A Polyhedral Approach to Some Max-min Problems

Lisa Hellerstein[*]   Thomas Lidbetter[†]

Abstract

We consider a max-min variation of the classical problem of maximizing a linear function over the base of a polymatroid. In our problem we assume that the vector of coefficients of the linear function is not a known parameter of the problem but is some vertex of a simplex, and we maximize the linear function in the worst case. Equivalently, we view the problem as a zero-sum game between a maximizing player whose mixed strategy set is the base of the polymatroid and a minimizing player whose mixed strategy set is a simplex. We show how to efficiently obtain optimal strategies for both players and an expression for the value of the game. Furthermore, we give a characterization of the set of optimal strategies for the minimizing player. We consider four versions of the game and discuss the implications of our results for problems in search, sequential testing and queueing.

1   Introduction

A well understood problem in combinatorial optimization is that of maximizing a linear function over a polymatroid. As shown in Edmonds (1970), the solution of the problem is given by a simple greedy algorithm whose output is some vertex of the base of the polymatroid. A similar algorithm can be used to minimize a linear function over a contrapolymatroid. (All concepts will be defined precisely in Section 2).

[*]Department of Computer Science and Engineering, New York University Tandon School of Engineering, New York, New York 11201, lisa.hellerstein@nyu.edu

[†]Department of Management Science and Information Systems, Rutgers Business School, Newark, NJ 07102, tlidbetter@business.rutgers.edu

Many optimization problems can be viewed as a special case of this problem. The general approach is to associate some "performance vector" with each possible choice of policy for the problem in question, then to show that the convex hull B of these vectors is the base of a polymatroid or a contrapolymatroid. The objective function is then expressed as a linear function over B, so that it can be optimized using the classic greedy algorithm.

One example of such a problem is the single machine scheduling problem $1 \| \sum w_j C_j$ of choosing what order to process a finite set of jobs with given processing times to minimize their weighted sum of completion times. The solution to this problem is given by a simple index rule known as *Smith's Rule* (Smith, 1956). Queyranne (1993) later showed that this solution can be derived by showing that the convex hull B of the vectors of completion times of jobs is the base of a contrapolymatroid, and observing that the weighted sum of completion times is a linear function over B. See also Queyranne and Schulz (1994). The problem is equivalent to a search problem considered in Bellman (1957) [Chapter III, Exercise 3, p.90], where a target in located in one of a finite number of boxes with search costs according to a known probability distribution and the aim is to locate the target in minimal expected cost.

Stadje (1995) introduced a single machine scheduling problem where a finite set of jobs each has a given reward and a given probability of being successfully processed. If the machine fails to process a job it cannot process any further jobs. The problem is to choose in which order to schedule the jobs to maximize the total expected reward. Like the first scheduling problem described above, this problem is also solved by a simple index rule. More recently, the same problem was studied independently under the nomenclature of the *unreliable jobs problem* by Agnetis et al. (2009), who considered performance vectors corresponding to the probabilities that the jobs are successfully processed under a given schedule, and showed the convex hull of these vectors is the base of a polymatroid. Hence, by writing the objective of the problem as a linear function over this polyhedron, the index rule for this problem can also be derived from the greedy algorithm for optimizing a linear function over a polymatroid. Kodialam (2001) had previously studied this same polymatroid to solve a different, but related problem in sequential testing.

By considering so-called conservation laws, Federgruen and Groenevelt (1988) showed that the performance space of several multiclass queueing systems have a polymatroid structure, and this was extended to many other queueing problems by Shanthikumar and Yao (1992). Some cases of the well-known $c\mu$ priority rule for queueing can be derived by con-

sidering a linear optimization problem over the base of a polymatroid.

In this paper we introduce and solve a new max-min version of the classic problem of maximizing a linear function over a polymatroid. It generalizes many problems on search games, sequential testing and queueing; some known and some new. In particular, we solve a case of the *weighted search game* introduced by Yolmeh and Baykal-Gürsoy (2021), where a Searcher aims to minimize a weighted time to find a target hidden among a finite number of locations with varying weights and search times. We extend the weighted search game to incorporate the variable speed search paradigm of Alpern and Lidbetter (2014), and give a solution to this problem too. We show that the solution of a search and rescue game introduced by Lidbetter (2020) also follows from a corollary of our main results; furthermore we solve a more elaborate search and rescue game.

We show that our approach yields an alternative solution to a problem in sequential testing previously solved by Kodialam (2001) and Condon et al. (2009), in which operators sequentially perform tests on some tuples until obtaining a negative test, and the objective is to find a randomized routing of tuples to maximize throughput.

Our main problem can also be used to address some max-min (or min-max) multiclass queueing problems, which, as far as we know, have not previously been considered in the literature. Although there are several possible applications, we consider one concrete example of a multiclass queueing problem in which one server processes jobs with exponentially distributed service times that arrive according to a Poisson process. The objective is to choose a randomized priority rule to minimize the maximum expected holding cost of any job class in the steady state of the system. A solution to this problem follows from our results.

In Section 2 we review the notion of a polymatroid and the classic greedy algorithm of Edmonds (1970). We then describe and solve our main problem, framing it as a zero-sum game between a maximizer whose pure strategies are the set of vertices x of the base of a polymatroid and a minimizer whose pure strategies are the coordinates i. The payoff of the game is $w_i x_i$ for some fixed positive weights w (in contrast to the classic problem where the objective is $w^T x$). We give optimal strategies for both players and an expression for the value of the game. Furthermore, we give a complete characterization of the set of optimal strategies for Player 2. The value of the game and optimal strategies for both players can be found in strongly polynomial time (in the dimension of the polymatroid). We show that in

some special cases, the value of the game can be found particularly quickly. In Section 3, we consider some variations of our game involving contrapolymatroids and min-max objectives.

We apply our general result to several search games in Section 4, some of which have known solutions and some of which do not. These are two-person zero-sum games, where one player hides a "target", which the other player must locate. See Alpern and Gal (2003) or Hohzaki (2016) for an overview on the search games literature. We also make a link to a problem in sequential testing in Section 5, showing that it is a special case of the problem addressed in this paper. In Section 6, we discuss further implications of our results in the field of queueing theory, giving an example of a min-max problem in queueing theory whose solution follows from this work.

Finally, in Section 7, we consider a special case of our game in which the payoff function satisfies a certain monotonicity property. For this case, we give an efficient procedure that implements an optimal strategy for Player 1. The support of this strategy is of exponential size, and the procedure does not output an explicit representation of it as a convex combination of pure strategies. Instead, the procedure can be used to efficiently generate a pure strategy, drawn from the support of this optimal strategy with appropriate probability.

2 Problem Statement and Solution

In this section we define and solve our main problem, then consider some special cases and variations. But first, we review the definition of polymatroids and some elementary facts about them.

2.1 Review of Elementary Polymatroid Theory

Recall that a function $f : 2^V \to \mathbb{R}$ is submodular if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq V$ and $g : 2^V \to \mathbb{R}$ is supermodular if $g(A) + g(B) \leq g(A \cup B) + g(A \cap B)$ for all $A, B \subseteq V$.

