Randomized Strategies for Robust Combinatorial Optimization*

Yasushi Kawase
Tokyo Institute of Technology
RIKEN AIP Center
kawase.y.ab@m.titech.ac.jp

Hanna Sumita
Tokyo Metropolitan University
sumita@tmu.ac.jp

Abstract

In this paper, we study the following robust optimization problem. Given an independence system and candidate objective functions, we choose an independent set, and then an adversary chooses one objective function, knowing our choice. The goal is to find a randomized strategy (i.e., a probability distribution over the independent sets) that maximizes the expected objective value in the worst case. This problem is fundamental in wide areas such as artificial intelligence, machine learning, game theory and optimization. To solve the problem, we propose two types of schemes for designing approximation algorithms. One scheme is for the case when objective functions are linear. It first finds an approximately optimal aggregated strategy and then retrieves a desired solution with little loss of the objective value. The approximation ratio depends on a relaxation of an independence system polytope. As applications, we provide approximation algorithms for a knapsack constraint or a matroid intersection by developing appropriate relaxations and retrievals. The other scheme is based on the multiplicative weights update (MWU) method. The direct application of the MWU method does not yield a strict multiplicative approximation algorithm but yield one with an additional additive error term. A key technique to overcome the issue is to introduce a new concept called (η,γ)-reductions for objective functions with parameters η and γ. We show that our scheme outputs a nearly α-approximate solution if there exists an α-approximation algorithm for a subproblem defined by (η,γ)-reductions. This improves approximation ratios in previous results. Using our result, we provide approximation algorithms when the objective functions are submodular or correspond to the cardinality robustness for the knapsack problem.

1 Introduction

Optimization under uncertainty about the objective is a fundamental task in artificial intelligence and machine learning. For example, consider the problem of controlling pan-tilt-zoom cameras to protect against adversarial intrusions (Krause, Roper, and Golovin 2011). We need to choose where to point the cameras under some scenarios of intrusions. Thus, we aim to maximize the chance of detecting intrusions in the worst case. See also (Chen et al. 2017) for another example. Krause, Roper, and Golovin (2011) and Chen et al. (2017) formulated such problems as the following robust combinatorial optimization problem; given an independence system $(E,\mathcal{I})$ (where $\mathcal{I}\subseteq 2^E$) and set functions $f_1,\dots,f_n\colon 2^E\to\mathbb{R}_+$, the goal is to find a minimax randomized strategy p that maximizes the worst case performance, i.e., $\min_{k\in\{1,\dots,n\}}\sum_{X\in\mathcal{I}}p_X f_k(X)$. Throughout this paper, we denote $[n]=\{1,\dots,n\}$ for a positive integer n. An independence system is a set system generalizing families of knapsack solutions and matroids; we give the formal definition in the preliminaries. The above problem is regarded as the problem of computing the *game value* in a two-person zero-sum game where one player (Algorithm) selects a feasible solution and the other player (Adversary) selects an objective function.

The robust optimization problem has also widespread application in game theory and combinatorial optimization.

One is the problem of computing the *Stackelberg equilibrium* of the (zero-sum) *security games*. This game models the interaction between a system *defender* (Algorithm) and a malicious *attacker* (Adversary) to the system. The model and its game-theoretic solution have various applications in the real world (Tambe 2011).

Another application is the problem of maximizing the *cardinality robustness* for the maximum weight independent set problem (Hassin and Rubinstein 2002; Fujita, Kobayashi, and Makino 2013; Kakimura and Makino 2013; Matuschke, Skutella, and Soto 2018; Kobayashi and Takazawa 2016). The goal is to choose an independent set of size at most k with as large total weight as possible, but the cardinality bound k is not known in advance. We refer this problem to the *maximum cardinality robustness problem (MCRP)*. We can regard MCRP as the game where Algorithm chooses an independent set X and then Adversary chooses k knowing X. We will describe details of these applications in the preliminaries.

One most standard way to solve the robust optimization problem is to use the *linear programming* (LP). In fact, it is known that we can compute the exact game value in polynomial time with respect to the numbers of deterministic (pure) strategies for both players (see, e.g., Nisan et al.; Bowles (2007; 2009) for more details). However, in our setting, direct use of the LP formulation is not effective. The set of deterministic strategies for the algorithm has exponen-

*A full version appears in (Kawase and Sumita 2018).

tially large cardinality in general, and hence the number of the variables in the LP formulation is exponentially large.

For another way, we can use the multiplicative weights update (MWU) method to solve the problem. The MWU method is an algorithmic technique which maintains a distribution on a certain set of interest and updates it iteratively by multiplying the probability mass of elements by suitably chosen factors based on feedback obtained by running another algorithm on the distribution (Kale 2007). MWU is simple but so powerful that it is widely used in game theory, machine learning, computational geometry, optimization, and so on. Freund and Schapire (1999) apply the MWU method to calculate the approximate value of a two-person zero-sum game, and showed that if (i) the set of deterministic strategies for Adversary is polynomially sized, and (ii) Algorithm can compute a *best response*, then MWU gives a polynomial-time algorithm to compute the game value up to an additive error of ϵ for any fixed constant $\epsilon > 0$. Krause, Roper, and Golovin (2011) and Chen et al. (2017) extended this result for the case when the algorithm can compute only an approximately best response. They provided a polynomial-time algorithm that finds an α-approximation of the game value up to additive error of $\epsilon \cdot \max_{k\in[n],\ X\in\mathcal{I}} f_k(X)$ for any fixed constant $\epsilon > 0$. This result leads no theoretical guarantee in general because the maximum objective value can be arbitrarily large compared with the optimal value. Moreover, obtaining an $(\alpha - \epsilon')$-approximation solution for a fixed constant $\epsilon' > 0$ by their algorithms requires pseudo-polynomial time. Recently, Hellerstein, Lidbetter, and Pirutinsky (2018) provided an $(\alpha-\epsilon)$-approximation algorithm with the MWU method for the case when the minimizer has exponentially many strategies. However, in our problem, it is hard to obtain a similar result by applying their technique.

In this paper, to solve the robust optimization problem, we provide two general schemes based on LP and MWU. As consequences, we develop (approximation) algorithms that works when the objective functions and the constraint that defines $\mathcal{I}$ belong to well-known classes in combinatorial optimization, such as submodular functions, knapsack/matroid/μ-matroid intersection constraints.

