Department of Electrical Engineering and Computer Science
Case Western Reserve University

Randomization Constrained

V. Liberatore

Technical Report EECS–02–14, 2002

Randomization Constrained

Vincenzo Liberatore*

April 27, 2002

Abstract

Yao's principle is a fundamental techique for proving lower bounds on randomized algorithm and is based on a game-theoretical duality result by von Neumann. In this paper, we prove an extension of the principle to the case when one of the players is further constrained by a set of linear inequalities. The corresponding duality result is interpreted in a variety of algorithmic contexts, including multi-objective optimization problems, performance tail of randomized algorithms, constrained adversaries, resource augmentation method, smoothed analysis, high-probability results, and loose-competitiveness.

*Electrical Engineering and Computer Science Department, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, Ohio 44106-7221. E-mail: vxl11@po.cwru.edu. URL: http://vorlon.cwru.edu/~vxl11/. This work has been supported in part under NSF grant ANI-0123929.

1 Introduction

Game theory is central to the design and the analysis of algorithms. Its relevance can ultimately be attributed to the widespread adoption of the worst-case paradigm, which views the algorithm as a player against an adversary that produces worst-case input instances. A striking application of the game-theoretical viewpoint is *Yao's principle* [26], which applies von Neumann's original result [25] to establish lower bounds on randomized algorithms. The elegance of the method ultimately derives from its reliance on universal and abstract results in linear algebra.

In this paper, we prove an extension to the original duality result for the case when one of the player is subject to additional constraints. The extension is applicable to several contexts in the area of randomized algorithms, including multi-objective optimization problems [20], performance tail of randomized algorithms [14], the resource augmentation method [11], smoothed analysis [23], and loose-competitiveness [27]. The rest of this section discusses the consequences of our main duality result for algorithm design and analysis.

Algorithms. Several optimization problem naturally lend themselves to a multi-objective formulation, where algorithms can attempt to minimize any one of multiple objective functions. Several methods have been proposed to deal with multi-objective problems, including the derivation of exact [21] or approximate [20] Pareto equilibria. Our goal is to provide and demonstrate a general method to prove lower bounds on the performance of randomized algorithms for multi-objective optimization problems. In this case, the explicit construction of Pareto equilibria is not feasible in that the game-theoretical formulation of the problem is typically too large. However, we can abstractly exploit duality, much like in Yao's principle where the linear program is not solved numerically but duality gives us a general method to prove lower bounds. As an application of the general method, we analyze the multi-objective version of the *shut-down scheduling* problem [15]. Furthermore, our method is general enough that we can apply it to the following non-standard multi-objective optimization problem: find a randomized algorithm whose performance is good both in the expectation and with high-probability. The performance tail is relevant both in a theoretical and in an applied context. In applied fields, the expected performance of algorithms is seldom considered a sufficient characterization of algorithm behavior, and, in particular, Service Level Agreements between customers and suppliers almost always include clauses on both expected and tail behavior [5]. From a theoretical standpoint, certain randomized algorithms, especially within the *Classify-and-Randomly-Select* framework [1, 16], have optimal expected performance, but fail to achieve a constant fraction of their expected payoff with constant probability [14]. In this paper, we prove that the lower bound of [14] can be obtained as an application of our method.

Adversaries. Worst-case analysis often runs into the "triviality barrier": algorithms are trivially optimal in the worst-case but are not intuitively appealling [7]. A method to weaken worst-case analysis is to impose additional constraints on the adversary. The idea of constrained adversaries is well-known in Mathematical Economics [8, 10] and Control Theory [2], and it has been recently and independently introduced into Computer Science [13]. The main idea behind constrained adversaries is that the input instance is generated according to a worst-case distribution chosen from a given class of distributions. In other words, the adversary has to satisfy an additional set of constraints that define the set of feasible input distributions. In this paper, we model constrained adversary literally in that we append a new set of linear constraints to the original linear programming formulation of the game. The technique fits our general approach once the roles of the two players are exchanged.

For the purpose of applying the framework to algorithm design, we first consider an adversary which is forced to choose a strategy x with probability at most b_x. If the adversary were allowed to be a worst-case probabilistic player, then such an adversary would be equivalent to conferring the power of randomization onto the algorithm. Since the upper-bounded adversary is further restricted, then the adversary should be equivalent to conferring the power of randomizatiom onto the algorithm as well as other powers. We prove that the additional power is as follows: the optimal cost on input x undergoes a relative increase $\delta(x)$. Furthermore, the algorithm player can choose any positive value for $\delta(x)$ but it is charged a penalty proportional to $E_{\mathcal{B}}[\delta]$ for doing so, where $\mathcal{B}$ is a certain canonical distribution derived from the bounds b_x. We call the resulting model the *adversary cost increase model.* The ability to increase the adversary cost immediately leads us to compare upper-bounded adversary with the resource augmentation method (RAM) and with smoothed analysis.

RAM and smoothed analysis. In the RAM, the algorithm resources are augmented as compared to those given to the adversary [11]. Thus, the resource augmentation causes the adversary cost to increase by $\delta(x)$ on input x. On the other hand, if we were allowed to increase arbitrarily the algorithm resources, we would often obtain the trivial result that the algorithm is super-optimal. Therefore, it is critical that the amount of resource augmentation be compared directly to the performance improvement. One such comparison is expressed by the penalty $E[\delta]$ which is embedded in the adversary cost increase model. The same argument holds mostly unchanged for smoothed analysis, where the algorithm cost is reduced by a $\delta(x)$ factor corresponding to its expected cost in a neighborhood of the original instance x [23]. Thus, the two techniques of RAM and smoothed analysis are given a unified interpretation in terms of the adversary cost increase model. In particular, results on randomized algorithms in either of the two models can be translated into results on deterministic algorithms against an upper bounded adversary. The main open problem in this paper is to deal with the non-linear trade-offs that are manifest in the RAM and in smoothed analysis. A method to attack such non-linearities is loose competitiveness (see below), but we conjecture that a more general theory should be possible.

Asymptotic analysis. The adversary cost increase method is related to the resource augmentation method and to smoothed analysis in that it flattens away the contribution of worst-case instances when such instances are only a small fraction of the input instances. An application of such method arises in an asymptotic sense, that is, when the fraction of bad instances tends to zero as the problem size increases. We will prove a general asymptotic result. However, the result relies on the law of large numbers, which in general gives no guarantee on the speed of convergence to the average or, in dual terms, it can require a heavily restricted upper-bounded adversary. In specific applications, asymptotic results can be obtained with less restricted adversaries. For example, we will easily show that results against mildly restricted adversaries follow immediately from a high-probability characterization of the performance of a randomized algorithm. As an example of a problem where the objective is to minimize a cost ratio, we analyze a version of the *paging problem* [22] where the page request sequence is generated by combining i.i.d. hard phases, and show that the asymptotic result is obtained even when the upper-bounded adversary is constrained by an exponentially smaller extent than what we would get from the generic law of large numbers.

Another application of asymtotic analysis is to deal with the non-linearity implicit in RAM. To this end, we consider the loose competitiveness of the paging problem, where the adversary chooses a value of the cache size. If the page set size tends to infinity and the adversary is upper bounded, the difficult values of the cache size give a negligible contribution to the algorithm cost. Our result can be viewed as the derandomization of the original result on *loose competitiveness* [27] in that the previous result required both the cache size and the algorithm to be probabilistic, whereas we only need the cache size to be random. However, the results are formally incomparable due to our statement in terms of expectation of a ratio rather than ratio of expectations.

Lower-bounded adversaries. The upper-bounded adversaries have a very rich structure that leads to their applicability to a variety of contexts. It is then natural to ask whether similar results would also for *lower-bounded adversaries*, which are forced to choose an input instance with probability at least b_x. However, we easily show that lower-bounded adversaries are substantially less interesting, and basically only force a trade-off between worst-case analysis and an average-case analysis.

Contents. Section 2 introduces definitions and notation. Section 3 discusses our main duality result, which is applied to algorithms in Section 4 and to adversaries in Section 5. Further details on shut-down scheduling and lower-bounded adversaries are omitted from the body of the paper and can be found in the appendix.