For the rest of this section we assume that $f : 2^V \to \mathbb{R}_+$ is a non-negative, non-decreasing (with respect to set inclusion) submodular function with $f(\varnothing) = 0$, where $V = [n] \equiv \{1, \ldots, n\}$ for some positive integer n. (We set $[n] = \varnothing$ if $n = 0$.) We assume that the values $f(S)$ are given by an oracle. Let $P(f)$ be the polymatroid associated with f, given

4

by

$$P(f) = \{x \in \mathbb{R}_+^n : x(S) \le f(S) \text{ for all } S \subseteq V\},$$

where $x(S) \equiv \sum_{j \in S} x_j$. We first review the problem of maximizing a linear function $w^T x$ over $x \in P(f)$, where $w \in \mathbb{R}_+^n$ is a constant. Let $\sigma : V \to V$ be a permutation (or bijection) of V such that $w_{\sigma(1)} \ge \cdots \ge w_{\sigma(n)}$. The classic solution to the problem, given in Edmonds (1970) is the point x^σ given by

$$x_{\sigma(j)}^\sigma = f(\{\sigma(1), \ldots, \sigma(j)\}) - f(\{\sigma(1), \ldots, \sigma(j-1)\}), \quad j = 1, \ldots, n. \tag{1}$$

Notice that for any w and σ, we have $x^\sigma(V) = f(V)$, so an equivalent problem is to maximize $w^T x$ over the *base polyhedron $B(f)$* of f, given by

$$B(f) = \{x \in P(f) : x(V) = f(V)\}.$$

The vertices of $B(f)$ are given by all points x^σ defined by (1), as σ ranges over the set $\Sigma \equiv \Sigma(V)$ of all possible permutations of V.

Later, we will use the following fact, which is easy to verify.

Lemma 1 If $w_i > w_j$, then for any x^σ that maximizes $w^T x$, there exists some $\tilde{\sigma}$ such that $x^\sigma = x^{\tilde{\sigma}}$ and $\tilde{\sigma}^{-1}(i) < \tilde{\sigma}^{-1}(j)$ (that is, i precedes j in $\tilde{\sigma}$).

We note that in giving running times, we assume that it takes only constant time to answer an oracle query.

2.2 The Main Problem

The problem we consider in this paper is that of finding some $x \in B(f)$ to maximize $\min_j w_j x_j$. Equivalently, we consider a zero-sum game in which a pure strategy for Player 1 (the maximizer) is a permutation σ of V (or, equivalently, a vertex x^σ of $B(f)$) and a pure strategy for Player 2 (the minimizer) is a direction $j \in V$. For a given pair of pure strategies σ and j, the payoff is given by

$$P_{f,w}(\sigma, j) \equiv w_j x_j^\sigma.$$

We will usually drop the f and w from the subscript of P. We denote this game by $\Gamma_{max}(f, w)$. (In Section 3 we will consider a version of the game where Player 1 is the minimizer.) A

5

mixed strategy for Player 1 corresponds to a point x of $B(f)$ and the expected payoff of such a strategy against a pure strategy j of Player 2 is $w_j x_j$.

A mixed strategy for Player 2 is a randomized choice of directions, where each $j \in V$ is chosen with some probability $\theta_j \geq 0$, where $\sum_{j=1}^{n} \theta_j = 1$. For such a mixed strategy, the payoff against a strategy x of Player 1 is

$$\sum_{j=1}^{n} \theta_j w_j x_j = x^T y,$$

where $y = \sum_{j=1}^{n} \theta_j w_j e^j$, and e^j is the jth coordinate vector.

Equivalently, we may consider a mixed strategy for Player 2 as a point y of the simplex

$$C = \left\{ \sum_{j=1}^{n} \theta_j w_j e^j : \sum_{j=1}^{n} \theta_j = 1 \text{ and } \theta_i \geq 0 \text{ for all } j = 1, \ldots, n \right\},$$

so that a pure strategy for Player 2 is a vertex $w_j e^j$ of C. In a small abuse of our notation, we write $P(x, y)$ for the expected payoff $x^T y$ when Player 1 uses strategy x and Player 2 uses strategy y. When one player uses a pure strategy and the other uses a mixed strategy, we extend the use of P in the natural way.

Since each player has a finite number of pure strategies, the game has optimal mixed strategies and a value v, by the minimax theorem for zero-sum games, where

$$v = \max_{x \in B(f)} \min_{j} P(x, j) = \min_{y \in C} \max_{\sigma \in \Sigma} P(\sigma, y).$$

Note that for a given mixed strategy y of Player 2, the problem of finding a best response for Player 1 is that of choosing $x \in B(f)$ to maximize $x^T y$. This is the classical problem solved in Edmonds (1970) of maximizing a linear function over $B(f)$. With this observation, it follows that an optimal strategy for Player 1 can be computed in polynomial time using the ellipsoid algorithm (see e.g., Hellerstein et al. (2019)). In what follows, we give a *strongly* polynomial time algorithm.

Any given mixed strategy y of Player 2 can be expressed uniquely as a convex combination of his pure strategies (that is, vertices of $w_j e^j$ of C) simply by taking $\theta_i = y_j / w_j$. A given mixed strategy x of Player 1 can be written as a convex combination of at most n of her pure strategies x^g, by Carath´eodory's Theorem. In general, as discussed in Hoeksma et al. (2014), such a representation can be found in strongly polynomial time by combining the generic approach of Gr¨otschel et al. (2012) with the algorithm of Fonlupt and Skoda (2009)

for finding the intersection of a line with a polymatroid. The runtime of this algorithm is $O(n^9)$. For particular problems it is possible to exploit the structure of $B(f)$ in order to find a more efficient algorithm for representing a Player 1 mixed strategy as a convex combination of at most n of her pure strategies.

For a subset $S \subseteq V$, $S \neq \emptyset$, denote $\sum_{i \in S} 1/w_i$ by $w^{-1}(S)$. Consider the Player 2 mixed strategy

$$y^S = \sum_{i \in S} \frac{w_i^{-1}}{w^{-1}(S)} w_i e^i = \frac{1}{w^{-1}(S)} \sum_{i \in S} e^i.$$

For a Player 1 strategy $x \in B(f)$, the expected payoff against y^S is

$$P(x, y^S) = \sum_{i \in S} x_i \frac{1}{w^{-1}(S)} = \frac{x(S)}{w^{-1}(S)} \leq \frac{f(S)}{w^{-1}(S)},$$

by definition of $B(f)$. We summarize this in the following lemma.

Lemma 2 If Player 2 uses the strategy y^S for some $S \neq \emptyset$, the expected payoff is at most $f(S)/w^{-1}(S)$.

We will show in Theorem 4 that the strategy y^S is optimal for Player 2, where S is chosen to minimize $f(S)/w^{-1}(S)$. A minimizing set S can be found in strongly polynomial time, using a parametric search (see Iwata et al. (1997) [Section 6] for a parametric search algorithm for minimizing the ratio of a submodular function to a non-negative supermodular function). This relies on an algorithm for minimizing a submodular function. The fastest known strongly polynomial algorithm for submodular function minimization is that of Orlin (2009), whose runtime is $O(n^6)$, so that the minimization of $f(S)/w^{-1}(S)$ takes time $O(n^7)$.

Before stating and proving the theorem, we define a strategy which will be optimal for Player 1. To do this, we recursively define a partition of V into subsets $S_1, \cdots, S_r$.

Definition 3 (f-w **decomposition**) Set $S_0 = \emptyset$ and suppose $S_1, \cdots, S_j$ have already been defined for some $j \geq 0$. Then if $S^j \equiv S_1 \cup \cdots \cup S_j$ is equal to V, set $r = j$. If not, we define S_{j+1} to be any set $S \subseteq V \setminus S^j$ that minimizes $h_{S^j}(S)$, where

$$h_T(S) \equiv \frac{f(T \cup S) - f(T)}{w^{-1}(S)}.$$

We call $S \equiv (S_1, \cdots, S_r)$ an f-w decomposition of V.

Note that the function h_T is the ratio of a submodular function and a modular function, therefore, as remarked earlier, it can be minimized in strongly polynomial timeSince h_T is defined in terms of f and w, a more informative notation is $h_T^{f,w}$, but we omit the superscripts in general when they are clear from the context.