Related work While there exist still few papers on randomized strategies of the robust optimization problems, algorithms to find a deterministic strategy have been intensively studied in various setting. See survey papers (Aissi, Bazgan, and Vanderpooten 2009; Kasperski and Zieliński 2016) and references therein for details. Adopting a randomized strategy provides us two merits: the randomization improves the worst case value dramatically, and the optimal randomized strategy can be found easier than the deterministic one. We will describe these later in the preliminaries.

Since Hassin and Rubinstein (2002) introduced the notion of the cardinality robustness, many papers have been investigating the value of the maximum cardinality robustness (Hassin and Rubinstein 2002; Fujita, Kobayashi, and Makino 2013; Kakimura and Makino 2013; Kobayashi and Takazawa 2016; Kakimura, Makino, and Seimi 2012). Kakimura, Makino, and Seimi (2012) proved that the deterministic version of MCRP is weakly NP-hard but admits an FPTAS. Matuschke, Skutella, and Soto (2018) introduced randomized strategies for the cardinality robustness, and they presented a randomized strategy with $(1/\ln 4)$-robustness for a certain class of independence system. However, they did not consider the computational aspect of the cardinality robustness.

When $n = 1$, the deterministic version of the robust optimization problem is exactly the classical optimization problem $\max_{X\in\mathcal{I}} f(X)$. For the monotone submodular function maximization problem, there exist $(1-1/e)$-approximation algorithms under a knapsack constraint (Sviridenko 2004) or a matroid constraint (Calinescu et al. 2007; Filmus and Ward 2012), and there exists a $1/(\mu+\epsilon)$-approximation algorithm under a μ-matroid intersection constraint for any fixed $\epsilon > 0$ (Lee, Sviridenko, and Vondrák 2010). For the unconstrained non-monotone submodular function maximization problem, there exists a $1/2$-approximation algorithm, and this is best possible (Feige, Mirrokni, and Vondrák 2011; Buchbinder et al. 2015). As for the case when the objective function f is linear, the knapsack problem admits an FPTAS (Kellerer, Mansini, and Speranza 2000).

Main results and technique

LP-based algorithm We provide a two-step scheme for the case when all the objective functions $f_1, \ldots, f_n$ are linear. The first step solves the LP that finds an aggregated strategy for the original problem, and the second step retrieves a randomized strategy. In the both steps, we make use of a separation problem for the polytope of the feasible region in the LP, which consists of the independence system polytope. We show that if we can solve the separation problem efficiently, then we can also solve the robust optimization problem efficiently (Theorem 3). Consequently, the robust optimization problem can be solve in polynomial-time when $\mathcal{I}$ comes from a matroid, a matroid intersection, or s–t paths. This is a standard application of techniques obtained by Grötschel, Lovász, and Schrijver (2012). However, the scheme is not directly available when $\mathcal{I}$ comes from a knapsack or a μ-matroid intersection ($\mu \geq 3$) because the corresponding separation problems are NP-hard. A key point to resolve the issue is to use a slight relaxation of the feasible region. We show that if we can efficiently solve the separation problem for the relaxed polytope, then we can know an approximate optimal value (Theorem 4). The most difficult point is the retrieval step, because the LP optimal solution may not belong to the original feasible region. Instead we compute a randomized strategy by slightly shrinking the aggregated strategy vector. We prove the approximation ratio of the randomized strategy (Theorem 4). By developing appropriate relaxations and retrievals, we show a PTAS and a $2/(e\mu)$-approximation algorithm for the knapsack constraint and the μ-matroid intersection constraint, respectively.

The merit of the LP-based algorithm compared with MWU is that the LP-based one is applicable to the case when the set of possible objective functions is given by a half-space representation of a polytope. In the original problem, objective functions are given by a vertex representation, i.e., a convex hull of linear functions $\mathrm{conv}\{f_1, \ldots, f_n\}$. Both a half-space and a vertex representation of a polytope have different utility, and hence it is important that the LP-based algorithm can deal with both.

MWU-based algorithm We improve the technique of (Krause, Roper, and Golovin 2011; Chen et al. 2017) to obtain a better approximation algorithm based on the MWU method. Their algorithm adopts the value of $f_k(X)$ ($k \in [n]$) for update, but this may lead the slow convergence when $f_k(X)$ is too large for some k. In fact, the direct application of the MWU method does not yield a strict multiplicative approximation algorithm but a multiplicative approximation algorithm with an additional *additive error* term. To overcome this drawback, we make the convergence rate per iteration faster by introducing a novel concept called (η, γ)-*reduction* of objective functions (Definition 1). We assume that the algorithm can find an α-best response in the game where objective functions are (η, γ)-reductions of original ones, for any η and some polynomially bounded $\gamma \leq 1$. We use the procedure as a subroutine. Then we show that by appropriately setting η, for any fixed constant $\epsilon > 0$, our scheme gives an $(\alpha - \epsilon)$-approximation solution in polynomial time with respect to n and $1/\epsilon$ (Theorem 8). For example, we give an $(\eta, 1/|E|)$-reduction for submodular functions through submodular minimization. We remark that the support size of the output may be equal to the number of iterations. Without loss of the objective value, we can find a sparse solution whose support size is at most n by using LP.

The merit of the MWU-based algorithm is applicability to a wide class of the robust optimization problem. We also demonstrate our scheme for various optimization problems. For any $\eta \geq 0$, we show that a linear function has an $(\eta, 1/|E|)$-reduction to a linear function, a monotone submodular function has an $(\eta, 1)$-reduction to a monotone submodular function, and a non-monotone submodular function has an $(\eta, 1/|E|)$-reduction to a submodular function. Therefore, we can construct subroutines owing to existing work. Consequently, for the linear case, we obtain an FPTAS subject to the knapsack constraint (Theorem 13) and a $1/(\mu - 1 + \epsilon)$-approximation algorithm subject to the μ-matroid intersection constraint (Theorem 11). For the monotone submodular case, there exist a $(1 - 1/e - \epsilon)$-approximation algorithm for the knapsack or matroid constraint (Theorem 9), and a $1/(\mu + \epsilon)$-approximation for the μ-matroid intersection constraint (Theorem 10). For the non-monotone submodular case without a constraint, we derive a $(1/2 - \epsilon)$-approximation algorithm (Theorem 12).