2 Preliminaries

In this paper, we model randomized algorithms and their worst-case behavior through two person zero-sum games [4, 26]. Specifically, let Y be the set of instances of a given problem and X be the set of all deterministic algorithms for the same problem. We assume that both Y and X are finite and define $n = |X|$ and $m = |Y|$. In general, a deterministic algorithm $x \in X$ will achieve a certain level of "performance" on input $y \in Y$. To model such performance considerations, we define $u(x, y)$ as the *utility* that algorithm x accrues on input y. A randomized algorithm is a probability distribution G over X and its expected performance on input y is $E_G[u(x, y)]$ (which will also be denoted for simplicity as $u(G, y)$). A (worst-case) adversary chooses a y so as to minimize $u(G, y)$, and the objective of the algorithm designer is to find a G whose expected utility is

$$u^* = \max_G \left\{ \min_{y \in Y} u(G, y) \right\} . \tag{1}$$

Thus, the two-person zero-sum game (1) formulates the algorithm design problem. Game (1) can be expressed as the following linear program

$$u^* = \max\{u : A\mathbf{p} \geq u\mathbf{e}_m, \mathbf{e}_n^T \mathbf{p} = 1, \mathbf{p} \geq 0\} ,$$

where $A = (a_{yx})$ is the $m \times n$ matrix with entries $a_{yx} = u(x, y)$ and $\mathbf{e}_n = (1\ 1\ \ldots\ 1)^T$ is the n-vector of 1's. In the notation $\mathbf{e}_n$, we will omit the subscript n when the size of $\mathbf{e}$ is clear from the context. Another notation used in the paper is that $\mathbf{b} = (b_x)$ is a vector with an entry for each element in $x \in X$, which can also be regarded as a function from X to the reals. A simple application of linear programming duality leads to

$$u^* = \min\{u : A^T\mathbf{q} \leq u\mathbf{e}, \mathbf{e}^T\mathbf{q} = 1, \mathbf{q} \geq 0\} ,$$

and since $\mathbf{q}$ can be interpreted as a probability distribution over Y, we have *von Neumann's minimax principle* [25]:

$$u^* = \min_H \left\{ \max_{x \in X} u(x, H) \right\} , \tag{2}$$

where $u(x, H) = E_H[u(x, y)]$. In particular, observe that if H is a probability distribution over Y, then

$$u^* \leq \max_{x \in X} u(x, H) , \tag{3}$$

which is *Yao's principle* [26]: the utility of the best *randomized algorithm* on a *worst-case input* is no more than the expected utility of the best *deterministic algorithm* on a *random input*. Typically, the application of Yao's principle requires a careful choice of the input distribution H to make inequality (3) as tight as possible. Notice that formulation (1) and (2) are symmetric, and thus the role of the algorithm and of the adversary can be switched. Therefore, we will talk about player I (which chooses a probability distribution over X) and player II (which chooses a probability distribution over Y). As we subsequently identify player I with the algorithm or with the adversary, the duality and game-theoretical results will be interpreted in different ways. The game (1) is defined for n and m finite, but algorithm analysis is often concerned with the asymptotic behavior of algorithms. In these cases, we consider an infinite sequence of programs (1) that are parameterized by a certain value s. Typically, the parameter s is the input size in the case of off-line algorithms and is the request sequence length in the case of on-line algorithms. Then, Y is the set of all instances of size at most s and X is the set of all distinct algorithms on Y. The game value $u^* = u^*(s)$ now depends on the parameter s and our objective will be to obtain an asymptotic characterization of the function $u^*(s)$. It is beyond the scope of this paper to examine games with infinitely many strategies. A major thrust of this paper is to consider players that have additional constrains, as we discuss in the next section.

3 Main Duality Result

We consider an additional set of strategies Y_v for player II. We assume again that Y_v is finite and let $m_v = |Y_v|$. In many cases, $Y_v \subseteq Y$, but in other cases $Y_v - Y \neq \emptyset$ or even $Y_v = X$. Let $v(x, y)$ be a *payoff* obtained by player I when player I uses strategy X and player II uses strategy $y \in Y_v$. We consider a game (1) with the additional constraint that the expected payoff of player I is not negative:

$$u_v^* \stackrel{\text{def}}{=} \max_G \left\{ \min_{y \in Y} u(G, y) : \min_{y \in Y_v} v(G, y) \geq 0 \right\} . \tag{4}$$

Proposition 3.1. *Consider a game* (4), *and let* $A = (a_{yx})$ *be an* $m \times n$ *matrix with* $a_{yx} = u(x, y)$ *and* $B = (b_{yx})$ *an* $m_c \times n$ *matrix with* $b_{yx} = v(x, y)$. *Then,*

$$u_v^* = \min\{u : A^T\mathbf{q} + B^T\mathbf{r} \leq u\mathbf{e}, \mathbf{e}^T\mathbf{q} = 1, \mathbf{q} \geq 0, \mathbf{r} \geq 0\} .$$

Proof. The linear programming formulation of (4) is

$$u_v^* = \max\{u : A\mathbf{p} \geq u\mathbf{e}, B\mathbf{p} \geq 0, \mathbf{e}^T\mathbf{p} = 1, \mathbf{p} \geq 0\} .$$

The dual of this linear program is the one in the claim, which completes the proof. □

Corollary 3.2. *Let $\alpha \geq 0$ be an* interaction factor, *H a probability distribution over Y, and H_v a probability distribution over Y_v. Then for all $x \in X$, $u_v^* \leq u(x,H) + \alpha v(x,H_v)$. Conversely, there are H, H_v, and α such that $u_v^* = u(x,H) + \alpha v(x,H_v)$.*

Proof. Consider any choice of H, H_v, and α as in the statement of the corollary. Let $p_x = \Pr_H[x]$ and $r_x = \alpha \Pr_{H_v}[x]$ and observe that the resulting $\mathbf{p}$ and $\mathbf{r}$ are a feasible solution of the program in the previous proposition.

Conversely, consider any feasible solution $\mathbf{q}, \mathbf{r}, u$ for player II, and let $\alpha = \mathbf{e}^T\mathbf{r}$. Observe that $\mathbf{q}$ can be interpreted as a probability distribution H over X. We now turn to $\mathbf{r}$ and first we assume that $\mathbf{r} = 0$. Then, take $\alpha = 0$ and H_v to be any distribution over Y. Suppose now that $\mathbf{r} \neq 0$. Normalize $\mathbf{r}$ to $\alpha = \mathbf{e}^T\mathbf{r}$ so as to interpret $\mathbf{r}$ as the probability distribution H_v over Y, which concludes the proof. □

The claims above generalize immediately for the case when there are multiple payoffs:

$$u_v^* \stackrel{\text{def}}{=} \max_G \left\{ \min_{y \in Y} u(G,y) : \min_{y \in Y_{v_1}} v_1(G,y) \geq 0, \ldots, \min_{y \in Y_{v_k}} v_k(G,y) \geq 0 \right\} .$$

We can assume without loss of generality that the Y_{v_j}'s are disjoint, take $Y = Y_{v_1} \cup \cdots \cup Y_{v_k}$, and let $v(x,y) = v_j(x,y)$ if $y \in Y_{v_j}$.

A general fact that we will use later on is:

Lemma 3.3. *If $U \geq u_v^*$ and G is a mixed strategy for player I with the property that $\min_{y \in Y_v} v(G,y) \geq 0$, then there exists a $y \in Y$ with $u(G,y) \leq U$.*

4 Algorithms

In this section, we examine the implications of Corollary 3.2 for randomized algorithms when additional constraints are imposed. In particular, we demonstrate the general approach with an detailed derivation of a lower bound for call admission on the line.

Given a line graph with n vertices and $m = n - 1$ edges (the definitions of n and m in this section are unrelated to those in the rest of the paper), an on-line algorithm is presented with a sequence of vertex pairs, each of which is termed a *call*. If the graph contains a path between the two call endpoints, the algorithm has the option of accepting the call, in which case it deletes all the edges in the path between the two call endpoints. The algorithm objective is to accept the maximum possible number of calls. The best competitive ratio for a randomized algorithm is $O(\log m)$ [1]. Define $c(x,y)$ as the number of calls accepted by x on the input sequence y, and $c^*(y)$ as the maximum number of calls that can be accepted from y. We use duality (Corollary 3.2) to prove the following result from [14].

Theorem 4.1. *For any randomized strictly $(c(\log m-1))$-competitive algorithm for call admission on the line of n nodes, and any constant probability of failure p bounded by $p \le P < 1/(4c)$, there is a sequence y with $c^*(y) \ge (m/2)^{1-2cp}$ such that with probability at least p the algorithm fails to achieve any constant fraction of the expected profit ratio.*

Proof. We define suitable utility and payoff functions and cast the problem in terms of Corollary 3.2. First, Y_v is the set of call sequences on a line graph with n vertices and X is the set of deterministic on-line algorithms. Let $\tilde{v}(x,y) = c(x,y)/c^*(y)$ and $v(x,y) = \tilde{v}(x,y) - 1/(c(\log m - 1))$, so that the constraint that $\min_{y\in Y_v} v(G,y) \ge 0$ implies that the on-line algorithm G is strictly $(c(\log m-1))$-competitive. As for Y, it is the set of request sequences y with the property that $c^*(y) \ge (m/2)^{1-\epsilon}$, where $\epsilon = 2cp$. Notice that $0 \le \epsilon < 1/2$, which is a fact that we will use later on in the proof. Define $\widetilde{m}$ to be the largest power of 2 that is not larger than m, so that $\widetilde{m} \le m < 2\widetilde{m}$. Let

$$u(x,y) = \begin{cases} -1 & \text{if } \tilde{v}(x,y) \le \widetilde{m}^{2\epsilon-1} \\ 0 & \text{otherwise} \end{cases} ,$$

Hence, $E[u] = -\Pr[v \le \widetilde{m}^{2\epsilon-1}]$. Therefore, an adversary can force a randomized strictly $(c(\log m - 1))$-competitive algorithm to achieve an $\widetilde{m}^{2\epsilon-1} = o(1/(c\log m))$ fraction of the optimal profit with probability at least $-u_v^*$. It remains to show that $u_v^* \le -p$.

