbf{w^{t-1}}) = D - \psi(\mathbf{w^{t-1}})$, which is the same as if he did not jump. The reader can now check the following:

Lemma 4.1 *For any fixed lazy strategy A for Jack, the change $(\Delta C^*)_t(\mathbf{w})$ in the modified cost at time t is given by the following table:*

$(\Delta w)_t$	$(\Delta w)_{t-1}$	j_{t-1}	$(\Delta C^*)_t(\mathbf{w})$
$+1$	$+1$	$+D$	α_R
$+1$	$+1$	$-D$	0
-1	$+1$	$+D$	$\alpha_R - D$
-1	$+1$	$-D$	D
-1	-1	$-D$	α_L
-1	-1	$+D$	0
$+1$	-1	$-D$	$\alpha_L - D$
$+1$	-1	$+D$	D

▮

Recall that if $|w_{t-1}| = D$ then, by the rules of the game, it must be the case that $j_{t-1} = -w_{t-1}$, $(\Delta w)_{t-1} = w_{t-1}/D$,and $(\Delta w)_t = j_{t-1}/D$. Thus, in this case $(\Delta C^*)_t(\mathbf{w}) = D$.

If $|w_{t-1}| < D$ then when Wendy moves at time t, her move can depend on $(\Delta w)_{t-1}$, the direction of her last move, but not on j_{t-1}, which she does not know. So we try a strategy for Wendy in which her probability of moving in each direction depends on the direction of her last move. This motivates condition 1 in the definition of the strategy. So consider a strategy satisfying this condition, with σ_R and σ_L as yet unspecified. We now can write down an expression in terms of w_{t-1} and j_{t-1}, for the expectation (with respect to this randomized strategy) in the change of the modified cost at time t, for the case $|w_{t-1}| < D$.

$$(\Delta C^*)_t(\tilde{\mathbf{w}}) = \begin{cases} \alpha_R - D\sigma_L & \text{if } ((\Delta w)_{t-1}, j_{t-1}) = (+1, +D) \\ D\sigma_L & \qquad\qquad\qquad\qquad = (+1, -D) \\ \alpha_L - D\sigma_R & \qquad\qquad\qquad\qquad = (-1, -D) \\ D\sigma_R & \qquad\qquad\qquad\qquad = (-1, +D) \end{cases}$$

By selecting σ_L and σ_R so that this expectation is independent of j_{t-1}, we obtain condition 2 of the strategy.

Having motivated Wendy's strategy, we now continue with its analysis. The change in the expected modified cost at time t can now be written:

$$(\Delta C^*)_t(\tilde{\mathbf{w}}) = \begin{cases} D & \text{if } |w_{t-1}| = D \\ \alpha_R/2 & \text{if } |w_{t-1}| < D \text{ and } (\Delta w)_{t-1} = +1 \\ \alpha_L/2 & \text{if } |w_{t-1}| < D \text{ and } (\Delta w)_{t-1} = -1 \end{cases}$$

To apply Corollary 4.2, we need to lower bound Jack's expected cost against the sequence $\mathbf{w}^j$ generated by the first j steps of Wendy's strategy.

Wendy's strategy can be described by a Markov chain with state space $\{L_i : -D \leq i < D\} \bigcup \{R_i : -D < i \leq D\}$ where Wendy is in state L_i at time $t-1$ if $w_{t-1} = i$ and $(\Delta_w)_{t-1} = -1$ (she is at point i and her last move was to the left) and Wendy is in state R_i at time $t-1$ if $w_{t-1} = i$ and $(\Delta_w)_{t-1} = +1$ (she is at point i and her last move was to the right). For state U, let $N_j(U)$ denote the expected number of visits to state U during the first j steps by Wendy. Also, let $p(U)$ denote the steady state probability for state U. For large j, $N_j(U) = p(U)j(1 + o(1))$. Thus for a sequence $\tilde{\mathbf{w}}$ of j steps chosen according to Wendy's strategy:

$$
\begin{aligned}
C^*(\tilde{\mathbf{w}}) &= D[N_j(L_{-D}) + N_j(R_D)] + \frac{1}{2}\sum_{i=1-D}^{D-1}[N_j(L_i)\alpha_L + N_j(R_i)\alpha_R] \\
&= j(1+o(1))[D(p(L_{-D}) + p(R_D)) + \frac{1}{2}\sum_{i=1-D}^{D-1}(p(L_i)\alpha_L + p(R_i)\alpha_R)].
\end{aligned}
$$

Applying Corollary 4.2, we obtain:

$\lambda \geq 2[D(p(L_{-D}) + p(R_D)) + \frac{1}{2}\sum_{i=1-D}^{D-1}(p(L_i)\alpha_L + p(R_i)\alpha_R)]$.

We proceed to determine the steady state probabilities of the Markov chain. The transition matrix of the chain yields the following equations for the steady state probabilities:

$$
\begin{aligned}
p(L_i) = &\ (1-\sigma_R)p(L_{i+1}) + \sigma_L p(R_{i+1}) \quad \text{if } -D \leq i < D-1, \\
p(L_{D-1}) = &\ p(R_D), \\
p(R_i) = &\ (1-\sigma_L)p(R_{i-1}) + \sigma_R p(L_{i-1}) \quad \text{if } 1-D < i \leq D, \\
p(R_{1-D}) = &\ p(L_{-D}).
\end{aligned}
$$

Solving this system and recalling that $\beta_R = 1 - \alpha_R/2D = 1 - \sigma_L$, and $\beta_L = 1 - \alpha_R/2D = 1 - \sigma_R$ we get the following solution (which can be easily checked against the defining equations):

$$
\begin{aligned}
p(L_i) &= \frac{J}{4D}(\frac{\beta_R}{\beta_L})^i, &(1)\\
p(R_i) &= \frac{J}{4D}(\frac{\beta_R}{\beta_L})^{i-1}, &(2)\\
& &(3)
\end{aligned}
$$

where

$$
J = \begin{cases} 1 & \text{if } \alpha_R = \alpha_L, \\ \frac{(\alpha_R - \alpha_L)(\beta_L\beta_R)^D}{\beta_L(\beta_L^{2D} - \beta_R^{2D})} & \text{if } \alpha_R \neq \alpha_L \end{cases}
$$

is chosen so that the sum of the probabilities is 1. Notice that $p(L_i) = p(R_{i+1})$ for all i and therefore $\sum_i p(L_i) = \sum_i p(R_{i+1}) = 1/2$. Thus we can rewrite the lower bound on λ as:

$$
\begin{aligned}
\lambda &= 2[D(p(L_{-D}) + p(R_D)) + (\frac{1}{2} - p(L_{-D}))\frac{\alpha_L}{2} + (\frac{1}{2} - p(R_D))\frac{\alpha_R}{2}] \\
&= 2[D(\beta_L p(L_{-D}) + \beta_R p(R_D)) + \frac{\alpha_L + \alpha_R}{4}] \\
&= \frac{1}{2}[(\frac{J(\beta_L^{2D} + \beta_R^{2D})}{(\beta_L^{D-1}\beta_R^D)} + \alpha_L + \alpha_R].
\end{aligned}
$$

A routine calculation shows that this simplifies to the expression in the theorem.

▮

5 Proof of the Decomposition Theorem

This section is devoted to the proof of Theorem 1.7. Recall the notation of the theorem: $\mathcal{M}$ is a metric space partitioned into subspaces $\mathcal{M}_L$ and $\mathcal{M}_R$. Their respective diameters and competitive ratios are denoted $\delta, \delta_L, \delta_R$ and $\lambda, \lambda_L, \lambda_R$. Also $\delta_{\max} = \max\{\delta_L, \delta_R\}$, $\lambda_{\max} = \max\{\lambda_L, \lambda_R\}$, and $\lambda_{\text{diff}} = |\lambda_L - \lambda_R|$.

The overview of the proof in Section 3 developed the Walker-Jumper game as a rough model for the pursuit-evasion game on a partitioned space. We now make this connection precise.

Lemma 5.1 *Let $\mathcal{M}_L, \mathcal{M}_R$ be a partition of $\mathcal{M}$ such that $\frac{\delta}{\delta_{\max}}$ is at least 32. Let α_R and α_L be nonnegative numbers and $\alpha_{\max}$ be their maximum. Let D be an integer satisfying* $\max\{2\lambda_{\max} + 2, \sqrt{\frac{\delta}{\delta_{\max}}}\} \leq D \leq \frac{\delta}{4\delta_{\max}}$.

1. *If $\alpha_L \geq \lambda_L$ and $\alpha_R \geq \lambda_R$ then:*

$$\lambda \leq \lambda(WJ[D, \alpha_R, \alpha_L])(1 + \eta),$$

2. *If $\alpha_L \leq \lambda_L$ and $\alpha_R \leq \lambda_R$ then:*

$$\lambda \geq \lambda(WJ[D, \alpha_R, \alpha_L])(1 - \eta),$$

where

$$\eta \leq 6\frac{\delta_{\max} D}{\delta}.$$

Given Lemma 5.1 and the analysis of the Walker-Jumper game, the decomposition theorem 1.7 is easily proved.

Proof of Theorem 1.7.

For the first part, we need to show:

$$\frac{\lambda - \alpha_{\max}}{Z(\alpha_{\text{diff}})} - 1 \leq \ 23e^{\alpha_{\text{diff}}}\sqrt{\frac{\delta_{\max}}{\delta}\alpha_{\max}^3}.$$

Leaving D unspecified for now, let us abbreviate $\lambda(WJ[D, \alpha_R, \alpha_L])$ by $\lambda(WJ)$. Then, the left hand side may be written and upper bounded as follows:

$$\frac{\lambda - \lambda(WJ)}{Z(\alpha_{\text{diff}})} + \frac{\lambda(WJ) - \alpha_{\max}}{Z(\alpha_{\text{diff}})} - 1 \leq |\frac{\lambda - \lambda(WJ)}{Z(\alpha_{\text{diff}})}| + |\frac{\lambda(WJ) - \alpha_{\max}}{Z(\alpha_{\text{diff}})} - 1|$$

We bound the two terms in the sum separately. Corollary 4.3 implies that, as long as D is chosen larger than $\alpha_{\max}^2$, the second term is at most $4\alpha_{\max}^2/D$. The first term is bounded using lemma 5.1, the trivial upper bound $\lambda(WJ) \leq 2\alpha_{\max} + 1 \leq 3\alpha_{\max}$ observed early in section 4, and the fact that $Z(x) \geq e^{-x}$ for all nonnegative x:

$$\begin{aligned}
|\frac{\lambda - \lambda(WJ)}{Z(\alpha_{\text{diff}})}| &\leq \frac{3\alpha_{\max}\eta}{Z(\alpha_{\text{diff}})} \\
&\leq \frac{3\alpha_{\max} \cdot 6 \cdot \delta_{\max} D}{Z(\alpha_{\text{diff}})\delta} \\
&\leq \frac{18\alpha_{\max}\delta_{\max} D e^{\alpha_{\text{diff}}}}{\delta}.
\end{aligned}$$

Now set D to be an integer satisfying $\sqrt{\delta\alpha_{\max}/\delta_{\max}} \leq D \leq \frac{19}{18}\sqrt{\delta\alpha_{\max}/\delta_{\max}}$. This is possible because the hypothesis of the theorem implies that $\sqrt{\delta\alpha_{\max}/\delta_{\max}} \geq 18$. It is easily verified that, under the hypothesis of Theorem 1.7, the resulting D satisfies both the hypothesis $D \geq \alpha_{\max}^2$ of Corollary 4.3 and the conditions in Lemma 5.1. Summing the upper bounds on the two terms using this value of D yields an upper bound of:

$$19e^{\alpha_{\text{diff}}}\sqrt{\frac{\alpha_{\max}^3\delta_{\max}}{\delta}} + 4\sqrt{\frac{\alpha_{\max}^3\delta_{\max}}{\delta}} \leq 23e^{\alpha_{\text{diff}}}\sqrt{\frac{\alpha_{\max}^3\delta_{\max}}{\delta}},$$

as required for the first part of the Theorem.