Moreover, by applying our MWU-based scheme, we demonstrate an FPTAS for MCRP where $\mathcal{I}$ is defined from the knapsack problem (Theorem 14). To construct the subroutine for computing an approximate best response, we give a gap-preserving reduction of the subproblem to $\max_{X \in \mathcal{I}} v_{\leq k}(X)$ for any k, which admits an FPTAS (Caprara et al. 2000). We also show that MCRP is NP-hard.

We remark that both schemes produce a randomized strategy, but the schemes themselves are deterministic. Our results are summarized in Table 1.

2 Preliminaries

Linear and submodular functions Throughout this paper, we consider set functions f with $f(\emptyset) = 0$. We say that a set function $f\colon 2^E \to \mathbb{R}$ is *submodular* if $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$ holds for all $X, Y \subseteq E$ (Fujishige 2005; Krause and Golovin 2014). In particular, a set function $f\colon 2^E \to \mathbb{R}$ is called *linear* (modular) if $f(X) + f(Y) = f(X \cup Y) + f(X \cap Y)$ holds for all $X, Y \subseteq E$. A linear function f is represented as $f(X) = \sum_{e \in X} w_e$ for some $(w_e)_{e \in E}$. A function f is said to be *monotone* if $f(X) \leq f(Y)$ for all $X \subseteq Y \subseteq E$. A linear function $f(X) = \sum_{e \in X} w_e$ is monotone if and only if $w_e \geq 0$ ($e \in E$).

Independence system Let E be a finite ground set. An *independence system* is a set system $(E, \mathcal{I})$ with the following properties: (I1) $\emptyset \in \mathcal{I}$, and (I2) $X \subseteq Y \in \mathcal{I}$ implies $X \in \mathcal{I}$. A set $I \subseteq \mathcal{I}$ is said to be *independent*, and an inclusion-wise maximal independent set is called a *base*. The class of independence systems is wide and it includes matroids, μ-matroid intersections, and families of knapsack solutions.

A *matroid* is an independence system $(E, \mathcal{I})$ satisfying that (I3) if $X, Y \in \mathcal{I}$ and $|X| < |Y|$ then there exists $e \in Y \setminus X$ such that $X \cup \{e\} \in \mathcal{I}$. All bases of a matroid have the same cardinality, which is called the *rank* of the matroid and is denoted by $\rho(\mathcal{I})$. An example of matroids is a uniform matroid $(E, \mathcal{I})$, where $\mathcal{I} = \{S \subseteq E \mid |S| \leq r\}$ for some r. Note that the rank of this uniform matroid is r. Given two matroids $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$, the *matroid intersection* of $\mathcal{M}_1$ and $\mathcal{M}_2$ is defined by $(E, \mathcal{I}_1 \cap \mathcal{I}_2)$. Similarly, given μ matroids $\mathcal{M}_i = (E, \mathcal{I}_i)$ $(i = 1, \dots, \mu)$, the μ-matroid intersection is defined by $(E, \bigcap_{i=1}^{\mu} \mathcal{I}_i)$.

Given an item set E with size $s(e)$ and value $v(e)$ for each $e \in E$, and the capacity $C \in \mathbb{Z}_+$, the *knapsack problem* is to find a subset X of E that maximizes the total value $\sum_{e \in X} v(e)$ subject to a knapsack constraint $\sum_{e \in X} s(e) \leq C$. Each subset satisfying the knapsack constraint is called a knapsack solution. Let $\mathcal{I} = \{X \mid \sum_{e \in X} s(e) \leq C\}$ be the family of knapsack solutions. Then, $(E, \mathcal{I})$ is an independence system.

Robust optimization problem Let E be a finite ground set, and let n be a positive integer. Let $\Delta(\mathcal{I})$ and Δ_n denote the set of probability distributions over a family $\mathcal{I} \subseteq 2^E$ and $[n]$, respectively. Given n set functions $f_1, \dots, f_n\colon 2^E \to \mathbb{R}_+$ and an independence system $(E, \mathcal{I})$, our task is to solve

$$\max \min_{k \in [n]} \sum_{X \in \mathcal{I}} p_X \cdot f_k(X) \quad \text{s.t.} \quad p \in \Delta(\mathcal{I}). \qquad (1)$$

For each $k \in [n]$, we denote $X_k^* \in \arg\max_{X \in \mathcal{I}} f_k(X)$ and assume that $f_k(X_k^*) > 0$. We assume that the functions are given by an *oracle*, i.e., for a given $X \subseteq E$, we can query an oracle about the values $f_1(X), \dots, f_n(X)$.

By von Neumann's minimax theorem, it holds that

$$\max_{p \in \Delta(\mathcal{I})} \min_{k \in [n]} \sum_{X \in \mathcal{I}} p_X f_k(X) = \min_{q \in \Delta_n} \max_{X \in \mathcal{I}} \sum_{k \in [n]} q_k f_k(X). \qquad (2)$$

This leads the following proposition, which is used later.

Table 1: The approximation ratios for robust optimization problems shown in the present paper.

	objective functions	constraint	approx. ratio	ref.
LP-based	linear (polytope)	matroid (intersection)	P	Theorem 3
	linear (polytope)	shortest s–t path	P	Theorem 3
	linear (polytope)	knapsack	PTAS	Theorem 6
	linear (polytope)	μ-matroid intersection	$\frac{2}{e\mu}$-approx.	Theorem 7
MWU-based	monotone submodular	matroid/knapsack	$(1-\frac{1}{e}-\epsilon)$-approx.	Theorem 9
	monotone submodular	μ-matroid intersection	$\frac{1}{\mu+\epsilon}$-approx.	Theorem 10
	linear	μ-matroid intersection	$\frac{1}{\mu-1+\epsilon}$-approx.	Theorem 11
	submodular	free	$(\frac{1}{2}-\epsilon)$-approx.	Theorem 12
	linear	knapsack	FPTAS	Theorem 13
	cardinality	knapsack	FPTAS	Theorem 14

Proposition 1. *Let ν^* denote the optimal value of* (1). *It holds that* $\min_{k\in[n]} f_k(X_k^*)/n \le \nu^* \le \min_{k\in[n]} f_k(X_k^*)$.

This implies that we can find a $1/n$-approximate solution by just computing X_k^* ($k \in [n]$).