We now define the Player 1 strategy x^S by

$$x_i^S = w_i^{-1} h_{S^{j-1}}(S_j) \text{ for all } i \in S_j, \, j = 1, \ldots, r.$$

To show that x^S it is indeed a strategy, we need to prove that it lies in $B(f)$. Let $T \subseteq V$ be arbitrary and let $T_j = T \cap S_j$ for $j = 0, 1, \ldots, r$. Also set $T^j = \cup_{i \leq j} T_i$. Then

$$x^S(T) = \sum_{j=1}^{r} \sum_{i \in T_j} w_i^{-1} h_{S^{j-1}}(S_j) = \sum_{j=1}^{r} w^{-1}(T_j) h_{S^{j-1}}(S_j) \leq \sum_{j=1}^{r} f(S^{j-1} \cup T_j) - f(S^{j-1}),$$

by definition of S_j. Since f is submodular, $f(S^{j-1} \cup T_j) + f(T^{j-1}) \leq f(S^{j-1}) + f(T^j)$, so

$$x^S(T) \leq \sum_{j=1}^{r} f(T^j) - f(T^{j-1}) = f(T).$$

Hence, $x^S \in P(f)$. It is also easy to see that $x^S(V) = f(V)$, so that $x^S \in B(f)$.

Theorem 4 Suppose S^* is a non-empty set that minimizes $f(S)/w^{-1}(S)$. Then the value of the game $\Gamma_{\max}(f, w)$ is equal to $f(S^*)/w^{-1}(S^*)$. An optimal strategy for Player 2 is y^{S^*}. An optimal strategy for Player 1 is x^S, where $S = (S_1, \ldots, S_r)$ is any f-w decomposition.

Proof. By Lemma 2, the value of the game is at most $f(S^*)/w^{-1}(S^*)$. To complete the proof, we will show that x^S ensures a payoff at least $f(S^*)/w^{-1}(S^*) = h_\varnothing(S_1)$ against any Player 2 strategy. Note that for a pure strategy i of Player 2 with $i \in S_j$, the expected payoff against x^S is

$$P(x^S, i) = w_i x_i^S = h_{S^{j-1}}(S_j).$$

So it is sufficient to show that $h_{S^{j-1}}(S_j)$ is non-decreasing in j. By definition of S_j, we have

$$\frac{f(S^j) - f(S^{j-1})}{w^{-1}(S_j)} \leq \frac{f(S^{j+1}) - f(S^{j-1})}{w^{-1}(S_j \cup S_{j+1})}, \tag{2}$$

for $j = 1, \ldots, r - 1$. Writing $w^{-1}(S_j \cup S_{j+1}) = w^{-1}(S_j) + w^{-1}(S_{j+1})$ and rearranging yields

$$w^{-1}(S_j)(f(S^{j+1}) - f(S^j)) \geq w^{-1}(S_{j+1})(f(S^j) - f(S^{j-1})).$$

8

This is equivalent to $h_{S_i}(S_{j+1}) \geq h_{S_{j-1}}(S_j)$, and the proof is complete.

In general, both players have multiple optimal strategies. For Player 2, we can characterize these strategies.

Let $F = F(f)$ be the family of sets $S \neq \varnothing$ that minimize $f(S)/\bar{w}^1(S)$, so that the value v of the game is equal to $f(S)/\bar{w}^{-1}(S)$ for any $S \in F$. We also set $f(\varnothing)/\bar{w}^1(\varnothing)$ to be equal to v, so that $\varnothing \in F$. It is useful to note that F is a lattice. Indeed, suppose $S, T \in F$. In the following calculation, we use the observation that for any $a, b, c, d > 0$ if $a/b, c/d \geq v$ then $(a + c)/(b + d) \geq v$, where the second inequality is tight if the first is also tight. We have

$$v = \frac{f(S) + f(T)}{w^{-1}(S) + w^{-1}(T)} \geq \frac{f(S \cup T) + f(S \cap T)}{w^{-1}(S \cup T) + w^{-1}(S \cap T)} \geq v,$$

where the equality and second inequality follow from our observation and the first inequality follows from the submodularity of f. Therefore, the two inequalities hold with equality, and $S \cup T, S \cap T \in F$.

Theorem 5 A Player 2 strategy y is optimal if and only if it is in the convex hull of $\{y^S : S \in F(f)\}$.

Proof. By Theorem 4, each element of $\{y^S : S \in F\}$ is optimal, so any convex combination of such points is also optimal.

For the opposite direction, suppose that y^* is an optimal Player 2 strategy. By relabeling, let us assume that $y_1^* \geq \cdots \geq y_n^*$. Then recalling that $y^S = (\sum_{i \in S} e^i)/w^{-1}(S)$ for $S \subseteq V$ and setting $y_{n+1}^* = 0$, we can write y^* as

$$y^* = \sum_{i=1}^{n} y_i^* e^i = \sum_{i=1}^{n} e^i \sum_{j=i}^{n} (y_j^* - y_{j+1}^*) = \sum_{j=1}^{n} (y_j^* - y_{j+1}^*) \sum_{i=1}^{j} e^i = \sum_{j=1}^{n} \lambda_j y^{[j]},$$

where $\lambda_j = (y_j^* - y_{j+1}^*) w^{-1}([j])$. Note that

$$\sum_{j=1}^{n} \lambda_j = \sum_{j=1}^{n} (y_j^* - y_{j+1}^*) \sum_{i=1}^{j} w_i^{-1} = \sum_{i=1}^{n} w_i^{-1} \sum_{j=i}^{n} (y_j^* - y_{j+1}^*) = \sum_{j=1}^{n} y_j^*/w_j = 1,$$

where the final equality follows from the fact that $y^* \in C$. So y^* is a convex combination of the strategies $y^{[j]}$. We claim that if $\lambda_k > 0$ for some k then $[k] \in F$, so that y^* is in fact a convex combination of strategies y^S with $S \in F$. Indeed, suppose that $\lambda_k > 0$, so that $y_k^* > y_{k+1}^*$. Since any pure strategy best response x to y^* maximizes $x^T y^*$, by Lemma 1, we

can express x as a point x^σ such that the first k terms of σ are $[k]$ in some order. So by definition of x^σ,

$$\sum_{i=1}^{k} x_i = f([k]). \tag{3}$$

Equation (3) also holds for any mixed strategy x which is a best response to y^* (since x must be a mixture of pure best responses to y^*). In particular, it holds for $x = x^S$, where S is any f-w decomposition of V whose first element S_1 is the maximal element $\cup_{S \in F} S$ of F.

We claim that $[k] \subseteq S_1$. Let $i \in [k]$ and suppose $i \in S_j$ for some $j > 1$. Since $y_i^* \geq y_k^* > y_{k+1}^* \geq 0$ and any Player 2 pure strategy in the support of y^* that is played with positive probability must be a best response to x^S, it follows that strategy i is a best response to x^S. But by the maximality of S_1, inequality (2) with $j = 1$ is strict, and rearranging gives $h_\varnothing(S_1) < h_{S^1}(S_2)$. Since $h_{\varnothing(S^{j-1}}(S_j))$ is non-decreasing, for any $i^0 \in S_1$,

$$P(x^S, i^0) = h_\varnothing(S_1) < h_{S^{j-1}}(S_j) = P(x^S, i),$$

so i cannot be a best response to x^S, a contradiction. Hence, $i \in S_1$ so $[k] \subseteq S_1$.