For the second part of the theorem, either $\lambda_{\max} \geq \alpha_{\max} + Z(\alpha_{\text{diff}})(1-\zeta)$, or else, we must prove:

$$\frac{\alpha_{\max} - \lambda}{Z(\alpha_{\text{diff}})} + 1 \leq \ 23 e^{\alpha_{\text{diff}}} \sqrt{\frac{\delta_{\max}}{\delta} \alpha_{\max}^3}.$$

The proof is similar to that of the first part. ∎

Thus it remains to prove Lemma 5.1. The proof of this follows the outline in Section 3, but needs a lot of technical work which is divided into three subsections. First, we state two technical results that bound the constants occuring in the definition of the competitive ratio; the proofs of these lemmas are deferred until the end of the paper. Then we prove the upper and lower bounds of the lemma.

5.1 Two technical lemmas

In the proof overview, we related a step to the left (resp. right) by the Walker in the Walker-Jumper game, to the addition of an s-block for some appropriately chosen s by the Pursuer in the Pursuit-Evasion game. In sketching how this works, we approximated the cost of such a left block to the Evader by $s\lambda_R$. When we formalize this argument one of the things we will have to do is to bound the error in this approximation. For this, we will need two lemmas concerning the existence of "good" strategies for the pursuer and the evader in the pursuit-evasion game.

The competitive ratio for a 2-person game was defined in general as an infimum over κ for which there is a κ-competitive algorithm. In general this infimum need not be attained, i.e., if the competitive ratio is λ, there need not be a λ competitive strategy for MIN. However, as we will see in the next lemma, for the pursuit-evasion game there is always a randomized evader strategy that attains the competitive ratio. Furthermore, we can also upper bound the constant K that occurs in the definition of the competitive ratio.

Lemma 5.2 *Let $\mathcal{M}$ be a metric space of diameter δ and let λ denote the competitive ratio of its pursuit-evasion problem. Then there exists an algorithm $\tilde{A}$ such that for any probe sequence ρ:*

$$C_{\tilde{A}}(\rho) \leq \lambda C_{\text{OPT}}(\rho) + \delta\lambda.$$

Corollary 2.1 provides a criterion for lower bounding the competitive ratio of any game. The sequence of strategies $\tilde{r}_i$ in the hypothesis of the corollary can be viewed as a *witness* to the fact that the competitive ratio is at least κ. To prove tight lower bounds on the competitive ratio, we would like to be able to find such a witness sequence in the case $c = \lambda$.

Lemma 5.3 *Let $\mathcal{M}$ be a metric space and λ the competitive ratio of its pursuit-evasion game. For any $s > 0$ there is a distribution $\tilde{\rho}_{\mathbf{s}}$ on s-blocks such that for any response algorithm A:*

$$C_A(\tilde{\rho}_{\mathbf{s}}) \geq \lambda(s - \delta) - \delta.$$

The proofs require a somewhat tedious technical formulation, after which the results follow from elementary analysis. So as not to distract from the flow of the main argument we defer the proof to the last section of the paper.

5.2 The Upper Bound (proof of Lemma 5.1, Part 1)

We now proceed with the proof of Lemma 5.1. We assume for now that both the left and right space have at least two points. In the case that one or both of them is degenerate (has only one point), the proof is similar but requires some technical modifications which we indicate at the end of the subsection.

The upper bound is proved by associating for each Jumper strategy $\tilde{J}$ for $WJ[D, \alpha_R, \alpha_L]$, an Evader algorithm $\tilde{A}(\tilde{J})$ for $PE(\mathcal{M})$ that satisfies that if $\tilde{J}$ is κ-competitive then $\tilde{A}(\tilde{J})$ is $\kappa(1+\eta)$-competitive. The evader algorithm $\tilde{A}(\tilde{J})$ must specify an online rule for responding to a probe sequence. The idea will be to associate the incoming probe sequence to a Walker sequence, apply $\tilde{J}$ to that Walker sequence and then translate the Jumper's moves into moves for the Evader.

We first describe a rule for associating a probe sequence ρ for $\mathcal{M}$ to a Walker sequence $\mathbf{w} = \mathbf{w}(\rho)$. View ρ as the interleaving of two probe sequences, one for $\mathcal{M}_R$ (the *right* subsequence) and one for $\mathcal{M}_L$ (the *left* subsequence). Define the parameter $s_2 = (\delta - 2\delta_{\max})/D$ (the reader should think of this as approximating δ/D). Parse each of these subsequences separately into its s_2-block partition, as defined in Section 2.3. Now build $\mathbf{w}$ in the following online manner. Each time a right (resp. left) block of ρ is completed, a right (resp., left) Walker step is added to $\mathbf{w}$ unless that step will take the Walker outside the bounds $[-D, D]$. Let $\rho^{\mathbf{i}}$ denote the prefix corresponding to the steps up to w_i. Let $k = |\mathbf{w}|$ and let μ be the portion of ρ coming after the last s_2-block of either subsequence; thus $\rho = \rho^{\mathbf{k}}\mu$.

Given the Jumper strategy $\tilde{J}$, the algorithm $\tilde{A}(\tilde{J})$ responds to the probe sequence ρ as follows. As the sequence ρ is received, the Evader constructs $\mathbf{w}(\rho)$ and simulates the response of $\tilde{J}$ to this sequence. The Evader uses the position (left or right) of the simulated Jumper to determine which space to be in; the response by $\tilde{J}$ to a step of $\mathbf{w}$ determines whether the Evader continues in the same space or moves to the other. The times at which the Evader switches spaces delimits a sequence of *left* and *right* intervals. During a right(resp. left) interval, the Evader ignores probes on the other side and responds to the subsequence of right (resp.,left) probes by applying the algorithm $\tilde{A}_R$(resp., $\tilde{A}_L$), where $\tilde{A}_R$ (resp., $\tilde{A}_L$) is the Evader strategy for $PE(\mathcal{M}_R)$ (resp., $PE(\mathcal{M}_L)$) that satisfies the conclusion of Lemma 5.2.

Now, supposing that $\tilde{J}$ is κ-competitive, we need to verify that $\tilde{A}(\tilde{J})$ is $\kappa(1+\eta)$-competitive. We will deduce this from:

Lemma 5.4 *Let $\tilde{J}$ be a Jumper strategy for $WJ[D, \alpha_R, \alpha_L]$. Then for any probe sequence ρ for $PE(\mathcal{M})$,*

1. $C_{\text{OPT}}(\rho) \geq (s_2 - \delta_{\max})C_{\text{BASE}}(\mathbf{w}(\rho)) - \delta$,
2. $C_{\tilde{A}(\tilde{J})}(\rho) \leq (s_2 + 2\delta_{\max})C_{\tilde{J}}(\mathbf{w}(\rho)) + K$, *for some constant K independent of ρ.*

Using this lemma and the assumption that $\tilde{J}$ is κ-competitive we obtain that for some constants H and H' independent of ρ:

$$\begin{aligned}
C_{\tilde{A}(\tilde{J})}(\rho) &\leq (s_2 + 2\delta_{\max})C_{\tilde{J}}(\mathbf{w}(\rho)) + K \\
&\leq (s_2 + 2\delta_{\max})\kappa C_{\text{BASE}}(\mathbf{w}(\rho)) + H \\
&\leq \frac{s_2 + 2\delta_{\max}}{s_2 - \delta_{\max}}\kappa C_{\text{OPT}}(\rho) + H' \\
&\leq (1 + \frac{3D\delta_{\max}}{\delta - (D+2)\delta_{\max}})\kappa C_{\text{OPT}}(\rho) + H' \\
&\leq (1 + \frac{6D\delta_{\max}}{\delta})\kappa C_{\text{OPT}}(\rho) + H',
\end{aligned}$$

where the last inequality follows from $\delta \geq 2(D+2)\delta_{\max}$ which is an easy consequence of the hypotheses of the lemma. This implies that $\tilde{A}(\tilde{J})$ is $(1+\eta)\kappa$-competitive, as required to prove the upper bound.

So it remains to prove Lemma 5.4.

Proof of Lemma 5.4. Define $C_L(i)$ (resp., $C_R(i)$), for $1 \leq i \leq k$, to be the minimum cost of a response sequence to $\rho^{\mathbf{i}}$ whose last point is in $\mathcal{M}_L$ (resp. $\mathcal{M}_R$).

Note that $C_{\mathrm{OPT}}(\rho) \geq C_{\mathrm{OPT}}(\rho^{\mathbf{k}}) = \min\{C_R(k), C_L(k)\}$. Since $|C_L(i) - C_R(i)| \leq \delta$, we have $C_{\mathrm{OPT}}(\rho) \geq C_R(k) - \delta$.

Lemma 5.5 *$C_R(i) \geq (s_2 - \delta_{\max})(i + w_i)/2$ and $C_L(i) \geq (s_2 - \delta_{\max})(i - w_i)/2$.*

Since $C_{\mathrm{BASE}}(\mathbf{w}) = (k + w_k)/2$, the first part of Lemma 5.4 follows.

Proof: We prove this by induction on i; the basis $i = 0$ is trivial. Suppose $i > 0$. Assume that $(\Delta w)_i = +1$; the case $(\Delta w)_i = -1$ is proved analogously. Thus $\rho^{\mathbf{i}}$ marks the end of a right block.

The induction step for $C_L(i)$ follows from $C_L(i) \geq C_L(i-1)$. For $C_R(i)$, let $j < i$ be the least index such that $(\Delta w)_h = -1$ for $j < h < i$. Thus either $j = 0$ and w_i is the first step to the right, or w_j is the last step to the right prior to w_i. It is easy to see that this implies that $i + w_i = j + w_j + 2$, and hence $j \geq i + w_i - D - 2$.

We need to show that any response sequence for $\rho^{\mathbf{i}}$ that ends on the right costs at least $(s_2 - \delta_{\max})(w_i + i)/2$. Consider such a response sequence, written as $\sigma\tau$ where σ is the portion that is a response sequence to $\rho^{\mathbf{j}}$.

Note that $d(\sigma)$ must be at least the minimum of $C_L(j)$ and $C_R(j)$ which, by induction and the above expression for j is at least $(s_2 - \delta_{\max})(j - D)/2 \geq (s_2 - \delta_{\max})(i + w_i - 2D - 2)/2$. Any move between the two spaces costs at least $\delta - 2\delta_{\max} = Ds_2$, which can be shown to be larger than $(s_2 - \delta_{\max})(D+1)$ using the hypotheses $D \geq \sqrt{\delta/\delta_{\max}}$ of the theorem. Thus it follows that if there is any move between spaces subsequent to σ, then the total cost of the response sequence is at least $(s_2 - \delta_{\max})(w_i + i)/2$, as required.

So assume that after σ there is no move between spaces. Then σ ends on the right and all steps of τ are on the right. Since the portion of ρ between $\rho^{\mathbf{j}}$ and $\rho^{\mathbf{i}}$ contains a right s_2-block, it follows that τ has optimal cost at least $s_2 - \delta_{\max}$. Thus:

$$
\begin{aligned}
d(\sigma\tau) &\geq (s_2 - \delta_{\max})(w_j + j)/2 + s_2 - \delta_{\max} \\
&= (s_2 - \delta_{\max})(w_j + j + 2)/2 \\
&= (s_2 - \delta_{\max})(w_i + i)/2,
\end{aligned}
$$

as required. ∎

We now turn to the proof of the second part of Lemma 5.4. For this we want to upper bound the expected cost incurred by $\tilde{A}(\tilde{J})$ on ρ, in terms of the expected cost of $\tilde{J}$ on $\mathbf{w}(\rho)$. Consider the cost incurred by the Evader during each interval that begins after a move into one space and ends with the move out of that space, or, in the case of the last such interval, with the end of the sequence ρ. We will compare the Evader's cost to the cost of the Jumper during the corresponding interval. Without loss of generality, assume that the Evader is on the right during this interval and thus the Jumper is at $+D$. This implies that the simulated Walker can not be at $+D$ at any step (except possibly the last one) during this interval.