We describe two merits to adopt a randomized strategy rather than a deterministic one for (1). One is that the randomization improves the worst case value dramatically. Suppose that $\mathcal{I} = \{\emptyset, \{a\}, \{b\}\}$, $f_1(X) = |X \cap \{a\}|$, and $f_2(X) = |X \cap \{b\}|$. Then, the maximum worst case value among deterministic strategies is $\max_{X\in\mathcal{I}} \min_{k\in\{1,2\}} f_k(X) = 0$, while that for randomized ones is $\max_{p\in\Delta(\mathcal{I})} \min_{k\in\{1,2\}} \sum_{X\in\mathcal{I}} p_X \cdot f_k(X) = 1/2$. The other merit is that the optimal randomized strategy can be found easier. It is known that finding an optimal deterministic solution is hard even in a simple setting (Aissi, Bazgan, and Vanderpooten 2009; Kasperski and Zieliński 2016). In particular, we see in the following that even for an easy case, computing the optimal worst case value among deterministic solutions is strongly NP-hard even to approximate.

Theorem 1. *It is NP-hard to compute $X \in \mathcal{I}$ maximizing $\min_{k\in[n]} f_k(X)$ even when the objective functions $f_1, \dots, f_k$ are linear and $\mathcal{I}$ is given by a uniform matroid. Moreover, there exists no approximation algorithm for the problem unless P=NP.*

Note that we will show that the randomized version of this problem is polynomial-time solvable (Theorem 3).

Application 1: security game In a security game, we are given n targets E. The defender selects a set of targets $X \in \mathcal{I} \subseteq 2^E$, and then the attacker selects a facility $i \in E$. The utility of defender is r_i if $i \in X$ and c_i if $i \notin X$. Then, we can interpret the game as the robust optimization with $f_i(X) = c_i + \sum_{j\in X} w_{ij}$ where $w_{ij} = r_i - c_i$ if $i = j$ and 0 if $i \neq j$ for $i, j \in E$. Then the problem of computing the Stackelberg equilibrium is equivalent to (1).

Application 2: MCRP Consider that given an independence system $(E, \mathcal{I})$ with weights of elements in E, we choose $X \in \mathcal{I}$ of size at most k with as large total weight as possible, but k is not known in advance. For each $X \in \mathcal{I}$, we denote the total weight of the k heaviest elements in X by $v_{\le k}(X)$. For $\alpha \in [0, 1]$, an independent set $X \in \mathcal{I}$ is said to be *α-robust* if $v_{\le k}(X) \ge \alpha \cdot \max_{Y\in\mathcal{I}} v_{\le k}(Y)$ for any $k \in [n]$. Then, MCRP is to find a randomized strategy that maximizes the robustness α, i.e., $\max_{p\in\Delta(\mathcal{I})} \min_{k\in[n]} \sum_{X\in\mathcal{I}} p_X \cdot v_{\le k}(X)/ \max_{Y\in\mathcal{I}} v_{\le k}(Y)$. This is formulated as (1) by setting $f_k(X) = v_{\le k}(X)/\max_{Y\in\mathcal{I}} v_{\le k}(Y)$.

3 LP-based Algorithms

In this section, we propose a computation scheme for the robust optimization problem (1) with linear functions $f_1, \dots, f_n$, i.e., $f_k(X) = \sum_{e\in X} w_{ke}$. Here, $w_{ke} \ge 0$ holds for $k \in [n]$ and $e \in E$ since we assume $f_k(X) \ge 0$. A key technique is the separation problem for an independence system polytope. An *independence system polytope* of $(E, \mathcal{I})$ is a polytope defined as $P(\mathcal{I}) = \mathrm{conv}\{\chi(X) \mid X \in \mathcal{I}\} \subseteq [0,1]^E$, where $\chi(X)$ is a characteristic vector in $\{0,1\}^E$, i.e., $\chi(X)_e = 1$ if and only if $e \in X$. For a probability distribution $p \in \Delta(\mathcal{I})$, we can get a point $x \in P(\mathcal{I})$ such that $x_e = \sum_{X\in\mathcal{I}:\, e\in X} p_X$ ($e \in E$). Then, x_e ($e \in E$) means a probability that e is chosen when we select an independent set according to the probability distribution p. Conversely, for any $x \in P(\mathcal{I})$, there exists $p \in \Delta(\mathcal{I})$ such that $\sum_{X\in\mathcal{I}} p_X\, \chi(X) = x$ by the definition of $P(\mathcal{I})$. Given $x \in \mathbb{R}^E$, the separation problem for $P(\mathcal{I})$ is to either assert $x \in P(\mathcal{I})$ or find a vector d such that $d^\top x < d^\top y$ for all $y \in P(\mathcal{I})$.

The rest of this section is organized as follows. In Section 3.1, we prove that we can solve (1) in polynomial time if there is a polynomial-time algorithm to solve the separation problem for $P(\mathcal{I})$. In Section 3.2, we tackle the case when it is hard to construct a separation algorithm for $P(\mathcal{I})$. We show that we can obtain an approximation solution when we can slightly relax $P(\mathcal{I})$. Moreover, we deal with a setting that objective functions are given by a polytope in Section 3.3.

3.1 Basic scheme

We observe that the optimal robust value of (1) is equal to the optimal value of the following linear programming (LP):

$$\max\ \nu\ \text{ s.t. }\ \nu \le \sum_{e\in E} w_{ie} x_e\ \ (\forall i \in [n]),\ \ x \in P(\mathcal{I}). \quad (3)$$

Lemma 1. *When $f_1, \dots, f_n$ are linear, the optimal value of* (3) *is equal to that of* (1).

Thus the optimal solution of (1) is obtained by the following two-step scheme.

1. compute the optimal solution of LP (3), which we denote as (ν^*, x^*),
2. compute $p^* \in \Delta(\mathcal{I})$ such that $x^* = \sum_{X \in \mathcal{I}} p^*_X\, \chi(X)$.

It is trivial that if $|\mathcal{I}|$ is bounded by a polynomial in $|E|$ and n, then we can obtain p^* by replacing x with $\sum_{X \in \mathcal{I}} p_X \chi(X)$ in (3) and solving it. In general, we can solve the two problems in polynomial time by the ellipsoid method when we have a polynomial-time algorithm to solve the separation problem for $P(\mathcal{I})$. This is due to the following theorems given by Grötschel, Lovász, and Schrijver (2012).