Now, by definition of x^S,

$$\sum_{i=1}^{k} x_i^S = \sum_{i=1}^{k} \frac{w_i^{-1} f(S_1)}{w^{-1}(S_1)} = w^{-1}([k])v,$$

where v is the value of the game. Combining this with (3) yields $f([k])/w^{-1}([k]) = v$, so $[k] \in F$. This completes the proof. □

2.3 Special Cases

To find optimal strategies in the game $\Gamma_{\max}(f, w)$, it is necessary to minimize the function $h_T(S) = (f(T \cup S) - f(T))/w^{-1}(S)$. As previously remarked, there is a strongly polynomial time algorithm for this problem with runtime $O(n^7)$. To calculate an optimal Player 1 strategy, this algorithm must be run at most n times, so the overall runtime is $O(n^8)$. For some functions f, this minimization can be performed much faster, as we show in the remainder of this section.

Definition 6 We say that the payoff $P = P_{f,w}$ is ζ-*decreasing* if there exists $\zeta \in \mathbb{R}_+^n$ such

10

that for any $\sigma \in \Sigma(V)$ and any $i, j \in V$ with $\sigma^{-1}(i) < \sigma^{-1}(j)$,

$$\frac{P(\sigma, i)}{P(\sigma, j)} \geq \frac{\zeta_i}{\zeta_j}. \tag{4}$$

If $\frac{P(\sigma,i)}{P(\sigma,j)} \leq \frac{\zeta_i}{\zeta_j}$ we say P is ζ-increasing. If $\zeta_i = 1$ for all i, then we say P is *decreasing* (or respectively *increasing*).

If the payoff is ζ-decreasing (or increasing) we assume that the values ζ are given as part of the input of the problem.

Lemma 7 Suppose $P = P_{f,w}$ is ζ-decreasing. Then $S^* \equiv \cup_{S \in F} S$ is equal to $\{i \in V : \zeta_i \leq r\}$ for some $r > 0$.

Proof. It is sufficient to show that if P is ζ-decreasing and $i \in S^*$ and $j \notin S^*$, then $\zeta_i < \zeta_j$. Let x^S be any optimal Player 1 strategy such that the first set in the partition S is S^*, and write $x^S = \sum_{\sigma \in \Sigma} \theta_\sigma x^\sigma$ as a convex combination of pure strategies. Since $y_i^{S^*} > 0 = y_j^{S^*}$, for any best response σ to y^{S^*}, we can write $x^\sigma = x^{\tilde{\sigma}}$, where $\sigma^{-1}(i) < \tilde{\sigma}^{-1}(j)$, by Lemma 1. Since every pure strategy in the support of x^S must be a best response to y^{S^*}, we can assume that if $\theta_\sigma > 0$ then $\sigma^{-1}(i) < \sigma^{-1}(j)$. It follows from (4) that if $i \in S^*$ and $j \notin S^*$, then

$$\frac{P(x^S, i)}{\zeta_i} = \sum_{\sigma \in \Sigma} \frac{\theta_\sigma P(x^\sigma, i)}{\zeta_i} \geq \sum_{\sigma \in \Sigma} \frac{\theta_\sigma P(x^\sigma, j)}{\zeta_j} = \frac{P(x^S, j)}{\zeta_j}. \tag{5}$$

By Theorem 5, every element of S^* (in particular, i) is in the support of some optimal Player 2 strategy and j cannot be in the support of any Player 2 strategy. Therefore, i must be a best response to x^S and j cannot be a best response, so that

$$P(x^S, i) < P(x^S, j). \tag{6}$$

Combining (5) and (6) yields $\zeta_i < \zeta_j$.   2

It is worth pointing out that although the definition of ζ-decreasing and the proof of Lemma 7 are given in game theoretic terms, the lemma is not exactly a game theoretic result, and could be stated without reference to the game $\Gamma(f, w)$. Indeed, it is easy to see that $P_{f,w}$ is ζ-decreasing if and only if there exists $\zeta^0 \in R_+^n$ such that

$$\frac{f(S \cup \{i\}) - f(S)}{f(T \cup \{j\}) - f(T)} \geq \frac{\zeta_i^0}{\zeta_j^0}, \tag{7}$$

11

for any $S \subset T$ with $i \notin S, j \notin T$.

Lemma 7 implies that for games $\Gamma_{\max}(f, w)$ with a ζ-decreasing payoff function, the set $S^* = \cup_{S \in F} S$ can be found in time $O(n \log n)$, simply by relabeling the the elements of V so that they are in non-decreasing order of the index ζ_i, computing $f([k])/w^{-1}([k])$ for each $k \in [n]$ and choosing the largest k that minimizes this function. (Note that these n computations can done in time $O(n)$ by keeping a record of $\bar{w}^1([k])$ each time and adding w_{k+1}^{-1} to obtain $w^{-1}([k+1])$.) Therefore the value of the game $f(S)/w^{-1}(S^*)$ and the optimal Player 2 strategy y^{S^*} can be found in time $O(n \log n)$.

In order to compute the optimal Player 1 strategy x^S it is necessary to calculate an f-w decomposition S, which involves at most n minimizations of functions of the form $_T f(S)$. It is easy to check that if $P(f, w)$ is ζ-decreasing, then so is the function $P(f, w)$, where

$$f_T(S) = f(T \cup S) - f(S).$$

It follows that an f-w decomposition can be found in time $O(n^2)$. (However, expressing x^S as a convex combination of at most n pure strategies takes additional computation in general.)

We conclude this section by showing that when the payoff is decreasing, the solution of the game is particularly simple.

Lemma 8 If $P = P_{f,w}$ is decreasing then $f(S)/\bar{w}^1(S)$ is non-increasing in S and the value of the game is $f(V)/\bar{w}^1(V)$. The strategy x^S is optimal for Player 1, where S consists only of the set V, and y^y is optimal for Player 2.

Proof. Let $S \neq \varnothing$ be a proper subset of V, and without loss of generality, assume that $S = \{1, \ldots, k\}$ for some k. Let $j \notin S$ and let σ be any permutation of V that starts with $(1, 2, \ldots, k, j)$. Since P is decreasing, for any $i \in S$,

$$w_j(f(S \cup \{j\}) - f(S)) = P(\sigma, j) \leq P(\sigma, i) = w_i(f([i]) - f([i-1])).$$

Then setting $\theta = w_i^{-1}/w^{-1}(S)$, we obtain

$$\frac{f(S)}{w^{-1}(S)} = \sum_{i=1}^{k} \theta_i w_i (f([i]) - f([i-1]))$$

$$\geq \sum_{i=1}^{k} \theta_i w_j (f(S \cup \{j\}) - f(S))$$

$$= w_j (f(S \cup \{j\}) - f(S)).$$

Rearranging yields

$$f(S \cup \{j\}) w^{-1}(S) \leq f(S) w^{-1}(S \cup \{j\}),$$

or equivalently,

$$\frac{f(S \cup \{j\})}{w^{-1}(S \cup \{j\})} \leq \frac{f(S)}{w^{-1}(S)}.$$

This proves that $f(S)/w^{-1}(S)$ is non-increasing in S, so the value of the game is $\min_S f(S)/w^{-1}(S) = f(V)/w^{-1}(V)$.