In order to define the probability distributions H and H_v, we first introduce the notion of request classes. The *ith request class* $(0 \le i \le \log\widetilde{m})$ consists of the 2^i calls from $(j-1)\widetilde{m}/2^i + 1$ to $j\widetilde{m}/2^i + 1$ $(1 \le j \le 2^i)$. The ith request class contains 2^i pairwise disjoint calls of size $\widetilde{m}/2^i$. Define $\ell = \lfloor \epsilon \log \widetilde{m} \rfloor$. The probability distribution H_v is to request the calls in the ith request class ($i = 1, 2, \ldots, k$ in the given order), where k is extracted uniformly at random between 0 and ℓ. The distribution H is to request all calls in the ith request classes for $i = 0, 1, \ldots, \ell$ in the same order as H_v does, to select a call (u,v) uniformly at random in the ℓth request class and to request all edges in the path from u to v as individual calls of unit length. We claim that H is a distribution over Y: indeed, any such sequence y has optimum profit $c^*(y) = \widetilde{m}/2^\ell \ge \widetilde{m}/2^{\epsilon\log\widetilde{m}} = \widetilde{m}^{1-\epsilon} \ge (m/2)^{1-\epsilon}$. If G fails to accept the requests of unit length, its payoff is at most $2^\ell/\widetilde{m}^{1-\epsilon} \le \widetilde{m}^{\epsilon}/\widetilde{m}^{1-\epsilon} = \widetilde{m}^{2\epsilon-1}$. Consider a deterministic algorithm G that accepts q_i calls from the ith request class during H's request sequence. The probability that G fails to accept the calls of unit length is at least $\sum_{i=0}^{\ell} 2^{\ell-i} q_i/2^\ell = \sum_{i=0}^{\ell} q_i/2^i \le -E_H[u]$. Furthermore, H_v's request sequence is always a prefix of H's, and so the deterministic algorithm G accepts q_i calls from the ith request class during H_v's request sequence as well $(0 \le i \le k)$. It follows that

$$E_{H_v}[\tilde{v}] = \frac{1}{1+\ell}\sum_{k=0}^{\ell}\sum_{j=0}^{k}\frac{q_j}{2^k} = \frac{1}{1+\ell}\sum_{j=0}^{\ell} q_j \sum_{k=j}^{\ell}\frac{1}{2^k} < \frac{2}{1+\ell}\sum_{i=0}^{\ell}\frac{q_i}{2^i} .$$

Take $\alpha = (1+\ell)/2$, to obtain that

$$u_v^* \le E_H[u] + \alpha E_{H_v}[v] - \frac{\alpha}{c\log m - c} \le -\frac{\epsilon}{2c} = -p .$$

Therefore, the probability that G fails to achieve a constant fraction of its expected profit ratio is at least p, which concludes the proof. ∎

5 Adversaries

We will now turn to consider the case where the adversary chooses a worst-case probabilistic input that satisfies a set of additional constraints. In this case, player I corresponds to the adversary and player II to the algorithm. In this section, we consider *upper-bounded adversaries*, which are constrained to choose a strategy x with probability no more than b_x. The upper-bounded adversary chooses a distribution G over X with the property that $\Pr_G[x] \le b_x$ so as to achieve payoff

$$u_v^* = \max_G \left\{ \min_{y \in Y} u(G,y) : \Pr_G[x] \le b_x \right\} .$$

We formulate upper-bounded adversaries in term of Proposition 3.1. If we define $Y_v = X$ and

$$v(x,y) = \begin{cases} b_x - 1 & \text{if } x = y \\ b_x & \text{otherwise} \end{cases} ,$$

then $v(G,x) = b_x - \Pr_G[x]$. Hence, $\min_{y \in Y_v} v(G,y) \ge 0$ holds if and only if $p_x \le b_x$ for all $x \in X$. We now introduce definitions and certain assumptions that hold without loss of generality. First, let $\mathbf{b} = (b_x)$ and assume without loss of generality that $0 \le \mathbf{b} \le \mathbf{e}$. Observe that if $\mathbf{e}^T\mathbf{b} = 1$, then the vector $\mathbf{b}$ forces $p_x = b_x$. We will assume without loss of generality that $\mathbf{e}^T\mathbf{b} > 1$, and let $\lambda = \mathbf{e}^T\mathbf{b} - 1 > 0$. The quantity λ can be interpreted as a measure of how far away $\mathbf{b}$ is from a probability distribution. Indeed, observe that if $\lambda = 0$, then the adversary is completely forced to use a certain mixed strategy, whereas $\lambda \gg 0$ leaves ample room of maneuver to the adversary. Thus, λ can also be interpreted as the strength of the constraint on the adversary. Finally, we let $\mathbf{b}' = \mathbf{b}/(\mathbf{e}^T\mathbf{b}) = \mathbf{b}/(1+\lambda)$ and interpret $\mathbf{b}'$ as a probability distribution $\mathcal{B}$ over X. An upper-bounded adversary is completely characterized by $\mathcal{B}$ and λ, and so an equivalent definition is that an upper-bounded adverary is a pair $(\mathcal{B}, \lambda)$.

Definition 5.1. Denote by $\mathbb{R}_0^+$ the set of non-negative reals and by $\mathbb{R}^+$ the set of positive reals.

Theorem 5.1. *Consider a game* (4) *where player I is an upper-bounded adversary* $(\mathcal{B}, \lambda)$. *Let* H *be a probability distribution over* Y *and* $r : X \to \mathbb{R}_0^+$. *Then,*

$$u_v^* \le \max_{x \in X}\{u(x,H) - r(x)\} + (1+\lambda)E_{\mathcal{B}}[r] .$$

Proof Sketch. Establish the eigenvector of B and apply Proposition 3.1. □

The theorem implies the following fact for the performance of deterministic algorithms against probabilitic upper-bounded adversaries.

Corollary 5.2. *Let* $\mathcal{B}$ *and* G *be probability distributions over* X *with the property that* $\Pr_G[x] \le (1+\lambda)\Pr_{\mathcal{B}}[x]$ *for some* $\lambda > 0$. *Let* H *be a probability distribution over* Y *and* $r : X \to \mathbb{R}_0^+$. *Then, there is a* $y \in Y$ *such that*

$$u(G,y) \le \max_{x \in X}\{u(x,H) - r(x)\} + (1+\lambda)E_{\mathcal{B}}[r] .$$

The original question was to establish which powers are conferred by the upper-bounded adversary onto the algorithm in addition to randomization. The answer lies in Theorem 5.1. The algorithm can use randomization as well as can "borrow" a value $r(x)$ to increase its performance on an instances x. However, the expected amount of "borrowing" will contribute to the objective value. Of course, the algorithm can avoid any borrowing and fall back into the worst-case probabilistic adversary. Therefore, the process of borrowing (i.e., the restricition to upper-bounded adversaries) does not hamper the algorithm. The borrowing can work especially well when most instances are easily solved and the worst-case performance is tied to the payoff in the presence of few difficult instances. In this case, the algorithm can borrow to improve its own performance on the hard instances and can pay back on the easy instances. If there are many more easy instances than difficult one, the algorithm can keep the expected borrowing $E_{\mathcal{B}}[r]$ small and improve its performance. Examples will be shown throughtout the rest of the paper.

We will especially consider the case when the utility $u(x, y)$ can be expressed as the ratio of the algorithm cost $h(x, y)$ over the adversary cost $g(x)$, which is the typical scenario whenever approximation ratios or competitive ratios are of interest.

Corollary 5.3. *Consider a game* (4) *where* $u(x,y) = h(x,y)/g(x)$, $h(x,y) \geq g(x) > 0$ *(for all* $x \in X$ *and* $y \in Y$*), and player I is a upper-bounded adversary* $(\mathcal{B}, \lambda)$ *where* $\lambda > 0$. *Then,*

$$\max_{G} \left\{ \min_{y \in Y} E_G[u(x,y)] : \Pr_G[x] \leq (1+\lambda) \Pr_{\mathcal{B}}[x] \right\} = \min_{\delta \geq 0} (1 + (1+\lambda) E_{\mathcal{B}}[\delta]) \min_{H} \max_{x \in X} \frac{h(x,H)}{g(x)(1+\delta(x))} \ .$$

Proof Sketch. The main difficulty of the proof is to show that $r \neq 0$. One of the elements used to prove this fact is Loomis' lemma. □

The following definition formalizes the model in Corollary 5.3, namely the idea that the adversary cost is inflated by a factor of $1 + \delta$ when algorithm and optimum costs are compared.

Definition 5.2. We will say that a probability distribution H over Y (a randomized algorithm) is a *u-approximation algorithm in the adversary cost increase model* $(\mathcal{B}, \lambda)$ if there exists a function $\delta : X \to \mathbb{R}_0^+$ such that, for all $x \in X$, $h(x, H) \leq g_\delta(x)/(1+(1+\lambda)E_{\mathcal{B}}[\delta])$, where $g_\delta(x) = g(x)(1+\delta(x))$.

The following corollary basically restates Corollary 5.3 and expresses the fundamental equivalence of the upper-bounded and of the adversary cost increase models.

Corollary 5.4. *If there exists a u-approximation algorithm in the advesary cost increase model* $(\mathcal{B}, \lambda)$*, then for any probabilistic adversary G with the property that* $\Pr_G[x] \leq (1+\lambda)\Pr_{\mathcal{B}}[x]$ *there exists a deterministic algorithm y such that* $u(G, y) \leq u$.