Theorem 2 (Grötschel, Lovász, and Schrijver)**.** *Let $\mathcal{P} \subseteq \mathbb{R}^E$ be a polytope and suppose that the separation problem for $\mathcal{P}$ can be solved in polynomial time. Then we can solve a linear program over $\mathcal{P}$ in polynomial time. In addition, there exists a polynomial time algorithm that, for any vector $x \in \mathcal{P}$, computes affinely independent vertices $x_1, \dots, x_\ell$ of $\mathcal{P}$ ($\ell \le |E| + 1$) and positive reals $\lambda_1, \dots, \lambda_\ell$ with $\sum_{i=1}^{\ell} \lambda_i = 1$ such that $x = \sum_{i=1}^{\ell} \lambda_i x_i$.*

Therefore, we have the following general result.

Theorem 3. *If $f_1, \dots, f_n$ are linear and there is a polynomial-time algorithm to solve the separation problem for $P(\mathcal{I})$, then we can solve the linear robust optimization problem* (1) *in polynomial time.*

Hence, there exists a polynomial-time algorithm for (1) when $\mathcal{I}$ is a matroid (intersection) or the set of s–t paths, because a matroid (intersection) polytope and the dominant of an s–t path polytope admit a polynomial-time separation algorithm.

3.2 Relaxation of the polytope

We present an approximation scheme for the case when the separation problem for $P(\mathcal{I})$ is hard to solve. Recall that $f_k(X) = \sum_{e \in X} w_{ke}$ where $w_{ke} \ge 0$ for $k \in [n]$ and $e \in E$.

We modify the basic scheme as follows. First, instead of solving the separation problem for $P(\mathcal{I})$, we solve the one for a relaxation of $P(\mathcal{I})$. For a polytope P and a positive number $(1 \ge)\, \alpha > 0$, we denote $\alpha P = \{\alpha x \mid x \in P\}$. We call a polytope $\hat{P}(\mathcal{I}) \subseteq [0,1]^E$ α-relaxation of $P(\mathcal{I})$ if it holds that $\alpha \hat{P}(\mathcal{I}) \subseteq P(\mathcal{I}) \subseteq \hat{P}(\mathcal{I})$. Then we solve

$$\max_{x \in \hat{P}(\mathcal{I})} \min_{k \in [n]} \sum_{e \in E} w_{ke} x_e \tag{4}$$

instead of LP (3), and obtain an optimal solution $\hat{x}$.

Next, we compute a convex combination of $\hat{x}$ using $\chi(X)$ $(X \in \mathcal{I})$. Here, if $\hat{x} \in \hat{P}(\mathcal{I})$ is the optimal solution for (4), then $\alpha\hat{x} \in P(\mathcal{I})$ is an α-approximate solution of LP (3), because

$$\begin{aligned} \max_{x \in P(\mathcal{I})} \min_{k \in [n]} \sum_{e \in E} w_{ke} x_e &\le \max_{x \in \hat{P}(\mathcal{I})} \min_{k \in [n]} \sum_{e \in E} w_{ke} x_e \\ &= \min_{k \in [n]} \sum_{e \in E} w_{ke} \hat{x}_e = \frac{1}{\alpha} \cdot \min_{k \in [n]} \sum_{e \in E} w_{ke} (\alpha \hat{x}_e). \end{aligned}$$

As $\alpha\hat{x} \in P(\mathcal{I})$, there exists $p \in \Delta(\mathcal{I})$ such that $\alpha\hat{x} = \sum_{X \in \mathcal{I}} p_X\, \chi(X)$. However, the retrieval of such a probability distribution may be computationally hard, because the separation problem for $P(\mathcal{I})$ is hard to solve. Hence, we relax the problem and compute $p^* \in \Delta(\mathcal{I})$ such that $\beta\hat{x} \le \sum_{X \in \mathcal{I}} p^*_X\, \chi(X)$, where $(\alpha \ge)\, \beta > 0$. Then, p^* is a β-approximate solution of $\max_{p \in \Delta(\mathcal{I})} \min_{k \in [n]} \sum_{X \in \mathcal{I}} p^*_X \cdot f_k(X)$, because

$$\begin{aligned} \max_{p \in \Delta(\mathcal{I})} \min_{k \in [n]} \sum_{X \in \mathcal{I}} p_X \cdot f_k(X) &\le \min_{k \in [n]} \sum_{e \in E} w_{ke} \hat{x}_e \\ &\le \frac{1}{\beta} \cdot \min_{k \in [n]} \sum_{X \in \mathcal{I}} p^*_X \cdot f_k(X). \end{aligned}$$

Thus the basic scheme is modified as the following approximation scheme:

1. compute the optimal solution $\hat{x} \in \hat{P}(\mathcal{I})$ for LP (4),
2. compute $p^* \in \Delta(\mathcal{I})$ such that $\beta \cdot \hat{x}_e \le \sum_{X \in \mathcal{I} \colon e \in X} p^*_X$ for each $e \in E$.

Theorem 4. *Suppose that $f_1, \dots, f_n$ are linear. If there exists a polynomial-time algorithm to solve the separation problem for an α-relaxation $\hat{P}(\mathcal{I})$ of $P(\mathcal{I})$, then an α-approximation of the optimal value of* (1) *is computed in polynomial-time. In addition, if there exists a polynomial-time algorithm to find $p \in \Delta(\mathcal{I})$ such that $\beta \cdot \hat{x}_e \le \sum_{X \in \mathcal{I} \colon e \in X} p_X$ for any $x \in \hat{P}(\mathcal{I})$, then a β-approximate solution of* (1) *is found in polynomial-time.*

We remark that we can combine the result in Section 3.3 with this theorem.

In the following, we apply Theorem 4 to two important cases when $\mathcal{I}$ is defined from a knapsack constraint or a μ-matroid intersection. For this end, we develop appropriately relaxations of $P(\mathcal{I})$ and retrieval procedures for p^*.

Relaxation of a knapsack polytope Let E be a set of items with size $s(e)$ for each $e \in E$. Without loss of generality, we assume that a knapsack capacity is one, and $s(e) \le 1$ for all $e \in E$. Let $\mathcal{I}$ be a family of knapsack solutions, i.e., $\mathcal{I} = \{X \subseteq E \mid \sum_{e \in X} s(e) \le 1\}$.

It is known that $P(\mathcal{I})$ admits a *polynomial size relaxation scheme*, i.e., there exists a $(1-\epsilon)$-relaxation of $P(\mathcal{I})$ through a linear program of polynomial size for a fixed $\epsilon > 0$.