The optimality of the stated strategies is immediate from Theorem 4.   *2*

3   Other Variations of the Game

Now let g be an arbitrary non-decreasing, *supermodular* function with $g(\varnothing) = 0$. The contrapolymatroid $Q(g)$ associated with g is defined by

$$Q(g) \equiv \{x \in \mathbf{R}^n : x(S) \geq g(S) \text{ for all } S \subseteq V\}.$$

The base of $Q(g)$ is given by

$$B(g) = \{x \in \mathbf{R}^n : x(S) \geq g(S) \text{ for all } S \subseteq V \text{ and } x(V) = g(V)\}. \tag{8}$$

Consider the game $\Gamma_{max}(g, w)$ where a mixed strategy for Player 1 (the maximizer) is some $x \in B(g)$, a mixed strategy for Player 2 (the minimizer) is some $y \in C$ and the payoff is $P_{g,w}(x, y) \equiv x^T y$. Let $g^\#$ be the *dual* of g, given by $g^\#(S) = g(V) - g(V \setminus S)$. It is easy to show that $g^\#$ is submodular and non-decreasing with $g^\#(\varnothing) = 0$ and $B(g) = B(g^\#)$. Moreover, $P_{g,w}$ is ζ-increasing if and only if $P_{g^\#,w}$ is ζ-decreasing. Therefore, the game $\Gamma_{max}(g, w)$ is equivalent to $\Gamma_{max}(g^\#, w)$, and the solution follows immediately from Theorems 4 and 5. Versions of Lemmas 7 and 8 also hold.

So far, we have considered two equivalent zero-sum games where Player 1 is the maximizer and Player 2 is the minimizer. We now consider an alternative game with Player 1 as minimizer and Player 2 as maximizer. Let g be a non-decreasing, supermodular function with $g(\varnothing) = 0$, and let $\Gamma_{min}(g, w)$ be the game which is the same as $\Gamma_{max}(g, w)$ except that Player 1 is the minimizer and Player 2 is the maximizer. Similarly, for a non-decreasing, submodular function f with $f(\varnothing) = 0$, we define $\Gamma_{min}(f, w)$ to be the same as $\Gamma_{max}(f, w)$ except that Player 1 is the minimizer and Player 2 is the maximizer.

Although the games $\Gamma_{min}(g, w)$ and $\Gamma_{min}(f, w)$ do not seem to be equivalent to the games of the previous section, the solutions and analysis are almost identical. We briefly describe the solutions here and leave the proofs as an exercise.

Analogously to an f-w decomposition, we define a g-w max-decomposition $S_1, \cdots, (S_r , S)$ as follows. Set $S_0 = \varnothing$ and suppose $S_0, \cdots, S_j$ have already been defined for some $j \geq 0$. Then if $S^j \equiv S_1 \cup \cdots \cup S_j$ is equal to V, set $r = j$. If not, we define S_{j+1} to be any set $S \subseteq V \setminus S^j$ that *maximizes* $h_{S^j}^{g,w}(S)$. This time, the function $h_T^{g,w}$ is the ratio of a super-modular function and a modular function and can be maximized by using the procedure of Iwata et al. (1997) to minimize the inverse ratio. Then the Player 1 strategy x^S is defined in precisely the same way as in the original version of the game.

Theorem 9 Let f be a non-decreasing submodular function with $f(\varnothing) = 0$ and let g be a non-decreasing supermodular function with $g(\varnothing) = 0$. Then the solutions to the games $\Gamma_{max}(f, w)$, $\Gamma_{max}(g, w)$, $\Gamma_{min}(f, w)$ and $\Gamma_{min}(g, w)$ are given in Table 1. The value and an optimal Player 1 strategy are indicated in the second and third columns of the table. In each case, the set of optimal Player 2 strategies is the convex hull of the set of y^{S^*} where S^* ranges over all possible values as given in the second column of the table. The fourth column gives a condition on the payoff for the set S^* to have the form given in the fifth column. The sixth column gives a condition for S^* to be equal to V.

4   Applications to Search Games

In this section we apply our results to a number of search games between a Searcher (Player 1) and a Hider (Player 2), where V corresponds to a set of hiding locations. In each example, a Searcher pure strategy is a permutation σ of V, where $\sigma(i)$ is the location that is in position i

Table 1: Solutions to four version of the game with submodular f and supermodular g

Game	Value	S for optimal Player 1 strategy $\hat{x}^S$	Condition on payoff	S^*, if condition on payoff holds	Condition on payoff for $S^* = V$
$\Gamma_{max}(f, w)$	$\frac{f(S^*)}{w^{-1}(S^*)} = \min_{S \subseteq V} \frac{f(S)}{w^{-1}(S)}$	f-w decomposition	ζ-decreasing	$\{i : \zeta_i \leq r\}$	decreasing
$\Gamma_{max}(g, w)$	$\frac{g^{\#}(S^*)}{w^{-1}(S^*)} = \min_{S \subseteq V} \frac{g^{\#}(S)}{w^{-1}(S)}$	$g^{\#}$-w decomposition	ζ-increasing	$\{i : \zeta_i \leq r\}$	increasing
$\Gamma_{min}(g, w)$	$\frac{g(S^*)}{w^{-1}(S^*)} = \max_{S \subseteq V} \frac{g(S)}{w^{-1}(S)}$	g-w max-decomposition	ζ-increasing	$\{i : \zeta_i \geq r\}$	increasing
$\Gamma_{min}(f, w)$	$\frac{f^{\#}(S^*)}{w^{-1}(S^*)} = \max_{S \subseteq V} \frac{f^{\#}(S)}{w^{-1}(S)}$	$f^{\#}$-w max-decomposition	ζ-decreasing	$\{i : \zeta_i \geq r\}$	decreasing

in the order of search and a Hider pure strategy is a location $i \in V$ at which a target is hidden.

4.1 A Weighted Search Game

Consider a game where the time to search location i is given by $t_i > 0$ and each location i has a weight d_i, corresponding to the rate of damage incurred at location i while the target has not been found. The payoff is given by $P(\sigma, i) = d_i C_i^{\sigma}$, for a permutation σ and $i \in V$, where

$$C_i^{\sigma} = \sum_{\sigma^{-1}(j) \leq \sigma^{-1}(i)} t_j.$$

This payoff is the total time to find the Hider multiplied by the rate of damage. The Searcher is the minimizer and the Hider is the maximizer. This game was considered by Yolmeh and Baykal-Gürsoy (2021), who solved the special case when the search times t_i are all equal to 1, using a polyhedral approach (Yolmeh and Baykal-Gürsoy (2021) also applied a column and row generation approach to the game in a more general network setting, with multiple searchers and targets.)

Condon et al. (2009) studied the special case of this game for $d_i = 1/c_i$, which they called the *game theoretic multiplicative regret* game. This case was also studied by Angelopoulos

et al. (2019). Implicit in the results of Condon et al. (2009) is an optimal Player 1 (Searcher) strategy and the value of the game for the general weighted search game with arbitrary d.

Here we give an alternative solution using the framework presented in Sections 2 and 3. The searching of locations is analogous to the processing of jobs in single machine scheduling, and in the language of scheduling theory, we can interpret the time t_i as the *processing time* of job i and the time C_i^σ as the *completion time* of job i under the schedule σ. We associate a Searcher pure strategy σ with a point x given by $x_i^\sigma = t_i C_i^\sigma$, $i \in V$. It is well known from scheduling theory (see Queyranne and Schulz (1994)) that the set of feasible vectors x^σ are the vertices of $B(g)$, where g is the supermodular function given by

$$g(S) = \frac{1}{2}(t(S)^2 + t^2(S)),$$

and $t^2(S) = \sum_{i \in S} t_i^2$. The polyhedron $B(g)$ is known as the *scheduling polyhedron* and corresponds to the set of Searcher mixed strategies in the search game. Let $w_i = d_i/t_i$. Then for a Hider pure strategy i, the expected payoff against a Searcher mixed strategy given by x is $x_i w_i$. Hence, this is the game $\Gamma_{\min}(g, w)$ and its solution follows from Theorem 9. The value of the game is

$$\max_{S \subseteq V} \frac{g(S)}{w^{-1}(S)} = \max_{S \subseteq V} \frac{(t(S)^2 + t^2(S))/2}{\sum_{i \in S} t_i/d_i}.$$

It is easy to see that the payoff $P(\sigma, i)$ is ζ-increasing where $\zeta = d$. Hence, by Theorem 9, the value and optimal strategies can be found in time $O(n \log n)$. To express the optimal Searcher strategy as a mixture of at most n pure strategies, one can use the strongly polynomial time decomposition algorithm of Hoeksma et al. (2014).