In the next sections, we relate the adversary cost increase model with other techniques to analyze algorithm performance whose core method is to inflate the adversary's cost.

5.1 Resource Augmentation Method and Perturbation-Based Analysis

In the resource augmentation method (RAM), the algorithm cost is compared to that of an adversary that has less resources than the algortithm does [11]. Thus, the method gets its name by augmenting the resources available to the algorithm as compared to those available to the adversary. Equivalently, the adversary uses fewer resources than the algorithm does, which in turn implies that the adversary profit $g_\delta(x)$ on an instance x is $1+\delta(x) \geq 1$ times the optimum profit $g(x)$ on instance

x. The common practice is to fix the amount of resources available to the algorithm and to the adversary and to analyze the resulting worst-case performance. Suppose now that the adversary resources are not fixed but can be chosen in any arbitrary way with the objective of improving the relative performance of the algorithm. Since the relative algorithm performance should improve if we give the adversary less resources, the RAM method poses a trade-off between the relative performance of algorithms and the amount of resource augmentation. Therefore, the RAM method can be viewed as a bi-criterion optimization problem, where the algorithm performance should be weighted against the amount of resource augmentation. We will examine the trade-off by combining the two criteria in one objective function.

We now formalize the discussion above. In the most common cases, the performance metric is the cost ratio $h(x,y)/g(x)$, which the RAM will substitute with $h(x,y)/g_\delta(x)$ for some fixed relative adversary cost increase function δ. We will consider a multiplicative penalty of the form $-(1+\lambda)E_{\mathcal{B}}[\delta]$, where λ is an appropriate weight factor and $\mathcal{B}$ is a given probability distribution over X. We will denote the resulting problem as the *RAM trade-off problem* with parameters $\mathcal{B}$ and λ.

Corollary 5.5. *The cost of the best randomized algorithm in the RAM trade-off problem with parameters $\mathcal{B}$ and λ is not smaller than the smallest approximation factor u_v^* of a randomized algorithm in the adversary cost augmentation model $(\mathcal{B},\lambda)$.*

Analogous consideration can be made for perturbation-based analysis, where the algorithm cost on instance x is replaced by the expected algorithm cost on a set $N(x) \subseteq x$. For example, the set $N(x)$ can be taken as a neighborhood of x of a given fixed radius if the set X of instances is a metric space [23]. The analysis presents a trade-off between the algorithm performance and the radius on the neighborhood, where a zero radius specializes to worst-case analysis. The perturbation-based trade-off is substantially equivalent to an upper-bounded adversary under appropriate hypotheses on the trade-off definition. The argument is essentially similar to the one in Corollary 5.5, but with the addede complication that $\delta(x)$ can now be negative. Thus, we take δ to be the positive part of the expected cost in $N(x)$. The long but mostly technical formalization of this claim is omitted from this paper.

Both models can be formulated in terms of adversary cost increase functions, and, by Corollary 5.4, the game values are not smaller than the minimum cost ratio $u(G,y)$ of a deterministic algorithm y against a probabilistic and upper-bounded input distribution G. We remark that while the adversary cost increase model takes into account the ratio of the expected costs for the algorithms and for the adversary, the dual expresses a property of the expected value of the cost ratio. It is an open question to investigate the relation between the two cost models (see [4]).

5.2 Asymptotic Analysis for Paging

The adversary cost increase method is related to the resource augmentation method and to perturbation-based analysis in that it smooths away worst-case instances when such instances are only a small fraction of the input instances. In this and in the following sections, we consider similar scenarios that arise in an asymptotic sense, that is, when the fraction of bad instances tends to zero as the problem size increases.

The first application of this method is the asymptotic analysis of the the paging problem [3] when the working set size grows larger. In some sense, this application also illustrates a method to deal with the non-linearity implied by RAM. We denote by n the total number of pages in a sequence r, by k the cache size, and by m the length of a request sequence r (such notation is independent of that used in the previous and following sections). The cost of a paging strategy y on sequence r with a cache of size k is the number of times y evicts a page from the cache. It is easy to see that such cost function is equivalent to the number of page faults except in a constant additive factors. We will consider without loss of generality only sequences r where there are no consecutive requests for the same page.

Lemma 5.6 ([6]). *The randomized marking algorithm M is $(2H_k)$-competitive for the paging problem, where H_k is the kth harmonic number.*

At an intuitive level, the dilemma of competitive paging is due to cache sizes k that are only slightly smaller than the number n of pages. Indeed, if $n < k$, no paging algorithms ever evicts any page, and so any paging algorithm is optimal. Conversely, if $n \gg k$, then all paging algorithms, including the optimum, incur a high eviction rate, a situation which is often referred to as *thrashing* [24]. Since all algorithms thrash, the relative performance of on-line algorithms is good. It is only when n is slightly larger than k that the adversary can effectively leverage on its knowledge of the future to achieve a large performance ratio. Therefore, there should be only few values cache sizes that cause troubles to on-line paging strategies. In other words, if for any fixed request sequence r the adversary chooses a cache size k with probability $O(1/n)$, then there should be a deterministic paging algorithm with good expected relative performance. The rest of this section will formalize and prove this claim. In passing, we remark that a qualitatively similar claim is made within loose competitiveness [27], and we will use some of those results in our arguments. The results on loose competitiveness assumed that both the value of k and the paging algorithms should be probabilistic, and Theorem 5.8 will prove that a similar result holds when the algorithm is deterministic (and the value of k is chosen randomly).

In this analysis, the game is parameterized by the request sequence r. We set $h(k, y)$ to be the number of times that the replacement strategy y evicts a page when it process sequence r with a cache of size k, $\tilde{g}(k)$ to be the smallest number of evictions on sequence r when the cache size is k, b a constant that depends only on k, d any positive real, and $g(k) = \max\{c(k)\tilde{g}(x), m/n^d\} + b$. We will investigate the value of the game defined by h and g, and, to this end, we define $c(n) = 5 \ln \ln n$ ($n \geq 13$).

Lemma 5.7. *The randomized marking algorithm M is a $(1 + o(1))$-approximation algorithm in the adversary cost increase model $(\mathcal{B}, \lambda)$ where $\mathcal{B}$ is the uniform distribution over $\{1, 2, \dots, n\}$ and $\lambda = O(1)$.*

Proof Sketch. Use the previous lemma and $\delta(k) = 2H_k$ if $h(k, M) \geq g(k)$, $\delta(k) = 0$ otherwise. □

Theorem 5.8. *Consider the paging problem on a sequence r where the adversary chooses the cache size $k \in \{1, 2, \dots, n\}$ probabilistically subject to the additional constraint that $\Pr[k] = O(1/n)$. Then, for any such an adversary, there is an deterministic paging algorithm y whose expected relative performance is*

$$E\left[\frac{h(k, y)}{\max\{c(k)\tilde{g}(k), m/n^d\} + b}\right] \leq 1 + o(1) .$$

Proof. Combine the previous lemma with Corollary 5.4. □

5.3 High-probability Analysis

Theorem 5.1 and its applications (resource augmentation, perturbation-based analysis, loose competitiveness) show how few worst-cases instances are smoothed across more typical instances. In particular, any high probability analysis of randomized algorithms can be interpreted in terms of Theorem 5.1.

Proposition 5.9. *Let H be an randomized algorithm. Then, H is a $u(\mathcal{B}, H)$-approximation algorithm in the adversary cost increase model for all values of λ with the property that*

$$\lambda = O\left(\frac{\max_{x \in X} u(x, H)}{u(\mathcal{B}, H)} \Pr[u(x, H) \geq cu(\mathcal{B}, H)]\right) ,$$

where c is a constant.

Proof Sketch. Similar to the proof of Lemma 5.7. □

We now turn to consider asymptotic analysis when the utility is a ratio h/g. A possible application is when the instance x is a long request sequence. If the probability distribution $\mathcal{B}$ extracts the requests independently and with the same distribution, then the law of large numbers should show that the algorithm cost on x converges to the expected value with probability one. Therefore, as long as x is long enough, its cost is concentrated around the expected value, and a small values of δ should smooth out any deviations. The intuition is formalized in Theorem 5.10. We start with a definition.

Definition 5.3. A set X is said to be *based on s phases that are i.i.d. relative to $\mathcal{B}$* if and only if

- An instance $x \in X$ is a string obtained as the concatenation of s symbols $x_1, x_2, \ldots, x_s$ called *phases*,
- The costs h and g have the properties that $h(x, y) \geq 0$, $g(x) > 0$, and that $E_{\mathcal{B}}[h(x, y)]$ and $E_{\mathcal{B}}[g(x)]$ exist, are finite, and are positive,
- The profits are decomposable as $h(x, y) \leq \sum_{i=1}^{s} h(x_i, y)$ and $g(x) \geq \sum_{i=1}^{s} g(x_i)$,
- The distribution $\mathcal{B}$ over X has the property that each phase is extracted independently of all other phases and with the same distribution.