Theorem 5 (Bienstock)**.** *Let $0 < \epsilon \le 1$. There exist a polytope $P^\epsilon(\mathcal{I})$ and its extended formulation with $O(\epsilon^{-1} n^{1+\lceil 1/\epsilon \rceil})$ variables and $O(\epsilon^{-1} n^{2+\lceil 1/\epsilon \rceil})$ constraints such that $(1-\epsilon) P^\epsilon(\mathcal{I}) \subseteq P(\mathcal{I}) \subseteq P^\epsilon(\mathcal{I})$.*

Thus, the optimal solution $\hat{x}$ to $\max_{x \in P^\epsilon(\mathcal{I})} \min_{k \in [n]} \sum_{e \in E} w_{ke} x_e$ can be computed in polynomial time. The remaining task is to compute $p^* \in \Delta(\mathcal{I})$ such that $(1-\epsilon) \cdot \hat{x}_e \le \sum_{X \in \mathcal{I} \colon e \in X} p^*_X$ for each $e \in E$. We give an algorithm for this task.

Lemma 2. *There exists a polynomial-time algorithm that computes $p^* \in \Delta(\mathcal{I})$ such that $(1-\epsilon) \cdot \hat{x}_e \le \sum_{X \in \mathcal{I} \colon e \in X} p^*_X$ for each $e \in E$.*

Theorem 6. *There is a PTAS to compute the linear robust optimization problem* (1) *subject to a knapsack constraint.*

Finally, we remark that the existence of a *fully polynomial size relaxation scheme (FPSRS)* for $P(\mathcal{I})$ is open (Bienstock 2008). The existence of an FPSRS leads an FPTAS to compute the optimal value of the linear robust optimization problem (1) subject to a knapsack constraint.

Relaxation of a μ-matroid intersection polytope Let us consider the case where $\mathcal{I}$ is defined from a μ-matroid intersection. It is NP-hard to maximize a linear function subject to a μ-matroid intersection constraint if $\mu \geq 3$ (Garey and Johnson 1979). Hence, it is also NP-hard to solve the linear robust optimization subject to a μ-matroid intersection constraint if $\mu \geq 3$. For $i = 1, \ldots, \mu$, let $(E, \mathcal{I}_i)$ be a matroid whose rank function is ρ_i. Let $(E, \mathcal{I}) = (E, \bigcap_{i\in[\mu]} \mathcal{I}_i)$. We define $\hat{P}(\mathcal{I}) = \bigcap_{i\in[\mu]} P(\mathcal{I}_i)$. We see that $\hat{P}(\mathcal{I})$ is a $(1/\mu)$-relaxation of $P(\mathcal{I})$.

Lemma 3. $\frac{1}{\mu}\hat{P}(\mathcal{I}) \subseteq P(\mathcal{I}) \subseteq \hat{P}(\mathcal{I})$.

As we can solve the separation problem for $\hat{P}(\mathcal{I})$ in strongly polynomial time (Cunningham 1984), we can obtain an optimal solution $\hat{x} \in \hat{P}(\mathcal{I})$ for the relaxed problem $\max_{x\in\hat{P}(\mathcal{I})} \sum_{e\in E} w(e)x_e$. Since $\hat{x}/\mu \in P(\mathcal{I})$, the value $\sum_{e\in E} w(e)\hat{x}_e/\mu$ is a μ-approximation of the optimal value.

To obtain a μ-approximate solution, we need to compute $p^* \in \Delta(\mathcal{I})$ such that $\hat{x}_e/\mu \leq \sum_{X\in\mathcal{I}:\, e\in X} p^*_X$ for each $e \in E$. Unfortunately, it seems hard to obtain such a distribution. With the aid of the contention resolution (CR) scheme (Chekuri, Vondrák, and Zenklusen 2014), we can compute $p^* \in \Delta(\mathcal{I})$ such that $(2/e\mu)\cdot\hat{x}_e \leq \sum_{X\in\mathcal{I}:\, e\in X} p^*_X$ for each $e \in E$. We describe its procedure in the full version (Kawase and Sumita 2018). We can summarize our result as follows.

Theorem 7. *We can compute a μ-approximate value of the linear robust optimization problem subject to a μ-matroid intersection in polynomial time. Moreover, we can implement a procedure that efficiently outputs an independent set according to the distribution of a $2/(e\mu)$-approximate solution.*

3.3 Linear functions in a polytope

We consider the following variant of (1). Instead of n functions $f_1, \ldots, f_n$, suppose that we are given a set of functions

$$\mathcal{F} = \left\{ f \;\middle|\; f(\{e\}) = w_e \ (\forall e \in E),\ \begin{array}{l} Aw + B\psi \leq c, \\ w \geq 0, \\ \psi \geq 0 \end{array} \right\}$$

for some $A \in \mathbb{R}^{m\times|E|}$, $B \in \mathbb{R}^{m\times d}$, and $c \in \mathbb{R}^m$. Now, we aim to solve

$$\max_{p\in\Delta(\mathcal{I})} \min_{f\in\mathcal{F}} \sum_{X\in\mathcal{I}} p_X f(X). \tag{5}$$

Note that for linear functions $f_1, \ldots, f_n$, (1) is equivalent to (5) in which

$$\begin{aligned}\mathcal{F} &= \mathrm{conv}\{f_1, \ldots, f_n\} \\ &= \left\{ f \;\middle|\; \begin{array}{l} f(\{e\}) = w_e \ (e \in E), \\ w_e = \sum_{k\in[n]} q_k f_k(\{e\}), \\ \sum_{k\in[n]} q_k = 1,\ w, q \geq 0 \end{array} \right\}.\end{aligned}$$

We observe that (5) is equal to $\max_{x\in P(\mathcal{I})} \min\{x^\top w \mid Aw + B\psi \leq c,\ w \geq 0,\ \psi \geq 0\}$ by using a similar argument to Lemma 1. The LP duality implies that $\min\{x^\top w \mid Aw + B\psi \leq c,\ w \geq 0,\ \psi \geq 0\} = \max\{c^\top y \mid A^\top y \geq x,\ B^\top y \geq 0,\ y \geq 0\}$. Thus the optimal value of (5) is equal to that of the LP $\max_{x\in P(\mathcal{I})} \max_{y:\, A^\top y = x,\, B^\top y\geq 0,\, y\geq 0} \sum_{i\in[m]} b_i y_i$. Hence, Theorem 2 implies that if the separation problem for $P(\mathcal{I})$ can be solved in polynomial time, then we can solve (5) in polynomial time.