We note that two different solutions of the special case when the rates of damage d are all equal to 1 were given by Lidbetter (2013) and Alpern and Lidbetter (2013), though in each solution the size of the support of the optimal Searcher strategy was exponential in n. Condon et al. (2009) also considered this special case, calling it the *game theoretic total cost* problem. They found an optimal Searcher strategy of support size n. Theorem 9 implies an alternative polynomial time algorithm for finding an optimal Searcher strategy with support size n. Furthermore, the payoff is increasing in this case, so Theorem 9 implies that the optimal Hider strategy given by Lidbetter (2013) and Alpern and Lidbetter (2013) is unique.

4.2 A Weighted Search Game with Variable Speeds

We can extend the model of the previous subsection by adopting the variable speed network model, as considering by Alpern and Lidbetter (2014). Suppose that we think of the set of locations V as endpoints of n arcs, whose other endpoint is a common point O. The Searcher successively travels from O to the end of each arc and back again, where the time to travel from O to the end of arc i is $a > 0$ and the time to travel back again is $b > 0$. Let $t_i = a_i + b_i$ be the tour time of arc i. Similarly to the previous subsection, the vector $\check{C}^\sigma$ is defined as

$$\check{C}_i^\sigma = a_i + \sum_{\sigma^{-1}(j) < \sigma^{-1}(i)} t_j = C_i^\sigma - b_i,$$

and corresponds to the times the Searcher reaches each location under σ.

We consider a weighted search game with a minimizing Searcher and a maximizing Hider, whose payoff for a permutation σ and $i \in V$ is given by $\check{C}_i^\sigma$. If $b_i = 0$ for all i, then $\check{C}^\sigma = C^\sigma$ and this is equivalent to the model of the previous subsection.

The special case when the rates of damage d are all equal to 1 was solved by Alpern and Lidbetter (2014) in the more general setting of tree networks, but the optimal Searcher strategy given had exponential support size even in the case of no network structure. The case of arbitrary d has not been considered before.

Let $x_i^\sigma = t_i \check{C}_i^\sigma$ and let $w_i = d_i/t_i$, so that the payoff for a Searcher strategy σ and a Hider strategy i is $w_i x_i$. Note that we can write $x^\sigma = x^\sigma - c$, where x^σ is defined as in the previous subsection and c is given by $c = b_i t_i$. Therefore, the convex hull of the vectors x^σ is equal to $B(g) - c = B(\tilde{g})$, where $\tilde{g}$ is the non-decreasing supermodular function given by

$$\tilde{g}(S) = g(S) - c(S) = \frac{1}{2}(t(S)^2 + \sum_{j \in S} (a_j - b_j)t_j).$$

Therefore, this is the game $\Gamma_{\min}(\tilde{g}, w)$, and its solution follows from Theorem 9. The value of the game is

$$\max_{S \subseteq V} \frac{g(S)}{w^{-1}(S)} = \max_{S \subseteq V} \frac{(t(S)^2 + \sum_{j \in S} (a_j - b_j)t_j)/2}{\sum_{i \in S} t_i/d_i}.$$

Again, the payoff function here is ζ-increasing for $\zeta = d$, so the value and optimal strategies can be found in time $O(n \log n)$. Also, since $B(\tilde{g})$ is simply a translation of $B(g)$ by $-c$, we can again use the decomposition theorem of Hoeksma et al. (2014) for $B(g)$ to write

an optimal mixed Searcher strategy $x \in B(\tilde{g})$ as a convex combination of at most n pure strategies.

Similarly to the previous subsection, for the special case considered in Alpern and Lidbetter (2014) where $d = 1$ for all i, our solution here improves upon the optimal Searcher strategy of exponential support size. Also, since the payoff is increasing, Theorem 9 implies that the optimal Hider strategy is unique.

4.3 Application to a Search and Rescue Game

We now introduce a new search game in which we independently associate to every $i \in V$ a probability p_i that the Searcher does not get captured when searching location i and a probability q_i that a target located at i is found if location i is searched. The payoff of the game is the probability the Searcher finds the target without getting captured herself.

More precisely, for a given permutation σ and a given $i \in V$, the payoff is $q_i \pi_i^\sigma$, where

$$\pi_i^\sigma = \prod_{\sigma^{-1}(j) \leq \sigma^{-1}(i)} p_j.$$

The Searcher is the maximizer and the Hider is the minimizer. Let $x_i^\sigma = \frac{1-p_i}{p_i} \pi_i^\sigma$. It was shown by Kodialam (2001) and independently by Agnetis et al. (2009) that the set of feasible vectors x^σ are the vertices of $B(f)$ where f is the non-decreasing submodular function given by

$$f(S) = 1 - \prod_{i \in S} p_i. \tag{9}$$

Setting w_i to be equal to $q_i p_i / (1 - p_i)$, we see that this is the game $\Gamma_{\max}(f, w)$. Therefore, by Theorem 4, the value of the game is

$$\min_{S \subseteq V} \frac{f(S)}{w^{-1}(S)} = \min_{S \subseteq V} \frac{1 - \prod_{i \in S} p_i}{\sum_{i \in S} (1 - p_i)/(q_i p_i)}.$$

The payoff is easily seen to be ζ-decreasing where $\zeta_i = q_i$ (or indeed where $\zeta_i = q_i/p_i$). It follows from Theorem 9 that the value and optimal strategies can be found in time $O(n \log n)$.

Kodialam (2001) gave a strongly polynomial algorithm with runtime $O(n^3 \log n)$ for representing a point in $B(f)$ as a convex combination of at most n vertices, and we can use this to express the optimal Searcher strategy as a mixture of at most n pure strategies.

In the case that $q_i = 1$ for all i, the game reduces to the one considered in Lidbetter (2020), where a solution was given but the size of the support of the optimal Searcher strategy was exponential in n. This approach gives an optimal strategy with support size n. Since the payoff is decreasing in this case, the optimal Hider strategy given in Lidbetter (2020) is unique.

5   Relation to Sequential Testing

In this section we show that a sequential testing problem studied in Condon et al. (2009) and Kodialam (2001) is equivalent to the "minimization" version of the game considered in Subsection 4.3.

Suppose some items, or *tuples* must be routed in some order through a set V of operators, each of which tests whether the tuple satisfies some predicate (or filter) of a conjunction. To spread the load on the operators, different tuples may be routed in different orders. There is a known probability p_i that a tuple will fail the test of operator i, and the tuple is routed through the operators until it fails one of the tests (and is eliminated) or it passes all of them. The problem here is to maximize the rate of flow of tuples routed through the operators, subject to the constraint that operator i has a maximum flow rate of r_i. More precisely, the problem is given by the following linear program, where we denote that set of permutations of V by $\Sigma(V)$.

$$\max \sum_{\sigma \in \Sigma(V)} \lambda_\sigma \quad \text{s.t.} \quad \sum_{\sigma \in \Sigma(V)} \lambda_\sigma \prod_{\sigma^{-1}(j) < \sigma^{-1}(i)} p_j \leq r_i \text{ for all } i \in V,$$

$$\lambda_\sigma \geq 0 \text{ for all } \sigma \in \Sigma(V).$$

The variables λ_σ here can be interpreted as the rate that tuples are routed through the operators in the order given by the permutation σ. We adopt the terminology of Condon et al. (2009) and call this the *max-throughput* problem. The problem was solved in both Condon et al. (2009) and Kodialam (2001), the latter paper exploiting the polymatroid structure of a space associated with the problem and the former giving a more efficient combinatorial algorithm with no reference to polymatroids.