Theorem 5.10. *If the set of instance is based on s phases that are i.i.d. relative to $\mathcal{B}$ and $\lambda = O(1)$, then the game value in the adversary cost increase model is no more than* $\min_{y \in Y} E_{\mathcal{B}}[h(x_1, y)]/E_{\mathcal{B}}[g(x_1)] + o(1)$.

Actually, the previous theorem holds for $\lambda = o(1/E_{\mathcal{B}}[\delta(s)])$, but in general the law of large number does not allow us to derive any stronger bound on λ. We will consider the paging problem under the i.i.d. model, and show that constant approximation ratios in the adversary cost increase model are possible when λ grows exponentially with s. We use the same notation as in Section 5.2. We denote by n the total number of pages, by k the cache size, and by m the length of a request sequence x. We assume that $n = \alpha k$ for some constant $\alpha > 1$.

Definition 5.4 ([27]). Let x be a page request sequence. The first *phase* of x is the maximal prefix of x containing requests to at most k distinct pages, and, recursively, the ith phase is the maximal prefix of $x^{(i)}$ containing requests to at most k distinct pages, where $x^{(i)}$ is x with the first $i - 1$ phases removed.

Since algorithms start with an empty cache, no eviction occurs during the first phase. We will assume that the last phase contains requests to exactly k distinct pages, and it is easy to see that such an additional assumption only affects the additive constant. A phase of x other than the first will be called a *reasonable phase*. The first reasonable phase of x is the second phase of x, and so on. We will denote by P_x the number of reasonable phases of x.

Definition 5.5. A request sequence is *hard* if no page request has been previously requested in the current phase.

Such definition of hard sequences is less restrictive than the original one [18] and allows us to consider "difficult" sequences for the randomized marking algorithm [6] without knowing the algorithm's random choices. Equivalently, a hard sequence induces an equivalence class of infinitely many sequences, each of which causes the same cost to any deterministic marking algorithm. Furthermore, if we fix the random choices of the randomized marking algorithm M, we obtain a deterministic marking algorithm, and so any execution of M has exactly the same cost on all request sequences in the same equivalence class. By the previous assumptions, the length of a hard sequence x is $m = k(P_x + 1)$. The probability distribution $\mathcal{B}$ is to extract uniformly at random a hard sequence with P reasonable phases, and we will be interested in an asymptotic analysis as P grows.

Theorem 5.11. *Consider the paging problem with $n = \alpha k$ pages in the adversary cost increase model and let $\beta = (\alpha - 1)k/(\alpha k - 1)$. Let $\mathcal{B}$ is the uniform distribution over hard sequences with P reasonable phases and $\lambda = O\left(e^{\zeta P}\right)$ where $2\zeta < (\beta k)^2/(k-1)^2$. Then, the marking algorithm is an $(1 + o(1))$-approximation algorithm.*

Proof Sketch. We first derive that the expected number of *new requests* in a phase is $k(n-k)/(n-1)$. Then, we use the Hoeffding bound for the probability that the average number of new requests is far away from the average. This result and the choice of parameters yield that $E[\delta] = o(1)$. □

Remark. The previous theorem can be improved to $\lambda \simeq k^P$ by using $\mathrm{Var}(N_x)$ and a result by Hoeffding [9].

References

[1] Baruch Awerbuch, Yair Bartal, Amos Fiat, and Adi Rosén. Competitive non-preemptive call control. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, 1994)*, pages 312–320, New York, 1994. ACM.

[2] B. R. Barmish and H. Kettani. Monte Carlo analysis of resistive networks without a priori probability distributions. In *Proceedings of the International Symposium on Circuits and Systems*, 2000.

[3] L. A. Belady. A study of replacement algorithms for a virtual storage computer. *IBM Systems Journal*, 5(2):78–101, 1966.

[4] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[5] Michele Colajanni, Philip S. Yu, and Valeria Cardellini. Scalable web server systems: Architectures, models, and load balancing algorithms. In *Sigmetrics 2000 tutorials*, 2000.

[6] A. Fiat, R. M. Karp, L. A. McGeoch, D. D. Sleator, and N. E. Young. Competitive paging algorithms. *Journal of Algorithms*, 12:658–699, 1991.

[7] Amos Fiat. Private communication.

[8] Itzhak Gilboa and David Schmeidler. Maxmin expected utility with nonunique prior. *J. Math. Econom.*, 18(2):141–153, 1989.

[9] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.*, 58:13–30, 1963.

[10] L. Hurwicz. Some specification problems and application to econometric models. *Econometrica*, 19:343–344.

[11] Bala Kalyanasundaram and Kirk Pruhs. Speed is as powerful as clairvoyance. *J. ACM*, 47(4):617–643, 2000.

[12] Irving Kaplansky. A contribution to von Neumann's theory of games. *Ann. of Math. (2)*, 46:474–479, 1945.

[13] E. Koutsoupias and C. H. Papadimitriou. Beyond competitive analysis. *SIAM Journal on Computing*, 30(1):394–400, 2000.

[14] Stefano Leonardi, Alberto Marchetti-Spaccamela, Alessio Presciutti, and Adi Rosén. On-line randomized call control revisited. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 1998)*, pages 323–332, New York, 1998. ACM.

[15] Vincenzo Liberatore. Scheduling jobs before shut-down. *Nordic J. Comput.*, 7(3):204–226, 2000.

[16] Richard J. Lipton and Andrew Tomkins. Online interval scheduling. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, 1994)*, pages 302–311, New York, 1994. ACM.

[17] Lynn H. Loomis. On a theorem of von Neumann. *Proc. Nat. Acad. Sci. U.S.A.*, 32:213–215, 1946.

[18] Mark S. Manasse, Lyle A. McGeoch, and Daniel D. Sleator. Competitive algorithms for server problems. *J. Algorithms*, 11(2):208–230, 1990.

[19] C. J. H. McDiarmid and R. B. Hayward. Large deviations for Quicksort. *J. Algorithms*, 21(3):476–507, 1996.

[20] Christos H. Papadimitriou and Mihalis Yannakakis. On the approximability of trade-offs and optimal access of Web sources. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, 2000.

[21] Vilfredo Pareto. *Manuel d'economie politique*. Girard Briere, Paris, 1909.

[22] Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Comm. ACM*, 28(2):202–208, 1985.

[23] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. In *Proceedings of the Thirty-third Annual ACM Symposium on the Theory of Computing*, 2001.

[24] Andrew S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[25] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, New Jersey, 1944.

[26] Andrew Chi Chih Yao. Probabilistic computations: toward a unified measure of complexity (extended abstract). In *18th Annual Symposium on Foundations of Computer Science (Providence, R.I., 1977)*, pages 222–227. IEEE Comput. Sci., Long Beach, Calif., 1977.

[27] N. E. Young. *Competitive Paging and Dual-Guided On-Line Weighted Caching and Matching Algorithms*. PhD thesis, Princeton University, October 1991.

A Omitted Proofs

A.1 Main Duality Result

Proof of Lemma 3.3. Observe that $\min_{y \in Y} u(G, y) \leq \max_G \{\min_{y \in Y} u(G, y) : \Pr_G[x] \leq b_x\} = u_v^* \leq U$. □

A.2 Shut-Down Scheduling

The shut-down scheduling problem is as follows. A set of n job is to be scheduled on one machine starting at time 0 (the notation for n is independent of that in the previous and following sections). Job i is characterized by a length $l(i)$ and a profit $p(i)$ and the objective of the scheduler is to maximize the profit of the jobs fully completed before an initially unknown deadline D. Let V be the ratio of the maximum over the minimum profit. There is an $O(\log V)$-competitive algorithm for shut-down scheduling and no better algorithm is possible [15].

Proposition A.1. *Consider the shut-down scheduling problem with $k+1$ objectives $p_0, p_1, p_2, \ldots, p_k$, and suppose that it is further requested that a randomized algorithm be $O(k \log V)$-competitive in the metrics $p_1, p_2, \ldots, p_k$. Then, the best randomized algorithm is $\Theta(k \log V)$-competitive in the p_0 metric.*

Proof Sketch. Consider a set of instances and a deadline D with the property that only one job can complete before D. The optimal choice of that one job depends on the value of D. Jobs profit will range from 1 to V, a job i has non-unit profit in at most one metric, and for every metric there is a set of jobs whose profit increases exponentially. Let $\tilde{v}_i(x, y)$ be the profit accrued by the deterministic algorithm x in the ith metric and $v_i = \tilde{v} - 1/(k \log V)$. The probability distributions H_{v_i} will be uniform among the jobs that have large profit according to p_i. Therefore, for any deterministic choice x of the first job, there is at most one metric where the algorithm can achieve non-negligible profit, and that happens with probability $1/O(\log V)$. Observe that in the expression of Corollary 3.2 there is a fixed term proportional to $-(P-1)/(P \log V)$ and that there is only one profit p_i that is going to contribute a positive term of $1/\log V$. Thus, the lower bound is proven. As for the upper bound, choose a number i uniformily at random in $0, 1, \ldots, k$, and follow an $O(\log V)$-competitive algorithm for the p_i metric. □

A.3 Upper Bounded Adversaries

The formulation above implies that the matrix B in Proposition 4 is $B = \mathbf{b}\mathbf{e}^T - I$, and so $B^T = \mathbf{e}\mathbf{b}^T - I$. The following lemma establishes the values of eigenvalues and eigenvectors of B^T.