4 MWU-based Algorithm

In this section, we present an algorithm based on the MWU method (Arora, Hazan, and Kale 2012). This algorithm is applicable to general cases. We assume that $f_k(X) \geq 0$ for any $k \in [n]$ and $X \in \mathcal{I}$.

We describe the idea of our algorithm. Let us focus on the right hand side of the minimax relation (2). We define weights ω_k for each function f_k, and iteratively update them. Intuitively, a function with a larger weight is likely to be chosen with higher probability. At the first round, all functions have the same weights. At each round t, we set a probability q_k ($k \in [n]$) that f_k is chosen by normalizing the weights. Then we compute an (approximate) optimal solution $X^{(t)}$ of $\max_{X\in\mathcal{I}} \sum_{k\in[n]} q_k \cdot f_k(X)$. To minimize the right hand side of (2), the probability q_k for a function f_k with a larger value $f_k(X)$ should be decreased. Thus we update the weights according to $f_k(X)$. We repeat this procedure, and set a randomized strategy $p \in \Delta(\mathcal{I})$ according to $X^{(t)}$'s.

Krause, Roper, and Golovin (2011) and Chen et al. (2017) proposed the above algorithm when $f_1, \ldots, f_n$ are functions with range $[0, 1]$. They proved that if there exists an α-approximation algorithm to $\max_{X\in\mathcal{I}} \sum_{k\in[n]} q_k f_k(X)$ for any $q \in \Delta_n$, then the approximation ratio is $\alpha - \epsilon$ for any fixed constant $\epsilon > 0$. This implies an approximation ratio of $\alpha - \epsilon \cdot \max_{k\in[n],\, X\in\mathcal{I}} f_k(X)/\nu^*$ when $f_1, \ldots, f_n$ are functions with range $\mathbb{R}_+$, where ν^* is the optimal value of (1). Here, $\max_{k\in[n],\, X\in\mathcal{I}} f_k(X)/\nu^*$ could be large in general. To remove this term from the approximation ratio, we introduce a novel concept of function transformation. We improve the existing algorithms (Chen et al. 2017; Krause, Roper, and Golovin 2011) with this concept, and show a stronger result later in Theorem 8.

Definition 1. *For positive reals η and γ (≤ 1), we call a function g is an (η, γ)*-reduction *of f if (i) $g(X) \leq \min\{f(X), \eta\}$ and (ii) $g(X) \leq \gamma\cdot\eta$ implies $g(X) = f(X)$.*

Intuitively, the condition (i) is useful for speeding up MWU and the condition (ii) is important for the purpose that the optimal value does not change significantly.

We fix a parameter $\gamma > 0$, where $1/\gamma$ is bounded by polynomial. The smaller γ is, the wider the class of (η, γ)-reduction of f is. We set another parameter η later. We denote (η, γ)-reduction of $f_1, \ldots, f_n$ by $f_1^\eta, \ldots, f_n^\eta$, respectively. In what follows, suppose that we have an α-approximation algorithm to

$$\max_{X\in\mathcal{I}} \sum_{k\in[n]} q_k f_k^\eta(X) \tag{6}$$

for any $q \in \Delta_n$ and $\eta \in \mathbb{R}_+ \cup \{\infty\}$. In our proposed algorithm, we use f_k^η instead of the original f_k. The smaller

η is, the faster our algorithm converges. However, the limit outcome of our algorithm as T goes to infinity moves over a little from the optimal solution. We overcome this issue by setting η to an appropriate value.

Our algorithm is summarized in Algorithm 1. Note that $f_k^\infty = f_k$ $(k \in [n])$. We remark that when the parameter γ is small, there may exist a better approximation algorithm for (6), but the running time of Algorithm 1 becomes longer.

Algorithm 1: MWU for the robust optimization

input : positive reals η, δ $(\leq 1/2)$, and an integer T
output: randomized strategy $p^* \in \Delta(\mathcal{I})$

Let $\omega_k^{(1)} \leftarrow 1$ for each $k \in [n]$;
for $t = 1, \dots, T$ **do**
  $q_k^{(t)} \leftarrow \omega_k^{(t)} / \sum_{k \in [n]} \omega_k^{(t)}$ for each $k \in [n]$;
  let $X^{(t)}$ be an α-approximate solution of $\max_{X \in \mathcal{I}} \sum_{k \in [n]} q_k^{(t)} \cdot f_k^\eta(X)$;
  $\omega_k^{(t+1)} \leftarrow \omega_k^{(t)} (1-\delta)^{f_k^\eta(X^{(t)})/\eta}$ for each $k \in [n]$;
return $p^* \in \Delta(\mathcal{I})$ such that $p_X^* = |\{t \in \{1, \dots, T\} \mid X^{(t)} = X\}|/T$;

The main result of this section is stated below.

Theorem 8. *If there exists an α-approximation algorithm to solve* (6) *for any $q \in \Delta_n$ and $\eta \in \mathbb{R}_+ \cup \{\infty\}$, then Algorithm 1 is an $(\alpha - \epsilon)$-approximation algorithm to the robust optimization problem* (1) *for any fixed $\epsilon > 0$. In addition, the running time of Algorithm 1 is $O(\frac{n^2 \ln n}{\alpha \epsilon^3 \gamma}\theta)$, where θ is the running time of the α-approximation algorithm to* (6)*.*

To show this, we use the following lemma, which can be proved by standard analysis of the MWU method (see, e.g., Arora, Hazan, and Kale (2012)). In the following, we denote by ν^* the optimal value of (1).

Lemma 4. *For any $\delta \in (0, 1/2]$, it holds that*

$$\sum_{t=1}^{T} \sum_{k \in [n]} q_k^{(t)} \cdot f_k^\eta(X^{(t)}) \leq \frac{\eta \ln n}{\delta} + (1+\delta) \cdot \min_{k \in [n]} \sum_{t=1}^{T} f_k^\eta(X^{(t)}).$$

Next, we see that the optimal value of (1) for $f_1, \dots, f_n$ and the one for $f_1^\eta, \dots, f_n^\eta$ are close if η is a large number.