Let $q_i = 1/(p_i r_i)$ and recall the notation $\pi_i^\sigma = \prod_{\sigma^{-1}(j) \leq \sigma^{-1}(i)} p_j$ of the previous section. Let $v = 1/(\sum_{\sigma \in \Sigma(V)} \lambda_\sigma)$ and let $\theta_\sigma = v \lambda_\sigma$. Then the max-throughput problem is equivalent

to the following LP

$$\min v \text{ s.t.} \sum_{\sigma \in \Sigma(V)} \theta_\sigma q_i \pi_i^\sigma \le v \text{ for all } i \in V,$$

$$\sum_{\sigma \in \Sigma(V)} \theta_\sigma = 1,$$

$$\theta_\sigma \ge 0 \text{ for all } \sigma \in \Sigma(V).$$

This is the problem of finding an optimal strategy for Player 1 in the game $\Gamma_{\min}(f, w)$, where f is given by (9) and $w_i = q_i p_i/(1 - p_i) = r_i/(1 - p_i)$. Therefore, the solution follows from Theorem 9. The algorithm of Kodialam (2001) is essentially a special case of the algorithm given in the proof of Theorem 4.

The derivation of the equivalence of these two problems closely follows the derivation in Condon et al. (2009) of the equivalence of the game theoretic multiplicative regret problem and an artificial problem they called the *cumulative cost limit* problem.

6 Applications to Queueing Theory

As mentioned in the Introduction, the performance space of several multiclass queueing systems have been shown in Federgruen and Groenevelt (1988) and Shanthikumar and Yao (1992) to have a polymatroid structure. Possible performance measures of interest include the expected delay of the first m jobs, the expected number of type i jobs in the system at time t or the expected number of job completions by time t. Depending on the context, the objective may be to maximize or minimize the performance measure and many such problems can be regarded as a special case of maximizing or minimizing a linear function over the base of a polymatroid.

For every maximization or minimization problem of this type we can consider a max-min or min-max variant. If we have an oracle for the submodular or supermodular function that defines the polymatroid or contrapolymatroid associated with a problem (in particular, if the function can be expressed in closed form), then the solution of the max-min or min-max problem follows from the results of this paper. We discuss one such problem here as an example rather than giving an exhaustive list of problems.

Coffman Jr and Mitrani (1980) consider a queueing system with a single server with n classes V of jobs whose arrival times follow a Poisson process and whose service times are

exponentially distributed (that is, a *M/M/*1 system). Jobs in class i arrive at rate λ_i and are serviced at rate μ_i. The *traffic intensity* of jobs of class i is $\rho_i = \lambda_i/\mu_i$. It is assumed that $\rho(V) \equiv \sum_{i=1}^{n} \rho_i < 1$, which ensures the existence of a stationary distribution for the number of jobs in the system. The expected time that jobs of class i spend in the system in the steady state is denoted W_i and depends on the scheduling strategy chosen.

Let $x \in R^n$ be defined by $x_i = \rho_i W_i$. It is shown in Coffman Jr and Mitrani (1980) that the space of feasible vectors x is the base $B(g)$ of the contrapolymatroid given by the supermodular function

$$g(S) = \frac{\sum_{i \in S} \rho_i/\mu_i}{1 - \rho(S)}.$$

Each vertex x^σ of $B(g)$ corresponds to a priority rule that assigns jobs to the server based on some fixed priority ordering of the job classes (given by the permutation σ). A non-vertex point $x = \sum_{\sigma \in \Sigma(V)} \theta_\sigma x^\sigma \in B(g)$ can be interpreted as a randomized priority rule where in each busy period the priority rule σ is chosen with probability θ_σ.

A well known consequence is that if the objective is to minimize some weighted sum $\sum_{i=1}^{n} c_i W_i$ of expected number of jobs in the system (where c_i may correspond to the holding cost per unit time of jobs of class i), we can simply use the greedy algorithms of Edmonds (1970) to minimize $w^T x$ with $w_i = c_i/\rho_i$. The solution is a priority rule that corresponds to some vertex of $B(g)$.

Now suppose we wish to minimize the (weighted) *maximum* expected holding cost of any class of jobs. That is, we wish to find a performance vector $x \in B(g)$ that minimizes $\max_i w_i x_i$. A solution is given by Theorem 9 of this paper, and the value of the min-max expected holding cost is

$$\max_{S \subseteq V} \frac{g(S)}{w^{-1}(S)} = \max_{S \subseteq V} \frac{\sum_{i \in S} \rho_i/\mu_i}{(1 - \rho(S)) \sum_{i \in S} \rho_i/c_i}.$$

7 Finding Optimal Strategies when the Payoff is Monotone

As mentioned in Section 2, expressing an optimal Player 1 strategy x^S as a convex combination of pure strategies relies on an algorithm whose runtime is $O(n^9)$, in general. We have also seen that for particular polymatroids, this runtime can be reduced. In this section

we show that if f is submodular and $P_{f,w}$ is decreasing, then an optimal Player 1 strategy can be efficiently implemented. More particularly, we show that a random pure strategy can be drawn from the (exponentially-sized) support of this optimal strategy, with appropriate probability, in time $O(n)$.

Theorem 10 Suppose f is submodular and $P_{f,w}$ is decreasing. Then there is an optimal Player 1 strategy x for $\Gamma_{\max}(f, w)$ such that a random pure strategy x^σ for Player 1, drawn from the distribution on pure strategies defined by x, can be generated in $O(n)$ time. An analogous result holds for $\Gamma_{\min}(g, w)$ if g is supermodular and $P_{g,w}$ is increasing.

Proof. First, we introduce some notation. For $A \subseteq V$, let $f_A : 2^{V \setminus A} \to R_+$ be given by $f_A(S) = f(S \cup A) - f(A)$. We also write $f|_A$ for the function f restricted to subsets of A and $w|_A$ for the vector w restricted to elements in A. It is easy to show that f_A and $f|_A$ are submodular and the payoffs $P_{f_A,w|_A}$ and $P_{f_A,w|_A}$ are decreasing.

We begin by constructing an optimal strategy for $P_{f,w}$. We construct the strategy recursively. If $n = 1$, only one strategy is available, which is optimal. Suppose $n \geq 2$ and we have a construction for games such that the number of Player 2 strategies is $n - 1$ and let $V^0 = V \setminus \{n\}$. Define

  (i) $\Gamma_1 \equiv \Gamma_{\min}(f|_{V^0}, w|_{V^0})$,

  (ii) $\Gamma_2 \equiv \Gamma_{\min}(f_{\{n\}}, w|_{V^0})$,

whose values are $V_1 \equiv f(V^0)/w^{-1}(V^0)$ and $V_2 \equiv (f(V) - f(\{n\}))/w^{-1}(V^0)$, respectively, by Lemma 8. By induction, we have a construction for an optimal strategy for both of these games. Denote these optimal strategies x^1 and x^2, respectively. We now define two new strategies $\tilde{x}^1$ and $\tilde{x}^2$ for $\Gamma_{\min}(f, w)$ as follows. The strategy $\tilde{x}^1$ is obtained by replacing each pure strategy x^σ in x^1 with x^{σ^0}, where σ^0 is σ followed by element n. The strategy $\tilde{x}^2$ is obtained by replacing each pure strategy x^σ in x^2 with $x^{\sigma^{00}}$, where σ^{00} is σ preceded by element n.

Table 2 displays the payoff of the strategies $\tilde{x}^1$ and $\tilde{x}^2$ against the element n and against any element of V^0.