First consider the case that the interval is not the last such interval. The Evader responds only to the probes that occur on the right during that interval. In the definition of the algorithm, the subsequence of right probes was partitioned into right s_2-blocks. Let m be the number of s_2-blocks that end during this interval. By the definition of $\mathbf{w}$ and the fact that the Walker is not at $+D$ at any time during the interval, it follows that the Walker took m right steps during the interval. Thus the cost to the simulated Jumper is $m\alpha_R$ for those steps, plus D for the final jump.

Now, consider the cost to the Evader corresponding to the interval. Since the Evader is on the right, he responds only to the right probes that occur during the interval. The subsequence of right probes has optimal cost (with respect to the right space) at most $m(s_2 + \delta_R)$, since it is the concatenation of m sequences each with optimal cost at most s_2. Since the Evader uses algorithm $\tilde{A}_R$ to respond, the expected cost of responding is at most $\alpha_R m(s_2 + \delta_R) + \lambda_R\delta_R \leq \alpha_R m(s_2 + 2\delta_R)$. Adding the cost of the final move to the left space, which is at most δ, we obtain an upper bound on the Evader's cost for the interval of:

$$
\begin{aligned}
\alpha_R m(s_2 + 2\delta_R) + \delta &= \alpha_R m(s_2 + 2\delta_R) + Ds_2 + 2\delta_{\text{max}} \\
&\leq \alpha_R m(s_2 + 2\delta_R) + D(s_2 + 2\delta_{\text{max}}) \\
&\leq (\alpha_R m + D)(s_2 + 2\delta_{\text{max}})
\end{aligned}
$$

Thus the Evader's cost on every interval except the last is at most $(s_2 + 2\delta_{\text{max}})$ times the simulated Jumper's cost on the interval.

For the last interval, the cost to the Jumper is $\alpha_R m$. To bound the expected cost to the Evader we must include the portion μ of ρ which occurs after the last step by the simulated Walker. This could increase the optimal cost (with respect to the right space) of the subsequence of right probes in this interval to $(m+1)(s_2+\delta_R)$. Thus the expected cost to the Evader within the interval is bounded above by $\alpha_R(m+1)(s_2 + \delta_R) + \alpha_R\delta_R$, which is in turn bounded above $(s_2 + 2\delta_R)$ times the simulated Jumper's cost of $m\alpha_R$ plus a constant that does not depend on ρ. ∎

We now indicate how to modify the proof in the case that one or both of the spaces has exactly one point. First we have to modify the definition of the Walker sequence $\mathbf{w}(\rho)$ associated to a probe sequence. As before, we view ρ as the interleaving of a right and a left subsequence, and we parse each of these sequences into their s_2-block partitions. Recall that for one point spaces, we defined an s_2-block partition to be the partition into singleton blocks. As before, at the completion of a left or right block, the next step of the Walker is generated. For a block corresponding to a nondegenerate subspace, the Walker step is generated as before. A block corresponding to a degenerate subsequence, will correspond to not one, but a sequence of Walker steps which take the Walker all the way to the corresponding endpoint ($-D$ for a left block and D for a right block). If this is the i^{th} completed block, then we abuse notation by setting $w_i = D$ for a right block, and $w_i = -D$ for a left block; thus we compress all of the Walker steps corresponding to a degenerate block into one step.

Lemma 5.4 can be extended to hold in this case. The definitions of $C_L(i)$ and $C_R(i)$ need to be modified in Lemma 5.6: if the left (resp. right) space is degenerate and $\rho^{(\mathbf{i})}$ ends with a left (resp. right) block, then the definition of $C_L(i)$ (resp. $C_R(i)$) does not make sense and we modify it by defining $C_L(i) = C_R(i)+\delta$ (resp. $C_R(i) = C_L(i)+\delta$). The proof of Lemma 5.6 is then routine.

In the proof of the second part of Lemma 5.4 we analyze intervals defined by moves of the Evader from one space to the other. The case of an interval in which the Evader occupies a degenerate space is different than those analyzed, but is actually easier, since during the interval there are no requests inside the degenerate space and the cost to the Evader is the cost of the move to the other space at the end of the interval which is bounded above by δ. The cost to the Jumper during the same interval is just D.

5.3 The Lower Bound (proof of Lemma 5.1, Part 2)

As with the upper bound, we prove the lower bound in the case that $\mathcal{M}_L$ and $\mathcal{M}_R$ each have size at least two. Afterwards, we indicate how to modify the proof to handle the degenerate case that one or both spaces has only one point.

The strategy of the lower bound is the "mirror image" of the upper bound proof. That is, we will define a function which maps each Evader algorithm $\tilde{A}$ for $PE(\mathcal{M})$ to a Jumper algorithm $\tilde{J}(\tilde{A})$ for $WJ[D, \alpha_R, \alpha_L]$ and has the following property: if $\tilde{A}$ is a κ-competitive algorithm for the $PE(\mathcal{M})$ then $\tilde{J}(\tilde{A})$ is a $\frac{\kappa}{1-\eta}$-competitive algorithm for $WJ[D, \alpha_R, \alpha_L]$, where $\eta \leq 6\frac{\delta_{\max}D}{\delta}$. This immediately implies the lower bound of the lemma.

There are two main steps. Define the parameter $s_1 = \delta/D$.

1. For each Walker strategy $\mathbf{w}$ for the game $WJ[D, \alpha_R, \alpha_L]$, we define a distribution $\tilde{\mu}(\mathbf{w})$ on probe sequences for the Pursuer in the pursuit-evasion game on $\mathcal{M}$ such that
$$C_{\mathrm{BASE}}(\mathbf{w})(s_1 + \delta_{\max}) \geq C_{\mathrm{OPT}}(\tilde{\mu}(\mathbf{w})) \tag{4}$$

2. For each (randomized) Evader algorithm $\tilde{A}$ we define a Jumper algorithm $\tilde{J}(\tilde{A})$ for $WJ[D, \alpha_R, \alpha_L]$ and show that for all Evader algorithms $\tilde{A}$ and Walker strategies $\mathbf{w}$:
$$C_{\tilde{A}}(\tilde{\mu}(\mathbf{w})) \geq C_{\tilde{J}(\tilde{A})}(\mathbf{w})(s_1 - 2\delta_{\max}) \tag{5}$$
(where the first cost function is with respect to $PE(\mathcal{M})$ and the second is with respect to $WJ[D, \alpha_R, \alpha_L]$).

It follows immediately from these two steps that if $\tilde{A}$ is κ-competitive then there is a real number K such that for any Walker sequence $\mathbf{w}$,

$$(1 - 2\delta_{\max}D/\delta)(1 - \delta_{\max}D/\delta)C_{\tilde{J}(\tilde{A})} \leq \kappa C_{\mathrm{BASE}}(\mathbf{w}) + K$$

and thus $\tilde{J}(\tilde{A})$ is $\frac{\kappa}{(1-3\delta_{\max}D/\delta)} - competitive$, which will complete the proof.

For this proof, we define a *left block* (resp. *right block*) to be an s_1-block of the space $\mathcal{M}_L$ (resp. $\mathcal{M}_R$). For a Walker sequence $\mathbf{w} = (w_0 = 0, w_1, \ldots, w_k)$ in the game $WJ[D, \alpha_R, \alpha_L]$ we say that probe sequence ν for $\mathcal{M}$ is *compatible* with $\mathbf{w}$ if ν is the concatenation of k probe sequences $\nu_1\nu_2\ldots\nu_k$ satisfying:

1. For each i such that $|w_i| \neq D$, ν_i is a single right block if $(\Delta_w)_i = 1$ and is a single left block if $(\Delta_w)_i = -1$.

2. Let N be a large integer parameter to be specified later. If $|w_i| = D$ then ν_i is the concatenation of N right blocks if $(\Delta_w)_i = 1$ and is the concatenation of N left blocks if $(\Delta_w)_i = -1$.

The sequences ν_i are referred to as *segments* and are designated as left or right segments depending on whether they are from $\mathcal{M}_R$ or $\mathcal{M}_L$. A segment consisting of a single block is called a *small* segment, and one consisting of N blocks is called a *large segment.*

The distribution $\tilde{\mu}(\mathbf{w})$ will be a distribution over probe sequences compatible with $\mathbf{w}$. To define this distribution, we first need the following special case (actually, a slight weakening) of Lemma 5.3:

Corollary 5.1 *There exists a distribution $\tilde{\rho}_L$ on the set of left s_1-blocks and a distribution $\tilde{\rho}_R$ on the set of right s_1-blocks such that:*

1. *For any response algorithm A for $\mathcal{M}_L$: $C_A(\tilde{\rho}_L) \geq \alpha_L(s_1 - 2\delta_L)$.*
2. *For any response algorithm A for $\mathcal{M}_R$: $C_A(\tilde{\rho}_R) \geq \alpha_R(s_1 - 2\delta_R)$.*

The distribution $\tilde{\mu}(\mathbf{w})$ is now defined to be the distribution over probe sequences compatible with $\mathbf{w}$ in which each left block is chosen independently from the distribution $\tilde{\rho}_L$ and each right block is chosen independently from the distribution $\tilde{\rho}_R$.

Next, we relate the base-cost of $\mathbf{w}$ to the expected optimal cost $\tilde{\mu}(\mathbf{w})$. The base-cost of $\mathbf{w}$ equals the number of right steps, which is equal to $(k+w_k)/2$ where $k = |\mathbf{w}|$. We will show that for any probe sequence ν compatible with $\mathbf{w}$, its optimal cost is at most $(s_1 + \delta_{max})(k + w_k)/2$. Since $\tilde{\mu}(\mathbf{w})$ is a distribution over such sequences, this will imply inequality (4).

So fix ν compatible with $\mathbf{w}$. Denote by $C_L(i)$ (resp. $C_R(i)$) the minimum cost of a response sequence for the first i segments of ν whose last point is in $\mathcal{M}_L$ (resp. $\mathcal{M}_R$). Note that $|C_L(i) - C_R(i)| \leq \delta$. Then $C_{\mathrm{OPT}}(\nu)$ is then equal to the minimum of $C_L(k)$ and $C_R(k)$ and thus the desired upper bound on $C_{\mathrm{OPT}}(\nu)$ is an immediate consequence of:

Lemma 5.6 *$C_R(i) \leq (s_1 + \delta_{\max})(i + w_i)/2$ and $C_L(i) \leq (s_1 + \delta_{\max})(i - w_i)/2$.*

Proof: We prove this by induction on i; the basis case $i = 0$ is trivial. Now suppose that $i > 0$ and that the result holds for $i - 1$.

We assume $(\Delta w)_i = +1$; the case $(\Delta w)_i = -1$ is proved analogously. Thus ν_i consists of one or N s_1-blocks from $\mathcal{M}_R$.

To prove the induction step for $C_L(i)$ it suffices to observe that $C_L(i) = C_L(i-1)$, which is obvious since the definition of $C_L(i-1)$ implies that there is a response sequence for $\nu_{\mathbf{1}} \ldots \nu_{\mathbf{i-1}}$ that ends at some point $y \in \mathcal{M}_L$ that costs $C_L(i-1)$. This sequence can be extended by remaining at y through $\nu_{\mathbf{i}}$ and the cost does not increase.

Next we prove the induction step for $C_R(i)$. In the case that segment ν_i consists of a single right block then by the definition of $C_R(i-1)$ there is a response sequence σ for $\nu_1 \ldots \nu_{i-1}$ ending at some point $y \in \mathcal{M}_R$ that costs at most $C_R(i-1)$. Since ν_i is an s_1-block for $\mathcal{M}_R$, there is a response sequence τ for ν_i that costs at most s_1 and ends in $\mathcal{M}_R$. Then the sequence $\sigma\tau$ costs at most $C_R(i-1)+s_1+d(y,\tau_1) \leq C_R(i-1)+s_1+\delta_{max}$ which is bounded by $(s_1+\delta_{max})(i+w_i)/2$ by the induction hypothesis.