Lemma A.2. *The matrix $B^T = \mathbf{e}\mathbf{b}^T - I$ has the properties that $B^T\mathbf{e} = \lambda\mathbf{e}$ and $B^T\mathbf{f} = -\mathbf{f}$ for all vectors $\mathbf{f}$ with $\mathbf{b}^T\mathbf{f} = 0$.*

Corollary A.3. *The matrix B^T has a set of n linearly independent eigenvector, one of which is $\mathbf{e}$ and the other $n-1$ are a basis of the hyperplane orthogonal to $\mathbf{b}$.*

Proof. Observe that $\mathbf{e}$ is an eigenvector corresponding to the eigenvalue λ, and any vector in the hyperplane orhtogonal to $\mathbf{b}$ is an eigenvector corresponding to the eigenvalue -1. If $\mathbf{e}$ were dependent on the other basis vectors, then $\mathbf{e}$ would lie in the hyperplane orthogonal to $\mathbf{b}$, and so $\mathbf{e}^T\mathbf{b} = 0$, which contradicts the assumption $\mathbf{e}^T\mathbf{b} > 1$. Thus, the corollary is proven. □

Corollary A.4. *Any n-vector* $\mathbf{r}$ *can be expressed as* $\mathbf{r} = \alpha\mathbf{e} - \mathbf{f}$ *where* $\mathbf{f}$ *has the property that* $\mathbf{b}^T\mathbf{f} = 0$. *Moreover,* $\mathbf{b}^T\mathbf{r} = \alpha(1+\lambda)$ *and* $B^T\mathbf{r} = \lambda\alpha\mathbf{e} + \mathbf{f} = (1+\lambda)\alpha\mathbf{e} - \mathbf{r}$.

Proof of 5.1. Apply Corollary A.4 to Proposition 3.1 to obtain that

$$\begin{aligned} u_v^* &= \min\{u : A^T\mathbf{q} + B^T\mathbf{r} \leq u\mathbf{e}, \mathbf{e}^T\mathbf{q} = 1, \mathbf{q} \geq 0, \mathbf{r} \geq 0\} \\ &= \min\{u : A^T\mathbf{q} - \mathbf{r} + (1+\lambda)\alpha\mathbf{e} \leq u\mathbf{e}, \mathbf{e}^T\mathbf{q} = 1, \mathbf{b}^T\mathbf{r} = \alpha(1+\lambda), \mathbf{q} \geq 0, \mathbf{r} \geq 0\} \\ &= \min\{w + (1+\lambda)\alpha : A^T\mathbf{q} - \mathbf{r} \leq w\mathbf{e}, \mathbf{b}^T\mathbf{r} = \alpha(1+\lambda), \mathbf{e}^T\mathbf{q} = 1, \mathbf{q} \geq 0, \mathbf{r} \geq 0\} \\ &= \min\{w + \mathbf{b}^T\mathbf{r} : A^T\mathbf{q} - \mathbf{r} \leq w\mathbf{e}, \mathbf{e}^T\mathbf{q} = 1, \mathbf{q} \geq 0, \mathbf{r} \geq 0\} \end{aligned} \tag{5}$$

Since $\mathbf{q}$ can be interpreted as a probability distribution H, $\mathbf{r}$ as $r : X \to \mathbb{R}_0^+$, and $\mathbf{b}^T\mathbf{r} = (1+\lambda)E_{\mathcal{B}}[r]$, the theorem follows. □

Proof of 5.2. Combine Theorem 5.1 with Lemma 3.3. □

First, we need the following

Lemma A.5 ([17]). *Let* $\mathcal{B}$ *be a mixed strategy for player I and* H *a mixed strategy for player II. Then,* $\min_{y \in Y} u(\mathcal{B}, y) \leq u(\mathcal{B}, H)$.

Proof of Corollary 5.3. Consider a feasible solution $\mathbf{q} = (q_x)$, $\mathbf{f} = (f_x)$, w, and α to the program (5). Assume first that $w \neq 0$ and let $\delta(x) = r_x/w$. Thus, the program can be rewritten as

$$\begin{aligned} u_v^* = \min \quad & w\left(1 + \sum_{x \in X} b_x \delta(x)\right) \\ \text{s.t.} \quad & \sum_{y \in Y} h(x,y) q_y \leq w g(x)(1 + \delta(x)) \quad && x \in X \\ & \sum_{y \in Y} q_y = 1 \\ & \delta(x) \geq 0 && x \in X \\ & q_y \geq 0 && y \in Y \end{aligned}$$

which is equal to $1 + (1+\lambda)E_{\mathcal{B}}[\delta]$ times the value of the unconstrained program (1) once the adversary costs $g(x)$ have been replaced by $g_\delta(x)$.

We now prove by contradiction that $w \neq 0$. Assume that $w = 0$ and let H be the randomized algorithm defined by a probability vector $\mathbf{q}$. Since $u(x,y) > 0$, we have that $r_x \geq \sum_{y \in Y} u(x,y) q_y \geq 0$, and since $b_x \geq 0$ we can assume without loss of generality that $r_x = \sum_{y \in Y} u(x,y) q_y$, and so the objective value $\mathbf{b}^T\mathbf{r}$ is equal to $(1+\lambda)u(\mathcal{B}, H)$. Lemma A.5 entails that the objective value is $(1+\lambda)u(\mathcal{B}, H) \geq \min_{y \in Y}(1+\lambda)u(\mathcal{B}, y)$. It follows that there is an optimal solution with $w = 0$ and $\mathbf{q}$ equal to zero except in one unit component. Therefore, there is an optimal solution where $\mathbf{r}^T$ is a row of A that minimizes $\mathbf{b}^T\mathbf{r}$. By duality, $\mathbf{b}^T\mathbf{r} = \max\{u : A\mathbf{p} \geq u\mathbf{e}, \mathbf{p} \leq \mathbf{b}, \mathbf{e}^T\mathbf{p} = 1, \mathbf{p} \geq 0\}$. In particular, the constraint corresponding to the row $\mathbf{r}^T$ gives $\mathbf{p}^T\mathbf{r} \geq \mathbf{b}^T\mathbf{r}$. However, $\mathbf{r} > 0$ and $\mathbf{p} \leq \mathbf{b}$, and so $\mathbf{p} = \mathbf{b}$. Therefore, $\lambda = 0$, which contradicts the hypothesis. It is then proved that $w \neq 0$, which completes the proof. □

Proof of Corollary 5.4. Combine Definition 5.2 and Corollaries 5.2 and 5.3. □

A.4 RAM

Proof of Corollary 5.5. Any $\delta \geq 0$ is feasible in the adversary cost augmentation model, but only the δ's that correspond to resource values are feasible in the RAM trade-off problem. □

A.5 Asymptotic Paging

We defined $c$more precisely as follows:

$$c(n) = \begin{cases} 2\ln n - 2/5 & \text{if } 1 \leq n \leq 12 \\ 5\ln\ln n & \text{if } n \geq 13 \end{cases}$$

Lemma A.6. *The functions $c(n)$ and $2\ln n - c(n)$ are increasing.*

Lemma A.7 ([27]). *Let M be randomized marking algorithm. Let $c(n)$ be any function with the property that both $c(n)$ and $2\ln n - c(n)$ are increasing. Then, for any $d > 0$ and for any b that depends only on k, $h(k, H) \leq g(k)$ except for $O((d+1)\exp(1 - c(n)/2)n\ln n)$ values of k in the interval $X = \{1, 2, \ldots, n\}$.*

Proof. The main idea of the proof is that there are only few strategies for the adversary (i.e., few values of k) that cause trouble to M. Thus, we chooses a function δ to amortize those values of k while keeping $E[\delta]$ small. First, we set

$$\delta(k) = \begin{cases} 2H_k & \text{if } h(k, M) \geq g(k) \\ 0 & \text{otherwise} \end{cases} .$$

Lemma 5.6 gives that $h(x, M) \leq 2H_k\tilde{g}(x)$, and so $h(x, M) \leq 2H_k\tilde{g}(x) \leq 2H_k g(x) = g_\delta(x)$ whenever $h(k, M) \geq c(k)g(k)$. Meanwhile, if $h(k, M) < g(k)$ then $h(k, M) < g_\delta(k)$, so that on the whole, the game value is 1 when the adversary cost g is inflated by δ. We will prove that $(1+\lambda)E[\delta]$ can be made arbitrarily small as n increases, which entails that M is a $(1+o(1))$-approximation algorithm in the adversary cost increase model. By Lemma A.7, there is a constant c such that

$$E_{\mathcal{B}}[\delta] \leq 2c(d+1)H_k \exp\left(1 - \frac{c(n)}{2}\right)\ln n \leq 2c(d+1)H_n \exp\left(1 - \frac{c(n)}{2}\right)\ln n ,$$

and so

$$\lim_{n\to\infty}(1+\lambda)E_{\mathcal{B}}[\delta] = \lim_{n\to\infty} 2c(d+1)\exp\left(1 + \ln(1+\lambda) + \ln H_n + \ln\ln n - \frac{c(n)}{2}\right) .$$