The function $f(S)/w^{-1}(S)$ is non-increasing, by Lemma 8. Hence,

$$w_n f(\{n\}) = \frac{f(\{n\})}{w_n^{-1}} \geq \frac{f(V)}{w^{-1}(V)} = \frac{f(V)}{w^{-1}(V^0) + w_n^{-1}}.$$

22

Table 2: Expected payoffs $P(x^1, i)$ and $P(x^2, i)$ for $i = n$ and $i \in V^0$.

	$i = n$	$i \in V^0$
x^1	$w_n(f(V) - f(V^0))$	V_1
x^2	$w_n f(\{n\})$	V_2

Rearranging, we obtain

$$w_n f(\{n\}) \geq \frac{f(V) - f(\{n\})}{w^{-1}(V^0)} = V_2.$$

Also,

$$\frac{f(V^0)}{w^{-1}(V^0)} \geq \frac{f(V)}{w^{-1}(V)} = \frac{f(V)}{1/w_n + w^{-1}(V^0)}.$$

Rearranging gives

$$w_n(f(V) - f(V^0)) \leq f(V^0)/w^{-1}(V^0) = V_1.$$

It follows that by mixing appropriately between strategies x^1 and x^2, Player 1 can construct a strategy x whose expected payoff against against any pure strategy (and therefore also any mixed strategy) of Player 2 is equal to some constant c. Therefore, by definition of the optimal Player 2 strategy, y,

$$c = P(x, y) = \frac{f(V)}{w^{-1}(V)},$$

so c is the value $f(v)/w^{-1}(V)$ of the game and x is optimal.

We note that, because each recursive call mixes between two strategies, the support of the final constructed strategy x has size 2.

We now describe how to generate a random pure strategy $\hat{x}$ from the distribution on pure strategies defined by x, without actually constructing x. The procedure is similar to the recursive construction above. However, in each recursive call, we do not recursively generate optimal strategies for both Γ_1 and Γ_2. Instead, we first generate the payoffs in Table 2 and calculate the mixing probabilities for x^1 and x^2, call them p^0 and $p^{00}(= 1 - p^0)$. We then randomly choose between recursively generating a pure strategy for Γ_1 or for Γ_2, choosing the first with probability p^0 and the second with probability p^{00}. Denote by x^σ the pure strategy that is generated. If it was generated for Γ_1, we return x^{σ^0}, where σ^0 is produced from σ by appending element n. If it was generated for Γ_2, we return $x^{\sigma^{00}}$ where σ^{00} is produced from σ by prepending element n. It is clear that this procedure generates a random pure strategy with the appropriate probability.

It remains to verify that this procedure can be implemented to run in time $O(n)$. Recall that we assume that each oracle query can be answered in constant time. The procedure makes $O(n)$ recursive calls. The only non-trivial part of the analysis is the computation of the mixing probabilities β and β^{00} in a recursive call. These are computed from the four entries in Table 2. The entries in the first column of the table can be computed in constant time. The entries in the second column, V_1 and V_2, are equal to $f(V^\beta)/w^{-1}(V^0)$ and $f(V)/w^{-1}(V^0)$ respectively. Computing these values from scratch in each recursive call would take linear time per recursive call. However, using the fact that $w^1(V) = 1/w_n + w^{-1}(V^0)$, we can easily reduce the computation in each recursive call to take constant time, by taking advantage of the computation done in the previous recursive call. Thus the runtime is $O(n)$.

An analogous result for $\Gamma_{\max}(g, w)$ can be proved similarly. $\quad\quad\quad\quad 2$

8 Conclusion

We have provided a unifying framework under which we can understand and analyze a number of natural games that arise in different research areas, gaining new insight into existing results and proving new results. There are many related problems in search theory and sequential testing that do not fall under this framework, including problems involving networks and multiple targets. A promising avenue for future research could be to explore polyhedral approaches to such problems.

Acknowledgements

References

Agnetis, A., Detti, P., Pranzo, M., and Sodhi, M. S. (2009). Sequencing unreliable jobs on parallel machines. *Journal of Scheduling*, 12(1):45.

Alpern, S. and Gal, S. (2003). *The theory of search games and rendezvous*, volume 55. Kluwer, Boston.

Alpern, S. and Lidbetter, T. (2013). Mining coal or finding terrorists: The expanding search paradigm. *Operations Research*, 61(2):265–279.

Alpern, S. and Lidbetter, T. (2014). Searching a variable speed network. *Mathematics of Operations Research*, 39(3):697–711.

Angelopoulos, S., Dürr, C., and Lidbetter, T. (2019). The expanding search ratio of a graph. *Discrete Applied Mathematics*, 260:51–65.

Bellman, R. E. (1957). *Dynamic programming.* Princeton University Press.

Coffman Jr, E. G. and Mitrani, I. (1980). A characterization of waiting time performance realizable by single-server queues. *Operations Research*, 28(3-part-ii):810–821.

Condon, A., Deshpande, A., Hellerstein, L., and Wu, N. (2009). Algorithms for distributional and adversarial pipelined filter ordering problems. *ACM Transactions on Algorithms (TALG)*, 5(2):1–34.

Edmonds, J. (1970). Submodular functions, matroids and certain polyhedra, combinatorial structures and their applications (R. Guy, H. Hanani, N. Sauer and J. Shonheim, eds.). *New York*, pages 69–87.

Federgruen, A. and Groenevelt, H. (1988). Characterization and optimization of achievable performance in general queueing systems. *Operations Research*, 36(5):733–741.

Fonlupt, J. and Skoda, A. (2009). Strongly polynomial algorithm for the intersection of a line with a polymatroid. In *Research Trends in Combinatorial Optimization*, pages 69–85. Springer.

Grötschel, M., Lov´asz, L., and Schrijver, A. (2012). *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media.

Hellerstein, L., Lidbetter, T., and Pirutinsky, D. (2019). Solving zero-sum games using best-response oracles with applications to search games. *Operations Research*, 67(3):731–743.

Hoeksma, R., Manthey, B., and Uetz, M. (2014). Decomposition algorithm for the single machine scheduling polytope. In *International Symposium on Combinatorial Optimization*, pages 280–291. Springer.

Hohzaki, R. (2016). Search games: literature and survey. *Journal of the Operations Research Society of Japan*, 59(1):1–34.

Iwata, S., Murota, K., and Shigeno, M. (1997). A fast parametric submodular intersection algorithm for strong map sequences. *Mathematics of Operations Research*, 22(4):803–813.

Kodialam, M. S. (2001). The throughput of sequential testing. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 280–292. Springer.

Lidbetter, T. (2013). Search games with multiple hidden objects. *SIAM Journal on Control and Optimization*, 51(4):3056–3074.

Lidbetter, T. (2020). Search and rescue in the face of uncertain threats. *European Journal of Operational Research*.

Orlin, J. B. (2009). A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251.

Queyranne, M. (1993). Structure of a simple scheduling polyhedron. *Mathematical Programming*, 58(1-3):263–285.

Queyranne, M. and Schulz, A. S. (1994). *Polyhedral approaches to machine scheduling*. TU, Fachbereich 3, Berlin.

Shanthikumar, J. G. and Yao, D. D. (1992). Multiclass queueing systems: Polymatroidal structure and optimal scheduling control. *Operations Research*, 40(3-supplement-2):S293–S299.

Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66.

Stadje, W. (1995). Selecting jobs for scheduling on a machine subject to failure. *Discrete Applied Mathematics*, 63(3):257–265.

Yolmeh, A. and Baykal-Gürsoy, M. (2021). Weighted network search games with multiple hidden objects and multiple search teams. *European Journal of Operational Research*, 289(4):338–349.