In the case that the segment ν_i consists of N right blocks, we must have $w_i = D$. Then we use the fact that $C_R(i) \leq C_L(i) + \delta$ to obtain $C_R(i) \leq (s_1+\delta_{\max})(i-D)/2 + s_1 D \leq (s_1+\delta_{\max})(i+D)/2$ as required. ∎

Now we turn to the second and final step: the definition of $\tilde{J}(\tilde{A})$ and the verification of inequality (5). Given an Evader strategy $\tilde{A}$, the Jumper strategy $\tilde{J}(\tilde{A})$ will be defined as follows. On being presented Walker sequence $\mathbf{w}$, J generates a probe sequence according to the distribution $\tilde{\mu}(\mathbf{w})$. Note that this generation can be done online with segment i being generated given w_i. At the same time, he simulates the algorithm A on the sequence $\tilde{\mu}(\mathbf{w})$. At step i, the Jumper responds to w_i as follows:

1. if $|w_i| = D$ then $j_i = -w_i$ (which is forced by the rules of the game).
2. if $|w_i| < D$ and $\mathbf{w}_i$ is in the direction opposite to j_{i-1} (i.e., they have opposite sign) then the Jumper does not move, i.e., $j_i = j_{i-1}$.
3. if $|w_i| < D$ and $\mathbf{w}_i$ is in the direction towards j_{i-1} then $j_i = D$ if $\tilde{A}$ is in $\mathcal{M}_R$ at the end of segment i and $j_i = -D$ otherwise.

It remains to verify inequality (5) and it suffices to verify this inequality for deterministic algorithms A; the result for randomized algorithms will follows by taking expectation with respect to the distribution over algorithms. So fix a deterministic algorithm A.

We will say that the simulated Evader and the Jumper are *synchronized at step i* if after the i^{th} step of the Jumper the Jumper is at $+D$ and the Evader is either in the left space or the Jumper is at $-D$ and the Evader is in the right space.

For technical reasons, it will be useful to modify the cost to the Jumper by adding D to his cost if at the last step he is not synchronized with the Evader. Obviously this modified cost is an upper bound on the true cost, so it suffices to work with this modified cost. What we will show is that the expected change in the cost incurred by the Evader during segment i is at least $(s_1 - 2\delta_{\max})$ times the expected change in the modified cost incurred by the Jumper at step i. We assume that the Jumper is on the right, $j_{i-1} = D$; the other case is handled similarly.

If step i of the Walker is to the left than the change in the Jumper's actual cost will be 0. His modified cost will go up by D if and only if the Evader was in $\mathcal{M}_R$ and moved to $\mathcal{M}_L$ during segment i. In this case, the Evader's cost increased by at least $\delta - 2\delta_{max}$ which is at least $(s_1 - 2\delta_{max})D$.

Now consider the case that the i^{th} step of the Walker is to the right. If the Jumper and Evader were not synchronized at step $i-1$, then at the end of step i, there are three possibilities: they both end on the left, they both end on the right, or they switch places (this can happen only if $w_i = D$ so that the Jumper is forced to jump). In the first two cases, the expected change in the modified cost of the Jumper is less than or equal to 0, while the Evader always incurs a nonnegative cost, and the desired inequality is trivial. In the third case, the Jumper's modified cost increases by D, while the Evader's cost increases by at least $\delta - 2\delta_{\max}$ which is at least $(s_1 - 2\delta_{max})D$.

So we may assume that the Jumper and Evader are both on the right after step $i-1$ and the next step is to the right. As the Evader is following a lazy algorithm, we may assume that if the Evader moves left during the block, he does not move again during the block.

We consider separately the cases that $w_i < D$ and $w_i = D$. In the case that $w_i < D$, then by the definition of the Jumper's strategy, they will still be synchronized at step i. So either they both stayed on the right, or they both moved to the left. If they both end on the left then the Jumper's modified cost increases by D and the Evader's cost increases by at least $\delta - 2\delta_{\max}$ which we have already noted is good for us. If they both stay on the right then the Jumper incurs a cost of α_R and we'd like to say that the Evader incurs an expected cost of at least $(s_1 - 2\delta_R)\alpha_R$. This would seem to be true: the Pursuer chooses her s_1-block from the distribution $\tilde{\rho}_{\mathbf{R}}$, and we know from Corollary 5.1 that any algorithm B incurs expected cost at least $(s_1 - 2\delta_R)\alpha_R$ against this distribution.

However, there is a subtle flaw in this reasoning. The algorithm A does not have to decide whether to move to the left at the beginning of the s_1-block; it can start on the right and at some point decide to move left. For example, suppose A stays on the right as long as the cost he has incurred during that block is less than some value V. The the conditional expectation of the cost incurred given that the algorithm finishes the block on the right is at most V. Since V can be chosen less than $(s_1 - 2\delta_R)\alpha_R$, this demonstrates that the above argument is fallacious.

To argue correctly, we must consider that, conditioned on the probe sequence up to the beginning of the current block, there is a probability p that A stays on the right for the entire block. Thus, the expected increase in the Jumper's modified cost is $p\alpha_R + (1-p)D$. We need to show that the expected increase in the Evader's cost is at least $(s_1 - 2\delta_R)$ times this. The expected increase in the Evader's cost is $(1-p)(\delta - 2\delta_{\max})$ plus the expected cost incurred in responding to probes while on the right. Now, the key point is that when computing this expected cost, not only must we consider the cost incurred when the Evader finishes on the right, but also, in the case that the Evader finishes on the left, we must consider the cost incurred by the Evader before moving to the left.

So let us consider the behavior of the algorithm A from the beginning of this s_1-block until the point that it jumps to the left. We can think of this algorithm as one for the game $PE(\mathcal{M}_R)$ which has the the additional option of *stopping* the game in the middle. Let us call such an algorithm a *stopping algorithm.*

Lemma 5.7 *Let B be a stopping algorithm for $PE(\mathcal{M}_R)$. Let p be the probability that B does not stop on input distribution $\tilde{\rho}_R$. Let q denote the expected cost incurred by B before it stops. Then $q \geq p\alpha_R(s_1 - 2\delta_R) - (1-p)\delta_R(3\lambda_R + 1)$.*

Proof: Define the (non-stopping) algorithm $\tilde{B}'$ for $PE(\mathcal{M}_R)$ as follows: respond using B until B stops. Then switch to using the algorithm $\tilde{A}_R$, the algorithm that satisfies the conclusion of Lemma 5.2.

Now, we upper bound $C_{\tilde{B}'}(\tilde{\boldsymbol{\rho}}_{\mathbf{R}})$. This cost is at most q plus the cost incurred after switching to $\tilde{A}_R$. The probability of ever switching to $\tilde{A}_R$ is $1-p$, and conditioned on switching, $\tilde{A}_R$ incurs a cost of at most δ_R for its first move and an expected cost of at most $\lambda_R(s_1 + \delta_R)$ for the rest of its moves since it is responding to an s_1 block (or a subsequence of it). Thus $C_{\tilde{B}'}(\tilde{\rho}_R) \leq q + (1-p)(\lambda_R(s_1 + \delta_R) + \delta_R)$. On the other hand, by Corollary 5.1, $C_{\tilde{B}'}(\tilde{\rho}_R) \geq \alpha_R(s_1 - 2\delta_R)$. We conclude, therefore, that $q \geq p\alpha_R(s_1 - 2\delta_R) - (1-p)(3\lambda_R + 1)(\delta_R)$. ∎

With Lemma 5.7 in hand, we now can lower bound the expected cost to the Evader in responding to the s_1-block by:

$$\begin{aligned} p\alpha_R(s_1 - 2\delta_R) - (1-p)\delta_R(3\lambda_R + 1) + (1-p)(\delta - 2\delta_{\max}) &\geq \\ p\alpha_R(s_1 - 2\delta_R) + (1-p)(\delta - (3\lambda_R + 3)\delta_{\max}) &\geq \\ (s_1 - 2\delta_{\max})(p\alpha_R) + (1-p)D(s_1 - 2\delta_{\max}), & \end{aligned}$$

where the last inequality is obtained by applying the hypotheses about D and s_1. The final term is $(s_1 - 2\delta_{\max})$ times the change in the Jumper's modified cost, as needed.

Finally, we consider the case that both the Jumper and Evader begin on the right, $w_{i-1} = D - 1$ and the step by the Walker is to the right. As before, since the Evader is following a lazy algorithm, either he stays on the right or he moves to the left at some time and stays there. Let p be the probability that the Evader stays on the right. The idea is that, since the segment corresponding to the last step of the Walker consists of N right blocks, where N is a large integer, we want to conclude that either p is extremely small, or the Evader incurs a very large cost. For $1 \leq j \leq N$, define p_j to be the probability that the Evader is still on the right after j of the s_1-blocks; thus, for all j, $p_j \geq p_N = p$. Note that the conditional probability that he is on the right after block j given that he is on the right after block $j-1$ is p_{j+1}/p_j. Then, by lemma 5.7, the expected cost incurred due to responding to requests on the right during block j is at least $p_j(\alpha_R(s_1 - 2\delta_R)) - (p_{j-1} - p_j)\delta_R(3\lambda_R + 1)$.

Summing this over j, we find that the expected cost incurred due to responses on the right is at least $pN\alpha_R(s_1 - 2\delta_R) - (1-p)\delta_R(3\lambda_R + 1)$. Adding in the expected cost of the final move to the left, and simplifying we get a lower bound on the Evader's cost of the form $(1-p)D(s_1 - 2\delta_{\max}) + pNT$ where T is a positive real number. We need this to be at least $(s_1 - 2\delta_{\max})$ times the expected change in the Jumper's modified cost, which is $D + pD$ since he must move to the left. For this we need pNT to be at least $2pD(s_1 - 2\delta_{\max})$ and this will hold as long as N was chosen large enough. ∎

Finally, let us indicate how the above proof changes if one of the spaces, say $\mathcal{M}_R$ has exactly one point p. The only change in the definitions of $\mathbf{w}(\rho)$ and $\tilde{J}(\tilde{A})$ comes from a change in the definition of compatibility. The modification is that for a step to the right $(\Delta w)_i = 1$, the segment $\nu_{\mathbf{i}}$ is defined to be empty if $w_i < D$ and is equal to the single point p if $w_i = D$. The analysis of this strategy involves similar considerations to the given proof, and is left to the reader.

6 Applications of the Decomposition Theorem

In this section we use Theorem 1.7 to prove Theorems 1.3 and 1.5. (Theorem 1.4 was already proved in the introduction).

Recall that a subspace $\mathcal{N}$ of space $\mathcal{M}$ is ϵ-small relative to $\mathcal{M}$ if $\delta(\mathcal{N}) \leq \epsilon\delta(\mathcal{M})$. We also say that $\mathcal{M}$ is *ϵ-uniform* if the distance between any two points in $\mathcal{M}$ is at least $\epsilon\delta(\mathcal{M})$. The following easy consequence of Theorem 1.1 is left to the reader (see also [KRR]).

Lemma 6.1 *The competitive ratio of an* ϵ-uniform *space on* n *points is between* $\epsilon \ln n$ *and* $(2/\epsilon)\ln n$*.*

It will be convenient to state three special cases of Theorem 1.7.