Observe that

$$\begin{aligned} \lim_{n\to\infty} 2\ln(1+\lambda) + 2\ln H_n + 2\ln\ln n - c(n) &\leq \lim_{n\to\infty} 2\ln(1+\lambda) + 2\ln(\ln + 1) + 2\ln\ln n - c(n) \\ &\leq \lim_{n\to\infty} 2\ln(1+\lambda) + 2\ln(2\ln n) + 2\ln\ln n - c(n) \\ &= \lim_{n\to\infty} 2\ln(1+\lambda) + 2\ln 2 + 4\ln\ln n - c(n) \\ &= -\infty , \end{aligned}$$

and so $\lim_{n\to\infty}(1+\lambda)E_{\mathcal{B}}[\delta] = 0$, which concludes the proof. □

Proof of Proposition 5.9. Let

$$\delta(x) = \begin{cases} \dfrac{\max_{x \in X} u(x,H)}{u(\mathcal{B},H)} & \text{if } u(x,H) \geq c u(\mathcal{B},H) \\ 0 & \text{otherwise} \end{cases}$$

to prove the claim. □

As an example, we observe that

Theorem A.8 ([19]). *Let Q_s be the a random variable that represents the number of comparisons executed by the randomized quicksort algorithm and $q_s = E[Q_s]$. Then,*

$$\Pr\left[\left|\frac{Q_s}{q_s} - 1\right| > \epsilon\right] = s^{-(2+o(1))\epsilon \ln \ln s} .$$

Corollary A.9. *The randomized quicksort algorithm runs achieves $O(s \log s)$ utility in the adversary cost increase model when $\lambda = s^{O(\ln \ln s)}$.*

A.6 Renewal Instances

Lemma A.10. *If an instance x is extracted according to the distribution $\mathcal{B}$ from a set based on s phases that are i.i.d. relative to $\mathcal{B}$, $h(x_i, y)$ and $g(x_i)$ are independent and identically distributed random variables with finite mean ($1 \leq i \leq s$).*

Proof. The last condition in Definition 5.3 immediately implies that $h(x_i, y)$ and $g(x_i)$ are independent and identically distributed. Moreover, the third condition and linearity of expectation give $E_{\mathcal{B}}[h(x_i, y)] \leq E_{\mathcal{B}}[h(x, y)]/s < \infty$, which shows that $h(x_i, y)$ has a finite mean. The same argument for $E_{\mathcal{B}}[g(x_i)]$ concludes the proof. □

Lemma A.11. *If an instance x is extracted according to $\mathcal{B}$ from a set based on s phases that are i.i.d. relative to $\mathcal{B}$, then*

$$h(x,y) \leq \frac{E_{\mathcal{B}}[h(x_1,y)]}{E_{\mathcal{B}}[g(x_1)]} g(x)(1+\delta(x)) ,$$

where $\delta(x) \geq 0$ and $\lim_{s\to\infty} E_{\mathcal{B}}[\delta(x)] = 0$.

Proof. Since the $h(x_i, y)$'s and $g(x_i)$'s are independent and identically distributed random variables, we can apply the law of large numbers to obtain that $\lim_{s\to\infty} h(x,y)/s \leq E_{\mathcal{B}}[h(x_1,y)]$ and $\lim_{s\to\infty} g(x)/s \geq E_{\mathcal{B}}[g(x_1)] > 0$ with probability one. Therefore,

$$\lim_{s\to\infty} \frac{h(x,y)}{g(x)} = \lim_{s\to\infty} \frac{h(x,y)}{s} \frac{s}{g(x)} \leq \frac{E_{\mathcal{B}}[h(x_1,y)]}{E_{\mathcal{B}}[g(x_1)]} \qquad \text{(with probability one)},$$

or, equivalently,

$$\frac{h(x,y)}{g(x)} \leq \frac{E_{\mathcal{B}}[h(x_1,y)]}{E_{\mathcal{B}}[g(x_1)]} + \eta(x) ,$$

where $\lim_{s\to\infty} \eta(x) = 0$ with probability one. In conjunction with $E_{\mathcal{B}}[h(x_1,y)] > 0$, this leads to

$$h(x,y) \leq \frac{E_{\mathcal{B}}[h(x_1,y)]}{E_{\mathcal{B}}[g(x_1)]} g(x)(1+\delta(x)) ,$$

where

$$\delta(x) = \frac{E_{\mathcal{B}}[g(x_1)]}{E_{\mathcal{B}}[h(x_1, y)]} \max\{0, \eta(x)\} \geq 0 \, .$$

Observe that

$$|\delta(x)| = \delta(x) = \frac{E_{\mathcal{B}}[g(x_1)]}{E_{\mathcal{B}}[h(x_1, y)]} \max\{0, \eta(x)\} \leq \frac{E_{\mathcal{B}}[g(x_1)]}{E_{\mathcal{B}}[h(x_1, y)]} |\eta(x)| \, .$$

Therefore, the events $E_\eta = \{\forall \epsilon > 0 \exists s_\epsilon > 0 \forall s \geq s_\epsilon : |\eta(x)| \leq \epsilon\}$ and $E_\delta = \{\forall \epsilon > 0 \exists s_\epsilon > 0 \forall s \geq s_\epsilon : |\delta(x)| \leq \epsilon E_{\mathcal{B}}[h(x_1, y)]/E_{\mathcal{B}}[g(x_1)]\}$ have the property that $E_\eta \subseteq E_\delta$. Since $1 = \Pr E_\eta \leq \Pr E_\delta \leq 1$, we have $\Pr E_\delta = 1$. However, E_δ is the intersection of infinitely many events $\{\forall s \geq s_\epsilon : \delta(x) \leq \epsilon\}$, and so any such event must occur with probability one. It follows that for any $\epsilon > 0$, there is an s such that when x is extracted from a set based on s phases that are i.i.d. relative to $\mathcal{B}$, $E[\delta(x)] \leq \epsilon$, and so $\lim_{s\to\infty} E[\delta(x)] = 0$, which concludes the proof. □

Proof of 5.10. The theorem is immediately proved by the combination of Corollary 5.3 and Lemma A.11. □

A.7 Renewal Paging

Definition A.1 ([27]). A request is *new* within a reasonable phase if it has not been requested yet in the current phase or in the previous one.

Lemma A.12 ([27]). *The first request of a reasonable phase is new.*

Lemma A.13 ([27]). *The expected cost of the marking algorithm during the ith reasonable phase is no more than $N_x(i)(H_k - H_{N_x(i)} + 1)$, where H_k is the kth harmonic number. Moreover, $g(x) \geq N_x P/2$.*

Lemma A.14 ([27]). *The harmonic numbers have the property that $H_n - H_k \leq \ln(n/k)$ for all $n > k \geq 1$.*

Let $N_x(i)$ be the number of new requests during the ith reasonable phase of x $(1 \leq i \leq P)$, and $N_x = \sum_{i=1}^{P} N_x(i)/P$. Let M be the randomized marking algorithm, which is a distribution over the set Y of deterministic algorithms. Let $h(x, y)$ be the cost of the deterministic algorithm y over x and $g(x)$ be the optimum cost (calculated as in [3]).

Lemma A.15. *On a hard sequence x of length $m = k(P+1)$, $h(x, M) \leq 2g(x)(1 - \ln(N_x/k))$.*

We consider a function

$$\delta(x) = \begin{cases} \ln k & \text{if } N_x \leq \beta k/2 \\ 1 & \text{otherwise} \end{cases} ,$$

where $\beta = (\alpha - 1)k/(\alpha k - 1)$ is bounded by a constant independent of n, k, and P. Then, if $N_x \leq \beta k/2$, Lemma A.15 leads to $h(x, M) \leq 2g(x)(1 + \ln k - \ln N_x) \leq 2g_\delta(x)$, and if $N_x > \beta k/2$, then $h(x, M) < 2(1 + \ln(2/\beta))g_\delta(x)$. As a result, the marking algorithm achieves a constant competive ratio if the optimum cost is inflated by a $1 + \delta(x)$ factor. It remains to estimate $(1 + \lambda)E[\delta]$.

Lemma A.16. *Suppose that pages are requested uniformily at random. Then, for $1 \leq i \leq P$, the $N_x(i)$'s are independent and identically distributed, have finite variance, and $E[N_x(i)] = k(n - k)/(n - 1)$.*

Henceforth, we will use the notation $\mu = E[N_x(1)] = E[N_x] = \beta k$.

Lemma A.17. *Let x be a hard request sequence with P phases. Then,*

$$\Pr\left[N_x < \frac{k\beta}{2}\right] \leq \exp\left(-\frac{P}{2}\left(\frac{\beta k}{k-1}\right)^2\right)$$

where $\beta = (\alpha - 1)k/(\alpha k - 1)$.