Corollary 6.1 *Let* $\mathcal{M}$ *be a metric space of at least three points and let* $\mathcal{M}_R, \mathcal{M}_L$ *be a partition. Let* α_R *and* α_L *be nonnegative numbers with* $\alpha_{\max}$ *their maximum and* α_{diff} *their absolute difference. Suppose that* $\alpha_{\max} \geq 1$ *and that both spaces are* $(e^{2\alpha_{\text{diff}}}\alpha_{\max}^3 2200)^{-1}$*-small in* $\mathcal{M}$*.*

1. *If* $\alpha_L \geq \lambda_L$ *and* $\alpha_R \geq \lambda_R$ *then* $\lambda \leq \alpha_{\max} + 3Z(\alpha_{\text{diff}})/2$*,*
2. *If* $\alpha_L \leq \lambda_L$ *and* $\alpha_R \leq \lambda_R$ *then* $\lambda \geq \alpha_{\max} + Z(\alpha_{\text{diff}})/2$*.*

Proof: Apply the bounds of Theorem 1.7, noting that the hypothesis of the Corollary guarantees $\zeta \leq 1/2$. ∎

The special case where $\alpha_R = \alpha_L$ is worth noting:

Corollary 6.2 *Let* $\mathcal{M}$ *be a metric space and let* $\mathcal{M}_R, \mathcal{M}_L$ *be a partition into two subspaces of size at least 2. Let* $\beta \geq 1$ *be a lower bound on both* λ_R *and* λ_L *and suppose that both spaces* $\mathcal{M}_R$ *and* $\mathcal{M}_L$ *are* $(2200\beta^3)^{-1}$*-small. Then* $\lambda \geq \beta + 1/2$*.*

We also have:

Corollary 6.3 *Let* $\mathcal{M}$ *be a metric space of at least three points and let* $\mathcal{N}$ *be a subspace. Let* $\beta \geq 1$ *be a lower bound on* $\lambda(\mathcal{N})$ *and suppose that* $\mathcal{N}$ *is* $(27000e^{3\beta})^{-1}$*-small. Then* $\lambda(\mathcal{M}) \geq \beta + e^{-2\beta}$*.*

Proof: Let x, y be points of $\mathcal{M}$ of distance δ. Then by the triangle inequality, one of the spaces obtained by adding exactly one of x and y to $\mathcal{N}$ has diameter at least $\delta/2$. Assume x is the point and let $\mathcal{K}$ be the union of $\mathcal{K}_R = \mathcal{N}$ and $\mathcal{K}_L = \{x\}$. The hypothesis of the Corollary guarantees that the ratio of $\delta(\mathcal{N})$ to $\delta(\mathcal{K})$ is at most $\frac{2e^{-3\beta}}{27000}$ which is less than or equal $\frac{\beta^{-3}e^{-2\beta}}{2200}$ for any $\beta \geq 1$. Thus we may apply Theorem 1.7 to the space $\mathcal{K}$. with $\alpha_R = \beta$ and $\alpha_L = 0$. We obtain a lower bound on the competitive ratio of $\mathcal{K}$ of $\beta + Z(\beta)(1-\zeta)$. Finally, note that $Z(\beta) \geq 2e^{-2\beta}$ for $\beta \geq 1$ and that the hypothesis of the corollary guarantees that $\zeta \leq 1/2$. ∎

6.1 Tight bounds for highly unbalanced spaces

Proof of Theorem 1.5.

We want to show that for some polynomial $p(n)$, any $p(n)$-unbalanced metric space has competitive ratio between $\ln n$ and $3 \ln n$. We prove only the upper bound here, the lower bound proof is very similar. For $n = 1, 2$ the result is trivial. Let $n > 2$ and let $\mathcal{M}$ be a $p(n)$-unbalanced metric space on n points. Let x and y be two points of distance δ, the diameter of the space. Then by the imbalance property, every other point z is close to exactly one of the points x or y, i.e., its distance to one of them is at most $\frac{\delta}{p(n)}$. This yields a $\frac{2}{p(n)}$-bipartition $(\mathcal{M}_L, \mathcal{M}_R)$. Let $n_L = |\mathcal{M}_L|$ and $n_R = |\mathcal{M}_R|$ and assume $n_L \geq n_R$. By induction, $\lambda(\mathcal{M}_L) \leq 3 \ln(n_L)$ and $\lambda(\mathcal{M}_R) \leq 3 \ln(n_R)$. Calling these upper bounds α_L and α_R, we have $\alpha_{\text{max}} = 3 \ln(n_L)$ and $\alpha_{\text{diff}} = 3 \ln(n_L/n_R)$. By choosing $p(n)$ to be a sufficiently large polynomial, the conditions of Corollary 6.1 apply, and $\lambda \leq 3 \ln(n_L) + (3/2) Z(3 \ln(n_L/n_R))$. Thus it suffices to show

$$3 \ln n \geq 3 \ln(n_L) + \frac{3}{2} Z(3 ln(n_L/n_R)).$$

For $n_L = n_R$, this reduces to $\ln 2 \geq 1/2$. For $n_L > n_R$, let $\rho = n_L/n_R - 1$, so that $\rho > 0$. Rewriting the desired inequality in terms of ρ, we need:

$$\ln(1 + \frac{1}{1+\rho}) \geq \frac{3 \ln(1+\rho)}{6\rho + 6\rho^2 + 2\rho^3}$$

Using $x \geq \ln(1+x) \geq x - x^2/2$ to lower bound the left hand side and upper bound the right hand side, it is enough to show:

$$\frac{1+2\rho}{2(1+\rho)^2} \geq \frac{3}{6 + 6\rho + 2\rho^2},$$

which is easily checked by cross-multiplying. ∎

6.2 A lower bound for all metric spaces

In order to prove Theorem 1.3 we will need a structure lemma for finite metric spaces, which says roughly that every finite metric space contains at least one of the following: (1) a small set of points whose removal reduces the diameter by a large fraction, or (2) a roughly uniform subspace of large size, or (3) a bipartite subspace in which both parts are large and have small diameter relative to their union.

Lemma 6.2 *Let $k \geq 0$ be an integer and $s \geq 1$. Every finite metric space $\mathcal{M}$ has a subspace satisfying at least one of the following conditions:*

1. *A 2^{1-s}-small subspace of size at least $(1-\frac{1}{s})|\mathcal{M}|$,*

2. *A $\frac{1}{4}$-uniform subspace of size at least s,*

3. *A 2^{1-k}-bipartite subspace, each of whose parts has size at least $\frac{|\mathcal{M}|}{2s^{k+2}}$.*

Proof: The first step in the proof is given by:

Proposition 6.1 *Let $k \geq 0$ be an integer and $s \geq 1$ be real. Any finite metric space $\mathcal{M}$ that has no $\frac{1}{4}$-uniform subspace of size at least s has a 2^{-k}-small subspace of size at least $\frac{|\mathcal{M}|}{s^k}$.*

Proof: We proceed by induction on k; the basis $k=0$ is trivial. Suppose $k>0$ and that $\mathcal{M}$ has no $\frac{1}{4}$-uniform subspace of size at least s. By the induction hypothesis there is a subspace $\mathcal{N}$ of size at least $\frac{|\mathcal{M}|}{s^{k-1}}$ and diameter at most $\delta 2^{1-k}$. Let $x_1, x_2, \ldots, x_t$ be a maximal sequence of points in $\mathcal{N}$ such that $d(x_i, x_j) \geq \delta(\mathcal{N})/4$ for $i \neq j$. For each $i \in \{1, \ldots, t\}$ let $X_i = \{y \in \mathcal{N} : d(y, x_i) < \delta(\mathcal{N})/4\}$. By the maximality of t, $\cup_i X_i = \mathcal{N}$. The largest X_i has size at least $|\mathcal{N}|/t \geq |\mathcal{M}|/s^k$ and has diameter at most $\delta(\mathcal{N})/2 \leq \delta(\mathcal{M})/2^k$. ∎

Proposition 6.2 *Let $k \geq 0$ be an integer, $s \geq 1$ and $\gamma \in (0,1)$. Any finite metric space $\mathcal{M}$ has at least one of the following:*

1. *a $1/2$-small subspace of size at least $(1-\gamma)|\mathcal{M}|$,*

2. *a $1/4$-uniform subspace of size at least s,*

3. *a 2^{1-k}-bipartite subspace, each of whose parts has size at least $\frac{\gamma}{s^k}|\mathcal{M}|$*

Proof: Assume that $\mathcal{M}$ has no $1/4$-uniform subspace of size at least s. Define sequences $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_u$ and $\mathcal{N}_1, \ldots, \mathcal{N}_u$ as follows. $\mathcal{M}_0 = \mathcal{M}$. Having constructed $\mathcal{M}_i$, if $|\mathcal{M}_i| < \gamma|\mathcal{M}|$ then stop and set $u = i$. Otherwise, the previous proposition implies that $\mathcal{M}_i$ has a 2^{-k}-small subspace of size at least $\frac{|\mathcal{M}_i|}{s^k} \geq \frac{\gamma}{s^k}|\mathcal{M}|$. Let $\mathcal{N}_{i+1}$ be such a subspace and let $\mathcal{M}_{i+1} = \mathcal{M}_i - \mathcal{N}_{i+1}$.

The union $\mathcal{N}$ of the $\mathcal{N}_i$ has size at least $(1-\gamma)|\mathcal{M}|$. If it is $1/2$-small then we have a space satisfying the first condition. Otherwise, there exist $x, y \in \mathcal{N}$ with $d(x,y) \geq \delta/2$. If $\mathcal{N}_i$ and $\mathcal{N}_j$ are the parts containing x and y respectively, then their union is a space satisfying the third condition. ∎

To complete the proof of Lemma 6.2, fix k, s as hypothesized. Let $\mathcal{M}$ be given and assume that $\mathcal{M}$ has no $\frac{1}{4}$-uniform subspace of size at least s and no 2^{1-k}-bipartite subspace in which each part has size at least $\frac{1}{2s^{k+2}}|\mathcal{M}|$. We show that $\mathcal{M}$ has a 2^{1-s}-small subset of size at least $(1-\frac{1}{s})|\mathcal{M}|$. This is trivial if $s = 1$ so assume $s > 1$ and set $\gamma = 1/s^2$. We define a sequence of metric spaces $\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{\lfloor s \rfloor}$, where $\mathcal{M}_i$ has size at least $(1-i\gamma)|\mathcal{M}|$ and has diameter at most $\delta 2^{-i}$. Then $\mathcal{M}_{\lfloor s \rfloor}$ has the desired properties.

To define the sequence, let $\mathcal{M}_0 = \mathcal{M}$. For $0 < i < \lfloor s \rfloor$, having defined $\mathcal{M}_i$ apply Proposition 6.2 to it. Our assumption about $\mathcal{M}$ implies that neither conclusion (2) nor conclusion (3) of Proposition 6.2 hold for $\mathcal{M}_i$ (for this we need to observe that since $i \leq \lfloor s \rfloor$, $|\mathcal{M}_i| \geq |\mathcal{M}|/2$.) $\mathcal{M}_i$ has a $\frac{1}{2}$-small subspace of size at least $(1-\gamma)|\mathcal{M}_i| \geq (1-(i+1)\gamma)|\mathcal{M}|$, which we take to be $\mathcal{M}_{i+1}$. ▮

Proof of Theorem 1.3.

Let $g(n)$ denote the minimum competitive ratio over all n point spaces. Because of the uniform space, $g(n) \leq 2\ln n$. We derive a recurrence inequality for $g(n)$. Let $\mathcal{M}$ be an n point space. Fix $s = s(n)$ and an integer $k = k(n)$ to be specified later and apply Lemma 6.2. If the second conclusion holds, then Lemma 6.1 implies that the $\lambda(\mathcal{M}) \geq \frac{1}{4} + \ln s$. If the third conclusion of Lemma 6.2 holds, we want to apply Corollary 6.2 with $\beta = g(\lceil \frac{n}{2s^{k+2}} \rceil)$ to conclude that $\lambda(\mathcal{M}) \geq g(\lceil \frac{n}{2s^{k+2}} \rceil) + 1/2$. To apply Corollary 6.2 it suffices that $2^{-k} \leq 1/(2200(2\ln n)^3)$ (using the fact that $g(n) \leq 2\ln n$)). So we choose $k = \lceil 20 + 3\log\ln n \rceil \leq 20 + 6\log\log n$. Finally, if the first conclusion of Lemma 6.2 holds, then we want to apply Corollary 6.3 with $\beta = g(\lceil n(1-\frac{1}{s}) \rceil)$ to conclude $\lambda(\mathcal{M}) \geq g(\lceil n(1-\frac{1}{s}) \rceil) + e^{-2g(\lceil n(1-\frac{1}{s}) \rceil)}$. To apply Corollary 6.3 it suffices that $2^{1-s} \leq 1/(27000e^{6\ln n})$ which holds if $s \geq 6\log n + 20$. Thus, under this assumption on s:

$$g(n) \geq \min\{\frac{\log s}{4}, g(\lceil n(1-\frac{1}{s}) \rceil) + e^{-2g(\lceil n(1-\frac{1}{s}) \rceil)}, g(\lceil \frac{n}{2s^{6\log\log n + 22}} \rceil) + \frac{1}{2}\} \qquad (6)$$

It now suffices to choose $s(n)$ satisfying the above condition, guess a function that lower bounds $g(n)$ and use the recurrence to verify the lower bound.

We choose $s(n) = 2^{C_0\sqrt{\log n/\log\log n}}$ where $C_0 > 0$ is a small constant to be chosen later. We choose $n_0 = n_0(C_0)$ so that for $n \geq n_0$, $s(n) \geq 6\log n + 20$. Finally set $h(n) = C_1 \log s(n)$, where $C_1 > 0$ is chosen to be at most $1/4$ and also small enough so that for $3 \leq n \leq n_0$, $h(n) \leq 1$.

We claim that $g(n) \geq h(n)$ for all $n \geq 3$, which suffices to prove the theorem. We proceed by induction on n.

For $n \le n_0$, the result is trivial since $g(n) \ge 1$. So assume $n > n_0$. Applying the recurrence (6) and the induction hypothesis we get:

$$g(n) \ge \min\{\frac{\log s}{4}, h(\lceil n(1-\frac{1}{s})\rceil) + e^{-2h(\lceil n(1-\frac{1}{s})\rceil)}, h(\lceil \frac{n}{2s^{6\log\log n+22}} \rceil) + \frac{1}{2}\}. \qquad (7)$$

In substituting h for g in the second term we observe that $x + e^{-2x}$ increases with x for $x \ge \frac{\ln 2}{2}$ and is at most 1 for $x \in [0, \frac{\ln 2}{2}]$. Next, since h increases with n we can drop the $\lceil \cdot \rceil$:

$$g(n) \ge \min\{\frac{\log s}{4}, h(n(1-\frac{1}{s})) + e^{-2h(n(1-\frac{1}{s}))}, h(\frac{n}{2s^{6\log\log n+22}}) + \frac{1}{2}\}. \qquad (8)$$

Now it suffices to show that each of the terms in the minimum is at least $h(n)$. This is true for the first term since $C_1 \le 1/4$. For the second and third terms, we bound $h(n) - h(n/B)$ for $B > 1$ by following chain of inequalities:

$$\begin{aligned} h(n) - h(n/B) &= C_1C_0\sqrt{\frac{\log n}{\log\log n}} - C_1C_0\sqrt{\frac{\log(n/B)}{\log\log(n/B)}} \\ &\le C_1C_0\sqrt{\frac{\log n}{\log\log n}}\left(1 - \sqrt{1 - \frac{\log B}{\log n}}\right) \\ &\le C_1C_0\sqrt{\frac{\log n}{\log\log n}}\frac{\log B}{\log n} \\ &= C_1C_0\frac{\log B}{\sqrt{\log n \log\log n}}. \end{aligned}$$

To show that the second term in (8) is at least $h(n)$ we take $B = s/(s-1)$ in the above inequality. The final expression is then at most $C_1C_0\frac{1}{s\sqrt{\log n\log\log n}}$. We need to show that this is at most $e^{-2h(n(1-1/s))}$. This is true, as $C_1C_0\frac{1}{s\sqrt{\log n\log\log n}} \le 1/s \le 1/s^{2C_1/\ln 2} = e^{-2h(n)} \le e^{-2h(n(1-1/s))}$, where the second inequality uses $C_1 \le 1/4$.

To show that the third term in (8) is at least $h(n)$ take $B = 2s^{6\log\log n+22}$ in the above inequality. The final expression in the inequality is then at most $C_1C_0^2(6\log\log n + 22)/\log\log n$ which is at most $1/2$ for sufficiently small C_0. ∎

7 Proof of Technical Lemmas

In this section, we present the proofs of Lemmas 5.2 and 5.3. Throughout this section $\mathcal{M}$ is a fixed finite metric space with distance function d and diameter δ. The minimum distance between (distinct) points in $\mathcal{M}$ is denoted $\delta_{\min}$.

We begin with some facts about the topological structure of the set of randomized evader algorithms.[1] Recall from Section 2.3 that an algorithm can be defined as a function that labels each E-node of $T_{\mathcal{M}}$ by a probability distribution on $\mathcal{M}$. The set $P(\mathcal{M})$ of probability distributions on $\mathcal{M}$ can be viewed as a topological subspace of Euclidean space $\mathbf{R}^{\mathcal{M}}$, and so the set of randomized algorithms can be viewed as a product of copies of this space (where the product is indexed by the E-nodes of $T_{\mathcal{M}}$).

The topological facts we need are summarized in:

Proposition 7.1 *1. Any sequence of randomized evader algorithms has a subsequence that converges.*

2. Suppose that the sequence $\tilde{A}_1, \tilde{A}_2, \ldots$ of algorithms converges to algorithm $\tilde{A}$ in the product topology. If ρ is any probe sequence then the sequence of real numbers $\tilde{A}_1(\rho), \tilde{A}_2(\rho), \ldots$ converges to $\tilde{A}(\rho)$.

Proof: The topological space of algorithms is equivalent to the product of a countable number of spaces isomorphic to $P(\mathcal{M})$. Notice that $P(\mathcal{M})$ is metrizable and compact. By Tychonoff's theorem (see [Kel][pp. 143–144]), the product of compact topological spaces is compact with respect to the product topology. Moreover, the product of a countable number of metrizable topological spaces is metrizable (see [Kel][p. 122]). The first part follows from the fact that every sequence in a compact metrizable topological space has a subsequence that converges to a point in the space (see [Kel][pp. 138–139]). For the second part, we note that for a probe sequence ρ, the mapping from the set of randomized algorithms to the reals, defined by $\tilde{A} \longrightarrow \tilde{A}(\rho)$ is continuous. ∎

We need a modification of the pursuit evasion game. Fix the metric space $\mathcal{M}$ and let λ denote the competitive ratio of the pursuit-evasion game. We define the *modified game* for $\mathcal{M}$ to be a game whose strategy sets are the same as for the pursuit-evasion game but whose cost function is $f_A(\rho) = C_A(\rho) - \lambda C_{\mathrm{OPT}}(\rho)$. For $s > 0$, let G_s (resp.

[1] In the following discussion we use the following concepts from set topology (see, e.g., [Kel]): A set system $\mathcal{F}$ is a *topology* iff the union of any number of elements of $\mathcal{F}$ is in $\mathcal{F}$ and the intersection of a finite number of elements of $\mathcal{F}$ is in $\mathcal{F}$. The pair $(X, \mathcal{F})$, where $X = \cup F \in \mathcal{F}$, is called a *topological space*. The elements of $\mathcal{F}$ are called *open sets*. If $Y \subset X$, then $(Y, \mathcal{G})$, where $\mathcal{G} = \{F \cap Y;\ F \in \mathcal{F}\}$, is a *topological subspace* of $(X, \mathcal{F})$. (Notice that a topological subspace is a topological space.) For a metric space $\mathcal{M} = (X, d)$, the associated *metric topology* $\mathcal{F}$ is derived by putting $\mathcal{F}$ to be the set of all the unions of open balls in $\mathcal{M}$. A topological space that can be derived this way is *metrizable*. Any subset of $\mathcal{F}$ whose union equals X is an *open cover* of the topological space $(X, \mathcal{F})$. A topological space is *compact* iff every open cover has a finite subset which is also an open cover. If $(X, \mathcal{F})$ and $(Y, \mathcal{G})$ are topological spaces, then a function $f : X \rightarrow Y$ is *continuous* iff for every $G \in \mathcal{G}$, $f^{-1}(G) \in \mathcal{F}$.

H_s) denote the game obtained from the modified game by restricting the strategies of the Pursuer to be probe sequences ρ satisfying $C_{\mathrm{OPT}}(\rho) \leq s$ (resp. satisfying $s - \delta \leq C_{\mathrm{OPT}}(\rho) \leq s$). We will prove:

Lemma 7.1 *For every $s > 0$,*

1. *The games G_s and H_s have the min-max property.*
2. *The values of G_s and H_s satisfy:*

$$-\delta \leq V(H_s) \leq V(G_s) \leq \lambda\delta$$

Assuming this lemma, it is easy to prove the two main technical lemmas.

Proof of Lemma 5.2: For $s > 0$, let $\tilde{A}_s$ denote an optimal strategy for G_s. By the first part of Proposition 7.1 there is a an infinite sequence $i_1 < i_2 < \ldots$ of positive integers such that the sequence $\tilde{A}_{i_j}$ converges to a limit algorithm $\tilde{A}$. If ρ is any probe sequence, and j is chosen such that $i_j \geq C_{\mathrm{OPT}}(\rho)$, the second part of Lemma 7.1 implies $C_{\tilde{A}_{i_j}}(\rho) - \lambda C_{\mathrm{OPT}}(\rho) \leq V(G_{i_j}) \leq \lambda\delta$. Letting i_j tend to ∞ yields $C_{\tilde{A}}(\rho) - \lambda C_{\mathrm{OPT}}(\rho) \leq \lambda\delta$, as required. ∎

Proof of Lemma 5.3: Fix $s > 0$ and let $\tilde{\rho}_s$ denote an optimal strategy for the Pursuer for the game H_s. Then for any Evader algorithm $\tilde{A}$, we have $C_{\tilde{A}}(\rho_s) - \lambda C_{\mathrm{OPT}}(\rho) \geq -\delta$, or $C_{\tilde{A}}(\rho_s) \geq \lambda(s - \delta) - \delta$, as required. ∎

The remainder of the section proves Lemma 7.1. To prove the first part of the Lemma, it would be enough, by Theorem 2.1, to show that the set of pure strategies is finite. Unfortunately, it is not in general true that the set of pure strategies is finite. However, we will reduce these games to an analysis of games with finite strategy sets. More precisely, we will define the notion of a *standard* probe sequence and a *standard* Evader algorithm, and show that the restrictions of the games G_s and H_s to these sets are essentially finite games, and that the the analysis of G_s and H_s can be reduced to that of the restricted game.

As a preliminary restriction, we say that an Evader algorithm A is *lazy* if it does not move unless it has to; i.e, the Evader moves from its current location only if the Pursuer probes the location he occupies. It is easy to show and well known (see [BE]) that the set of lazy algorithms dominates the set of all algorithms in the sense of Section 2.2. (Actually, the versions of this result that appear in the literature prove the dominance with respect to the cost function $C_A(\rho)$ rather than $f_A(\rho)$, but it is easy to show that the one result implies the other). Thus, applying Lemma 2.1 (2) it suffices to restrict the Evader to lazy algorithms, which we do from now on.

By the definition of a lazy algorithm, if the Evader is at point a after probe sequence ρ the Evader will move only if the next probe is to a. Thus for a lazy deterministic algorithm A, each probe sequence ρ induces a *transition function* $A[\rho]$ mapping the metric space to itself where $A[\rho](a)$ is the point the Evader moves to if it is at a after ρ and the next probe is to a.

Before we can define the notion of standard algorithm and standard probe sequence, we need to review some basic facts about the pursuit-evasion game and the function C_{OPT}. For the pursuit evasion game, C_{OPT} can be expressed as follows. For probe sequence ρ and point $p \in M$, let $C_A(\rho; p)$ denote the cost incurred by A in responding to ρ and then moving to point p; this is $C_A(\rho) + d(a, p)$ where a is the final point of $A(\rho)$. Let $C_{OPT}(\rho; p)$ denote the minimum of $C_A(\rho; p)$ over all algorithms A. Then $C_{\mathrm{OPT}}(\rho) = \min\{C_{\mathrm{OPT}}(\rho; p) : p \in \mathcal{M}\}$. For ρ equal to the empty string we have $C_{\mathrm{OPT}}(\rho; p) = 0$. For any ρ and any points a, p we have:

$$C_{\mathrm{OPT}}(\rho a; p) = \begin{cases} C_{\mathrm{OPT}}(\rho; p) & \text{if } a \neq p \\ \min_{q \neq p} C_{\mathrm{OPT}}(\rho; q) + d(q, a) & \text{if } a = p \end{cases}$$

For each ρ, $C_{\mathrm{OPT}}(\rho; p)$ is a function mapping $p \in \mathcal{M}$ to the nonnegative reals. This is the well known *work function* associated with this game (see, e.g., [BE]), and will be denoted $WF[\rho]$.