Proof of Lemma A.15. The cost of the marking algorithm on the first phase of x is naught. Therefore,

$$\begin{aligned} h(x, M) &\leq \sum_{i=1}^{P} N_x(i)(H_k - H_{N_x(i)} + 1) && \text{(by Lemma A.13)} \\ &\leq \sum_{i=1}^{P} N_x(i)\left(1 + \ln k - \ln N_x(i)\right) && \text{(by Lemma A.14)} \\ &= PN_x(1 + \ln k) - \sum_{i=1}^{P} N_x(i) \ln N_x(i) \ . \end{aligned}$$

Observe that

$$\begin{aligned} -\sum_{i=1}^{P} N_x(i) \ln N_x(i) &= -\frac{N_x P}{\log e} \sum_{i=1}^{P} \left(\frac{N_x(i)}{PN_x} \log \frac{N_x(i)}{PN_x} + \log PN_x\right) \\ &= \frac{N_x P}{\log e}\left(\mathcal{H}\left(\frac{N_x(1)}{PN_x}, \frac{N_x(2)}{PN_x}, \ldots, \frac{N_x(P)}{PN_x}\right) - \log(PN_x)\right) \ . \end{aligned}$$

where $\mathcal{H}$ is the entropy function. Recall that under the constraint that $\sum_{i=1}^{P} N_x(i) = PN_x$, the entropy $\mathcal{H}$ is maximized if all the $N_x(i)$'s are equal, and therefore

$$h(x, M) \leq PN_x - PN_x \ln \frac{N_x}{k} = PN_x\left(1 - \ln \frac{N_x}{k}\right) \ ,$$

which, combined with the previous lemma, proves the claim. □

Proof of Lemma A.16. The first request of a reasonable phase is new by Lemma A.12. Denote by A the set of pages requested in the phase other than the first requested page. Then, $|A| = k - 1$ and new requests correspond to the pages of A that were not requested during the previous phase. Hence, the number of new requests is equal to the hypergeometric random variable of extracting $k-1$ objects from $n-1$ of which $n-1-k$ belong to a target set. Since the process is probabilistic, $N_x(i)$ is independent of the identity of the requests during the previous phases. Moreover, the parameters of the hypergeometric distribution are independent of i and so the $N_x(i)$'s are identically distributed. Finally, the well known expression for the mean and variance of a hypergeometric random variable concludes the proof. □

Theorem A.18 ([9]). *Let $N(1), N(2), \ldots, N(P)$ be independent random variables with finite mean, finite variance, and with the property that $a \le N(i) \le b$ $(1 \le i \le P)$. Let $N = \sum_{i=1}^{P} N(i)/P$. Then,*

$$\Pr[N - E[N] \ge t] \le e^{-2Pt^2/(b-a)^2} \; .$$

Proof of Lemma A.17. Let $\widetilde{N_x} = k - N_x$ and $\widetilde{N_x}(i) = k - N_x(i)$. Then, $E[\widetilde{N_x}(i)] = k - \mu$ and $\mathrm{Var}[\widetilde{N_x}(i)] = \mathrm{Var}[N_x(i)] < \infty$. We then have

$$\begin{aligned}
\Pr\left[N_x < \frac{k\beta}{2}\right] &= \Pr\left[\widetilde{N_x} \ge \frac{k(2-\beta)}{2}\right] \\
&= \Pr\left[\widetilde{N_x} - (1-\beta)k \ge \frac{\beta k}{2}\right] \\
&\le \Pr\left[\widetilde{N_x} - (k-\mu) \ge \frac{\beta k}{2}\right] \\
&\le \exp\left(-\frac{P}{2}\left(\frac{\beta k}{k-1}\right)^2\right) \qquad \text{(by Theorem A.18)}\; ,
\end{aligned}$$

which proves the claim. □

Proof of Theorem 5.11. We have shown that the marking algorithm achieves a constant game value if the adversary cost is inflated by a factor of $1 + \delta$. It remains to estimate $(1 + \lambda)E_{\mathcal{B}}[\delta]$. Recall that α, n, and k are independent of P. We have that

$$(1+\lambda)E_{\mathcal{B}}[\delta] \le (1+\lambda)\exp\left(-\frac{P}{2}\left(\frac{\beta k}{k-1}\right)^2\right)\ln k = O\left(\exp\left(\frac{P}{2}\left(2\zeta - \left(\frac{\beta k}{k-1}\right)^2\right)\right)\right) = o(1)\; .$$

By using this expression in Corollary 5.3, we obtain the theorem. □

A.8 Lower-bounded Adversaries

We now turn to an adversary that chooses x with probability at least b_x. A natural constraint on the adversary is that x is chosen with probability at least b_x. The adversary can satisfy this constraint by choosing an x with probability b_x and following a worst-case (mixed) strategy with probability $1 - \sum_{x \in X} b_x$. We have proved that in fact the adversary has no better strategy than that. The implication for randomized algorithms is a criterion whereby the algorithm will attempt to minimize a convex combination of worst-case and average-case cost.

In this section, we will show that, in terms of H's strategy, such constrained game is equivalent to a different game, defined below, that embodies H's attitude toward *risk*. First, however, we need to make some reminders and considerations about rational behavior under risk. Suppose that player II seeks to find a mixed strategy H that $\min_H \max_{x \in X} u(x, Y)$, which is the classical formulation of a two-person zero-sum game (2). In this case, player II is extremely averse to risk in that its target strategy H must achieve minimum cost u against an adversary that adopts a worst case strategy x. Suppose now that player 2 is only interested in the average-case cost of H, that is, the player seeks a strategy H that $\min_H u(\mathcal{B}, H)$. In this case, the player can be thought of being risk-seeking, in that if G deviates from $\mathcal{B}$, then H will not be optimal in general, and potentially arbitrarily worse than its average-case performance. Thus, worst-case performance and average-case performance appear to be two extreme cases in the spectrum of criteria that a rational decision maker can adopt under risk. The two extreme criteria can also be blended, for example, through a convex combination, so that player II seeks a H with

$$\min_H \left(\left(\lambda \max_{x \in X} u(x, H) \right) + (1 - \lambda) u(\mathcal{B}, H) \right) , \tag{6}$$

where $0 \leq \lambda \leq 1$. In the case where $\lambda = 1$, the criterion reduces to worst-case performance, and in the case where $\lambda = 0$, the criterion reduces to average-case performance. In this section, we will show that player II optimal strategy under criterion (6) is the same as that for a game where player 1 has payoff $u(x, y)$ but is constrained to use each deterministic strategy with probability at least b_x.

We now introduce definitions and certain assumptions that hold without loss of generality. First, let $\mathbf{b} = (b_x)$ and assume without loss of generality that $0 \leq \mathbf{b}$. Observe that if $\mathbf{e}^T\mathbf{b} = 1$, then the vector $\mathbf{b}$ forces $p_x = b_x$. We will assume without loss of generality that $\mathbf{e}^T\mathbf{b} < 1$, and let $\lambda = 1 - \mathbf{e}^T\mathbf{b} > 0$. The quantity λ can be interpreted as a measure of how far away $\mathbf{b}$ is from a probability distribution. Indeed, observe that if $\lambda = 0$, then the adversary is completely forced to use a certain mixed strategy, whereas $\lambda \simeq 1$ leaves ample room of maneuver to the adversary. Thus, λ can also be interpreted as a measure of the weakness of the constraint on the adversary. Finally, we let $\mathbf{b}' = \mathbf{b}/(\mathbf{e}^T\mathbf{b}) = \mathbf{b}/(1 - \lambda)$ and interpret $\mathbf{b}'$ as a probability distribution $\mathcal{B}$ over X. We impose the lower bounds on the probabilities by letting $Y_v = X$ and

$$v(x, y) = \begin{cases} 1 - b_x & \text{if } x = y \\ -b_x & \text{otherwise} \end{cases} \tag{7}$$

As a result, the matrix B in Theorem 3.1 is the $n \times n$ matrix $I - \mathbf{b}\mathbf{e}^T$.

For comparison, recall that

Definition A.2 ([12]). A mixed strategy is said to be *completely mixed* if all deterministic strategies are chosen with positive probability.

Therefore, if $\mathbf{b} > 0$, every feasible strategy is completely mixed and the game is completely mixed.

Proposition A.19. *Let* $\lambda = 1 - \sum_{x \in X} b_x$. *Then,*

$$\begin{aligned} \max_G \left\{ \min_{y \in Y} u(G, y) : \Pr_G[x] \geq b_x \right\} &= \max_G \left\{ \min_{y \in Y} ((1 - \lambda) u(\mathcal{B}, y) + \lambda u(G, y)) \right\} \\ &= \min_H \left\{ (1 - \lambda) u(\mathcal{B}, H) + \max_{x \in X} \lambda u(x, H) \right\} . \end{aligned}$$

Proof. Consider any feasible solution $\mathbf{p}$ and express it as $\mathbf{p} = \mathbf{b} + \mathbf{c}$. Observe that $\mathbf{e}^T\mathbf{c} = 1 - \mathbf{e}^T\mathbf{b} = \lambda$ and let $\mathbf{c}' = \mathbf{c}/\lambda$. Therefore, $u_v^* = \max\{u : (1 - \lambda)A\mathbf{b}' + \lambda A\mathbf{c}' \geq u\mathbf{e}, \mathbf{e}^T\mathbf{c}' = 1, \mathbf{c}' \geq 0\}$, which proves the first part of the claim. Apply duality to this unconstrained game to obtain the second equality. □

Remark. The proof above gives a direct proof of the second equality. However, that equality can also be proven as an application of Proposition 3.1 with an argument similar to the one used for upper-bounded adversaries (eigenvalues of the B matrix). The proof above can be thought of being more direct in that it only uses linearity of expectation in a disguised manner.