It is easy to check that if τ is an extension of ρ then $WF[\tau] \geq WF[\rho]$ where the inequality of work functions is defined pointwise. τ is a *null extension* of ρ if they have the same work function. A point a is said to be *null* with respect to a sequence ρ if ρa is a null extension of ρ. Let $N(\rho)$ denote the set of points that are null with respect to ρ.

We state a few easy facts without proof:

Proposition 7.2 *Let $\rho = (\rho_1, \ldots, \rho_k)$ be an arbitrary probe sequence.*

1. $\rho_k \in N(\rho)$
2. *If τ has the same work function as ρ then $N(\rho) = N(\tau)$. In particular, $a \in N(\rho)$ then $N(\rho a) = N(\rho)$.*
3. *If a is any point that minimizes $WF[\rho](\cdot)$ then $a \notin N(\rho)$. In particular, there is at least one point that is not null with respect to ρ.*

A sequence $\rho = (\rho_1, \ldots, \rho_k)$ is said to be *standard*, if the work functions associated to its prefixes are all different, equivalently, for each i between 2 and k $\rho_i \notin N(\rho_1, \ldots, \rho_{i-1})$.

A *standard* algorithm A is a lazy algorithm that ignores null points. More precisely, a standard algorithm never moves to a point that is null with respect to the current sequence, and ignores requests to null points in the sense that its present and future behavior is unaffected by such requests.

Let G'_s (resp. H'_s) denote the game obtained from G_s (resp. H_s) by restricting to standard probe sequences and standard Evader algorithms.

Proposition 7.3 *For each $s > 0$ there is an integer $t(s)$ with the property that any standard probe sequence of optimal cost at most s has term length at most $t(s)$.*

Proof:

Let $\rho = (\rho_1, \ldots, \rho_t)$ be a standard probe sequence. Let w^i denote the work function associated to the prefix ρ^i. For each i, $w^i(p) = w^{i-1}(p)$ if $p \neq \rho_i$ and $w^i(\rho_i) > w^{i-1}(\rho_{i-1})$. Let $\Delta_i = w^i(\rho_i) - w^{i-1}(\rho_{i-1})$. Induction on s yields $\sum_p w^s(p) = \sum_{i=1}^s \Delta_i$. Since the maximum and minimum work function values associated to any probe sequence differ by at most δ, $w^i(p) \geq \frac{1}{n}(\sum_{i=1}^s \Delta_i) - \delta$. Recall that $\delta_{\min}$ is the minimum distance between two distinct points in the metric space. We will show,

Claim. In any subsequence of n^n consecutive indices there is an i such that $\Delta_i \geq \delta_{\min}$.

This implies that for any ρ of length t, $w^i(\rho) \geq \lfloor \frac{t}{n^n} \rfloor \delta_{\min}$ and so for any $s > 0$ we can choose $t(s)$ so that for $|\rho| \geq t(s)$, the minimum work function value is at least s.

So we prove the claim. Fix an arbitrary ordering $<$ on $\mathcal{M}$. For each i, let $p_1^i, \ldots, p_n^i$ denote the sequence of points ordered so that if $w^i(p_r^i) \leq w^i(p_{r+1}^i)$ with ties broken according to the ordering $<$. Let Γ_i denote the directed graph on vertex set $\mathcal{M}$, with arcs $p \longrightarrow_i q$ if $w^i(q) = w^i(p) + d(p,q)$. (Note that a point p is null with respect to ρ^i precisely if its in-degree in Γ_i is positive.) Define $\mathbf{d}^i = (d_1^i, \ldots, d_n^i)$ with d_r^i equal to the outdegree of p_r^i in Γ_i. We will show that for each i, if $\Delta_i < \delta_{\min}$ then the sequence $\mathbf{d}^i$ is lexicographically greater than $\mathbf{d}^{i-1}$. Since there are at most n^n distinct out-degree sequences this will prove the claim and the proposition.

Γ_i differs from Γ_{i-1} only on the arcs incident on ρ_i. The in-degree of ρ_i in Γ_{i-1} is 0 since ρ_i is non-null with respect to ρ^{i-1}, and is positive in Γ_i since ρ_i is null with respect to ρ^i. Let h be an index such that $p_h^{i-1} \longrightarrow_i \rho_i$ and let j be the index such that $\rho_i = p_j^{i-1}$. By definition of Γ_i, $w^i(\rho_i) = w^i(p_h^{i-1}) + d(p_h^{i-1}, \rho_i) \geq w^{i-1}(p_h^{i-1}) + \delta_{\min}$.

If $h > j$ then $w^{i-1}(\rho_i) \leq w^{i-1}(p_h^{i-1})$ and we conclude $w^i(\rho_i) - w^{i-1}(\rho_i) \geq \delta_{\min}$.

So suppose $h < j$. For $r < j$, $p_r^i = p_r^{i-1}$ and all edges out of p_r^{i-1} in Γ_{i-1} are present in Γ_i. Thus $d_r^i \geq d_r^{i-1}$. For $r = h$ we have strict inequality since $d_h^i \longrightarrow_i \rho_i$. This implies that $\mathbf{d}^i$ is lexicographically greater than $\mathbf{d}^{i-1}$. ∎

An immediate consequence of Proposition 7.3 is that the number of standard probe sequences of cost at most s is finite. Also, although the number of deterministic Evader algorithms is infinite, if we call two algorithms equivalent if they respond identically to any probe sequence of term length at most $t(s)$, then the number of equivalence classes is finite. Hence G'_s (resp. H'_s) is essentially a finite game, and thus by Theorem 2.1 it has the min-max property. We will show that the game G_s (resp. H_s) is "essentially" the same as G'_s (resp. H'_s), to prove the first part of Lemma 7.1. For ease of notation, we refer only to G_s in this argument. The argument holds as well if we replace G_s by H_s and G'_s by H'_s.

For an Evader algorithm $\tilde{A}$ we write $V_{\mathrm{MIN}}(\tilde{A})$ for the value of $\tilde{A}$ in the game G_s, i.e., the supremum over all probe sequences ρ of optimal cost at most s of $f_{\tilde{A}}(\rho)$. We write $V'_{\mathrm{MIN}}(\tilde{A})$ for the value of $\tilde{A}$ in the game G'_s. Similarly for a randomized probe sequence $\tilde{\rho}$ we write $V_{\mathrm{MAX}}(\tilde{\rho})$ and $V'_{\mathrm{MAX}}(\tilde{\rho})$ respectively for the value of $\tilde{\rho}$ with respect to the games G_s and G'_s.

Proposition 7.4 *For any randomized standard Evader algorithm $\tilde{A}$, $V_{\mathrm{MIN}}(\tilde{A}) = V'_{\mathrm{MIN}}(\tilde{A})$.*

Proof: Let $\rho = (\rho_1, \ldots, \rho_k)$ be an arbitrary probe sequence. Write w^i for the work function associated to the prefix ρ^i. We observed earlier that $w^1 \leq w^2 \leq \ldots w^k$ (where the inequality is pointwise). Define the sequence of indices $i_1 < i_2 < \ldots < i_j$ where $i_1 = 1$, i_2 is the least i such that $w^{i_2} \neq w^{i_1}$ and, in general, i_h is the least i such that $w^{i_h} \neq w^{i_{h-1}}$. It is easy to see that the sequence $\hat{\rho} = \rho_{i_1}\rho_{i_2}\ldots\rho_{i_j}$ is standard and has the same work function as ρ.

Now let $\tilde{A}$ be a randomized standard algorithm. Since $\tilde{A}$ ignores all probes to null points, it behaves the same on ρ as it does on $\hat{\rho}$. More precisely, the probability that $\tilde{A}$ responds to $\hat{\rho}$ with the sequence $\sigma_1\sigma_2\ldots\sigma_j$ is equal to the probability that it responds to ρ with $\sigma_1^{i_2-1}\sigma_2^{i_3-i_2}\ldots\sigma_{j-1}^{i_j-i_{j-1}}\sigma_j^{n+1-i_j}$, which has the same cost as $\sigma_1\ldots\sigma_j$. ∎

Given an analogous result for Pursuer strategies, the first part of Lemma 7.1 would follow immediately. However, such a result is not true: if $\tilde{\rho}$ is a randomized standard Pursuer strategy, then since the Pursuer never probes null points, a non-standard algorithm might be able to avoid the Pursuer by moving to null points.

So we need to prove something a little more subtle about Pursuer strategies. The key property of a null point is that the Pursuer can probe a null point without increasing the work function, and thus probes to a null point are "free" to the Pursuer. Thus we can modify any standard Pursuer strategy so that it probes null points often enough to ensure that it does not benefit the Evader to ever visit a null point. If we modify the optimal standard Pursuer strategy in this way, we will get a strategy whose value in the game G_s is the same as that of the optimal standard strategy in the game G'_s.

We now make this argument precise. Let $\rho = (\rho_1, \ldots, \rho_k)$ be a probe sequence. For i between 1 and k, Let $\eta_{\mathbf{i}}$ be some fixed ordering on $N(\rho^i)$, the set of points that are null with respect to ρ^i. For a nonnegative integer j, we define the *j-fold inflation of ρ* to be the sequence $\rho_1 {\eta_{\mathbf{1}}}^j \rho_2 {\eta_{\mathbf{2}}}^j \ldots \rho_k {\eta_{\mathbf{k}}}^j$. It is trivial that the work functions associated with a sequence and its j-fold inflation are identical.

If $\tilde{\rho}$ is a randomized Pursuer strategy, i.e., a probability distribution over probe sequences, then the j-fold inflation of $\tilde{\rho}$ is the distribution obtained by selecting a sequence according to $\tilde{\rho}$ and applying j-fold inflation to it.

Proposition 7.5 *For j a sufficiently large integer (depending on $\mathcal{M}$ and s) the following holds. For any deterministic lazy Evader algorithm A there is a deterministic standard algorithm $\bar{A}$ such that for any randomized Pursuer strategy $\tilde{\rho}$ in the game G_s, if $\tilde{\sigma}$ is the j-fold inflation of ρ then $C_{\bar{A}}(\tilde{\rho}) \leq C_A(\tilde{\sigma})$. Consequently, $V'_{\mathrm{MAX}}(\tilde{\rho}) \leq V_{\mathrm{MAX}}(\tilde{\sigma})$*

Before proving the Proposition, let us see that it implies the first part of Lemma 7.1. Let $\tilde{A}^*$ and $\tilde{\rho}^*$ be the optimal strategies of Evader and Pursuer in the game G'_s, and let $\tilde{\sigma}^*$ be the j-fold inflation of $\tilde{\rho}^*$ (where j is large enough for Proposition 7.5). From Proposition 7.4, $V_{\mathrm{MIN}}(\tilde{A}^*) = V'_{\mathrm{MIN}}(\tilde{A}^*)$. From Proposition 7.5, $V'_{\mathrm{MAX}}(\tilde{\rho}^*) \leq V_{\mathrm{MAX}}(\tilde{\sigma}^*)$. Since G'_s has the min-max property, we have that $V'_{\mathrm{MIN}}(\tilde{A}^*) = V'_{\mathrm{MAX}}(\tilde{\rho}^*)$. Hence $V_{\mathrm{MIN}}(\tilde{A}^*) \leq V_{\mathrm{MAX}}(\tilde{\sigma}^*)$, and combining this with Lemma 2.2 we conclude that $V_{\mathrm{MIN}}(\tilde{A}^*) = V_{\mathrm{MAX}}(\tilde{\sigma}^*)$, i.e., G_s has the min-max property.

Proof: (of Proposition 7.5)

Fix j sufficiently large. Given A, we need to define $\bar{A}$. The behavior of A on a probe sequence $\rho = (\rho_1, \ldots, \rho_k)$ is defined using an online simulation of A applied to the j-fold inflation $\sigma = \rho_1 {\eta_{\mathbf{1}}}^j \ldots \rho_k {\eta_{\mathbf{k}}}^j$ of ρ. Prior to the i-th request, $\bar{A}$ will have processed $\rho_1, \ldots, \rho_{i-1}$ and responded with $\sigma_1 \sigma_2 \ldots \sigma_{i-1}$. It will also have simulated A on the j-fold inflation of $\rho_1, \ldots, \rho_{i-1}$. Upon receipt of ρ_i, it continues the simulation by giving A the sequence $\rho_i {\eta_{\mathbf{i}}}^j$. Let τ_i denote the final point A occupies after processing that sequence.

$\bar{A}$ then chooses its response σ_i according to the following rule. If $\sigma_{i-1} \neq \rho_i$ then

$\sigma_i = \sigma_{i-1}$ (following laziness). Otherwise, if τ_i is not null with respect to $\rho_1 \dots \rho_i$ then $\sigma_i = \tau_i$ else $\bar{A}$ moves to any non-null point (say the least one under some predetermined ordering).

It is immediate that $\bar{A}$ is a standard algorithm. We claim that $C_{\bar{A}}(\rho) \leq C_A(\sigma)$. This then extends by linearity to the case of randomized Pursuer strategies.

We say that A is *well-behaved* on $\rho_1, \rho_2, \dots, \rho_k$ provided that for each i, the point τ_i is not null with respect to $\rho_1 \dots \rho_i$. By the definition of $\bar{A}$ (and the fact that A itself is lazy), if A is well-behaved then the response sequence $\sigma_1, \dots, \sigma_k$ of $\bar{A}$ is just $\tau_1, \tau_2, \dots, \tau_k$ and is thus a subsequence of A's response sequence. Hence $C_{\bar{A}}(\rho) \leq C_A(\sigma)$.

If A is not well-behaved then for some i, τ_i is null with respect to $\rho_1 \dots \rho_i$. Consider the responses of A to $\rho_i \eta_i^j$. If A ever responded with a point that is not null with respect to $\rho_1 \dots \rho_i$ then since A is lazy and all requests appearing in η_i are null, A would have ended in a non-null point. So it must be that each of A's responses to this subsequence was a null point. Now, since η_i contains each null point once, A was forced to move at least j times. Thus $c_A(\sigma) \geq j\delta_{\min}$. Taking j large enough (depending on $\mathcal{M}$ and s ensures that this is at least $C_{\bar{A}}(\rho)$. ∎

This completes the proof of the first part of Lemma 7.1. We proceed to the proof of the second part. We state without proof a routine fact concerning the function C_{OPT}.

Lemma 7.2 *Let $\mathcal{M}$ be a metric space of diameter δ and let $\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, \dots, \boldsymbol{\rho_w}$ be probe sequences. Then:*

$$\sum_{i=1}^{w} C_{\mathrm{OPT}}(\boldsymbol{\rho_i}) \leq C_{\mathrm{OPT}}(\boldsymbol{\rho_1}\boldsymbol{\rho_2} \dots \boldsymbol{\rho_w}) \leq \sum_{i=1}^{w} C_{\mathrm{OPT}}(\boldsymbol{\rho_i}) + (w-1)\delta.$$

Furthermore, the inequalities extend to the case that the ρ_i are replaced by randomized probe sequences $\tilde{\boldsymbol{\rho}}_{\mathbf{i}}$.

(Recall that the definition of C_{OPT} allows the evader to choose its own starting point. So, if each $\boldsymbol{\rho_i}$ misses some point in $\mathcal{M}$, then $\sum_{i=1}^{w} C_{\mathrm{OPT}}(\boldsymbol{\rho_i}) = 0$, and if all $\boldsymbol{\rho_i}$ miss the *same* point in $\mathcal{M}$, then $C_{\mathrm{OPT}}(\boldsymbol{\rho_1}\boldsymbol{\rho_2} \dots \boldsymbol{\rho_w}) = 0$.)

We now prove the second part of Lemma 7.1. The middle inequality is trivial, since the only difference between G_s and H_s is that the Pursuer strategy set in H_s is a subset of that in G_s.

Consider the first inequality and suppose for contradiction that there is a $s > 0$ and $\epsilon > 0$ such that $V(H_s) = -\delta - \epsilon$. Let $\tilde{B}_s$ denote an optimal Evader algorithm for H_s. We define an Evader algorithm $\tilde{B}$ for the original Pursuit-Evasion game as follows: for probe sequence τ determine its s-block partition, $\tau_1\tau_2\ldots\tau_r$, which can be parsed online. The algorithm $\tilde{B}$ is performed by applying $\tilde{B}_s$ to each τ_i. We obtain a contradiction by showing that $\tilde{B}$ is λ'-competitive for some $\lambda' < \lambda$. There is some absolute upper bound K on the cost incurred by $\tilde{B}_s$ on a sequence of optimal cost at most s. We have:

$$\begin{aligned}
C_{\tilde{B}}(\tau) &\leq \sum_{i=1}^{r-1}(C_{\tilde{B}_s}(\tau_i)+\delta)+C_{\tilde{B}_s}(\tau_r) \\
&\leq \sum_{i=1}^{r-1}(\lambda C_{\mathrm{OPT}}(\tau_i)-\epsilon)+K \\
&\leq \sum_{i=1}^{r-1}(\lambda-\frac{\epsilon}{s})C_{\mathrm{OPT}}(\tau_i)+K \\
&\leq (\lambda-\frac{\epsilon}{s})C_{\mathrm{OPT}}(\tau)+K,
\end{aligned}$$

where the first and last inequalities follow from Lemma 7.2, the second inequality follows from the optimality of $\tilde{B}_s$ for H_s and the third inequality follows from the fact that each τ_i has optimal cost at most s. We conclude that that $\tilde{B}$ is $(\lambda-\frac{\epsilon}{s})$-competitive, a contradiction that completes the proof of the first inequality.

Turning to the third inequality, we suppose, for contradiction that there is a $s > 0$ and $\epsilon > 0$ such that $V(G_s) \geq \lambda\delta+\epsilon$. Let $\tilde{\rho}_s$ be the optimal Pursuer strategy for G_s. Then for any algorithm $\tilde{B}$, $C_{\tilde{B}}(\tilde{\rho}_s) \geq \lambda(C_{\mathrm{OPT}}(\tilde{\rho}_s)+\delta)+\epsilon$. For each $j \in \mathrm{N}$, define the distribution $\tilde{\tau}_j$ on probe sequences obtained by concatenating j sequences generated independently from $\tilde{\rho}_s$. Then, by Lemma 7.2, $C_{\mathrm{OPT}}(\tilde{\tau}_j)$ is bounded above by $u_j = j(C_{\mathrm{OPT}}(\tilde{\rho}_s)+\delta)$. We will obtain a contradiction by showing that there exists a real number $\gamma > 0$ such that for any deterministic algorithm A, $C_A(\tilde{\tau}_j) \geq u_j(\lambda+\gamma)$, which by Proposition 2.3 would imply that the competitive ratio is greater than $\lambda+\gamma$.

Let A be a deterministic algorithm and consider the cost of A on $\tilde{\tau}_j$. By Lemma 7.2, we can lower bound this cost by the sum of costs that are incurred in responding to each of the j blocks. Note that the way that A responds to the i-th block depends on the previous $i-1$ blocks, and we can view A's behavior on the i-th block as a randomized algorithm $\tilde{B}_i$ where the randomization comes from the previous $i-1$ blocks. Thus,

$$C_A(\tilde{\tau}_j) \geq \sum_{i=1}^{j} C_{\tilde{B}_i}(\tilde{\rho}_s)$$

$$
\begin{aligned}
&\geq j[\lambda(C_{\mathrm{OPT}}(\tilde{\rho}_{\mathbf{s}}) + \delta) + \epsilon] \\
&= u_j(\lambda + \frac{\epsilon}{C_{\mathrm{OPT}}(\tilde{\rho}_{\mathbf{s}}) + \delta}),
\end{aligned}
$$

where the second inequality follows from the optimality of $\tilde{\rho}_s$. Thus we have the desired contradiction, which establishes the claim and the lemma.

8 Acknowledgement

We would like to thank the anonymous referees for the many useful comments in their reports, which helped improve the presentation of our results.

References

[BBKTW] S. Ben-David, A. Borodin, R.M. Karp, G. Tardos, and A. Wigderson. On the Power of Randomization in Online Algorithms. *Algorithmica*, 11:2–14,1994.

[BBBT] Y. Bartal, A. Blum, C. Burch, and A. Tomkins A polylog(n)-competitive algorithm for metrical task systems In *Proc. of the 29rd Ann. ACM Symp. on Theory of Computing*, pages 711–719, May 1997.

[BLS] A. Borodin, N. Linial, and M. Saks. An Optimal On-Line Algorithm for Metrical Task Systems. *Journal of the ACM*, 39:745–763, 1992.

[BE] A. Borodin and R. El Yaniv *Online computation and competitive analysis* Cambridge University Press, 1998.

[BRS] A. Blum, P. Raghavan, and B. Schieber. Navigating in Unfamiliar Geometric Terrain. In *Proc. of the 23rd Ann. ACM Symp. on Theory of Computing*, pages 494–504, May 1991.

[CDRS] D. Coppersmith, P. Doyle, P. Raghavan, and M. Snir. Random Walks on Weighted Graphs and Applications to On-line Algorithms. *Journal of the ACM*, 40(3):421–453, 1993.

[FKLMSY] A. Fiat, R.M. Karp, M. Luby, L.A. McGeoch, D.D. Sleator, and N.E. Young. Competitive Paging Algorithms. *Journal of Algorithms*, 12:685–699, 1991.

[FRR] A. Fiat, Y. Rabani, and Y. Ravid. Competitive k-Server Algorithms. *Journal of Computer and Systems Sciences* 48(3):410–428, 1994.

[Gro] E. GROVE. The Harmonic k-Server Algorithm is Competitive. In *Proc. of the 23nd Ann. ACM Symp. on Theory of Computing*, pages 260–266, May 1991.

[KMRS] A.R. KARLIN, M.S. MANASSE, L. RUDOLPH, AND D.D. SLEATOR. Competitive Snoopy Caching. *Algorithmica*, 3(1):79–119, 1988.

[KRR] H.J. KARLOFF, Y. RABANI, AND Y. RAVID. Lower Bounds for Randomized k-Server Algorithms. *SIAM Journal on Computing*, 22(2):293–312, 1994.

[Kel] J.L. KELLEY. *General Topology* New York: American Book Company, 1955.

[KP] E. KOUTSOUPIAS AND C. PAPADIMITRIOU. On the k-Server Conjecture. In *JACM*, 42:971-983, 1995.

[LR] C. LUND AND N. REINGOLD. Linear Programs for Randomized On-Line Algorithms. In *Proc. of the 5th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 382–391, January 1994.

[MMS] M.S. MANASSE, L.A. MCGEOCH, AND D.D. SLEATOR. Competitive Algorithms for On-line Problems. *Journal of Algorithms*, 11:208–230, 1990.

[Mat] J. Matoušek. Ramsey-like Properties for bi-Lipschitz Mappings of Finite Metric Spaces. *Commentationes Mathematicae Univ. Carolinae*, 33(3):451–463, 1992.

[MS] L.A. MCGEOCH AND D.D. SLEATOR. A Strongly Competitive Randomized Paging Algorithm. *Algorithmica*, 6:816–825, 1991.

[RS] P. RAGHAVAN AND M. SNIR. Memory versus Randomization in On-Line Algorithms. In *Lecture Notes in Computer Science* 372, pages 687–703, Springer-Verlag, 1989.

[Sag] H. SAGAN. *Advanced Calculus* Boston:Houghton-Mifflin, 1974.

[ST] D.D. SLEATOR AND R.E. TARJAN. Amortized Efficiency of List Update and Paging Rules. *Communication of the ACM*, 28(2):202–208, 1985.

[vNM] J. VON NEUMANN AND O. MORGENSTERN *Theory of Games and Economic Behavior*, 2nd Ed. Princeton: Princeton University Press, 1947.