Lemma 5. *If $\eta \geq \frac{n}{\delta\gamma} \cdot \nu^*$, we have*

$$\nu^* \geq \min_{q \in \Delta_n} \max_{X \in \mathcal{I}} \sum_{k \in [n]} q_k \cdot f_k^\eta(X) \geq (1-\delta)\nu^*.$$

By Lemmas 4 and 5, we have the following lemma, which implies Theorem 8.

Lemma 6. *For any fixed $\epsilon > 0$, the output p^* of Algorithm 1 is an $(\alpha - \epsilon)$-approximate solution of* (1) *when we set $T = \lceil \frac{n^2 \ln n}{\alpha \delta^3 \gamma} \rceil$, $\frac{n^2}{\alpha\delta\gamma}\nu^* \geq \eta \geq \frac{n}{\delta\gamma}\nu^*$, and $\delta = \min\{\epsilon/3, 1/2\}$.*

As applications of Theorem 8, we can obtain the following theorems.

When $f_1, \dots, f_n$ are monotone submodular, $f_k^\eta(X) = \min\{f_k(X), \eta\}$ is an $(\eta, 1)$-reduction of f_k, and f_k^η is a monotone submodular function (Lovász 1983; Fujito 2000). Thus, $\sum_{k \in [n]} q_k f_k^\eta(X)$ is monotone submodular for any $q \in \Delta_n$. Because there exist $(1 - 1/e)$-approximation algorithms for maximizing a monotone submodular function under a knapsack constraint (Sviridenko 2004) and under a matroid constraint (Calinescu et al. 2007; Filmus and Ward 2012), we can obtain the following theorems.

Theorem 9. *For any positive real $\epsilon > 0$, there exists a $(1 - 1/e - \epsilon)$-approximation algorithm for the robust optimization problem* (1) *when $f_1, \dots, f_n$ are monotone submodular and $\mathcal{I}$ is given by a knapsack constraint or a matroid.*

Theorem 10. *For any fixed positive real $\epsilon > 0$, there exists a $1/(\mu + \epsilon)$-approximation algorithm for the robust optimization problem* (1) *when $f_1, \dots, f_n$ are monotone submodular and $\mathcal{I}$ is given by a μ-matroid intersection.*

A monotone linear maximization subject to a μ-matroid intersection can be viewed as a monotone submodular maximization subject to a $(\mu-1)$-matroid intersection. Thus, we also obtain the following theorem.

Theorem 11. *For any fixed positive real $\epsilon > 0$, there exists a $1/(\mu - 1 + \epsilon)$ for the robust optimization problem* (1) *when $f_1, \dots, f_n$ are monotone linear and $\mathcal{I}$ is given by a μ-matroid intersection.*

When $f_1, \dots, f_n$ are (non-monotone) submodular, $\min\{f_k, \eta\}$ may not a submodular function. In this case, we define $f_k^\eta(X) = \min\{f(Z) + \eta \cdot |X - Z|/|E| \mid Z \subseteq X\}$. Then, f_k^η is an $(\eta, 1/|E|)$-reduction of f_k. Since $f_k^\eta(X)$ is a submodular function (Fujishige 2005), we can evaluate the value $f_k^\eta(X)$ in strongly polynomial time by a submodular function minimization algorithm. Thus, the following theorem holds.

Theorem 12. *For any fixed positive real $\epsilon > 0$, there exists a $(1/2 - \epsilon)$-approximation algorithm for the robust optimization problem* (1) *when $f_1, \dots, f_n$ are submodular and $\mathcal{I} = 2^E$.*

When $f_k(X) = \sum_{e \in X} w_{ke}$ for each $k \in [n]$, where $w_{ke} \geq 0$ and $e \in E$, $f_k^\eta(X) = \sum_{e \in X} \min\{w_{ke}, \eta/|E|\}$ is an $(\eta, 1/|E|)$-reduction of f_k. In addition, we can construct an FPTAS to compute $\max_{X \in \mathcal{I}} \sum_{k \in [n]} q_k f_k^\eta(X)$ for any $q \in \Delta_n$.

Theorem 13. *There exists an FPTAS for the robust optimization problem* (1) *when $f_1, \dots, f_n$ are monotone linear and $\mathcal{I}$ is given by a knapsack constraint.*

Finally, we apply Theorem 8 to MCRP for the knapsack problem. Recall that the objective functions are given as $f_k(X) = v_{\leq k}(X)/\max_{Y \in \mathcal{I}} v_{\leq k}(Y)$ for $i \in [k]$. Note that the evaluation of $f_k(X)$ for a given solution X is already NP-hard. We provide an FPTAS to evaluate the value of f_k's and then we develop an FPTAS to solve $\max_{X \in \mathcal{I}} \sum_{k \in [n]} q_k f_k^\eta(X)$. Therefore, we can obtain the following theorem.

Theorem 14. *There exists an FPTAS to solve the maximum cardinality robustness problem for the knapsack problem.*

Acknowledgments

The first author is supported by JSPS KAKENHI Grant Number JP16K16005 and JST ACT-I Grant Number JPMJPR17U7. The second author is supported by JST ERATO Grant Number JPMJER1201, Japan, and JSPS KAKENHI Grant Number JP17K12646.

References

Aissi, H.; Bazgan, C.; and Vanderpooten, D. 2009. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *EJOR* 197(2):427–438.

Arora, S.; Hazan, E.; and Kale, S. 2012. The multiplicative weights update method: a meta-algorithm and applications. *Theor. Comp.* 8(1):121–164.

Bienstock, D. 2008. Approximate formulations for 0–1 knapsack sets. *Oper. Res. Let.* 36(3):317–320.

Bowles, S. 2009. *Microeconomics: behavior, institutions, and evolution*. Princeton University Press.

Buchbinder, N.; Feldman, M.; Naor, J. S.; and Schwartz, R. 2015. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SICOMP* 44(5):1384–1402.

Calinescu, G.; Chekuri, C.; Pál, M.; and Vondrák, J. 2007. Maximizing a submodular set function subject to a matroid constraint. In *IPCO*, volume 7, 182–196. Springer.

Caprara, A.; Kellerer, H.; Pferschy, U.; and Pisinger, D. 2000. Approximation algorithms for knapsack problems with cardinality constraints. *EJOR* 123(2):333–345.

Chekuri, C.; Vondrák, J.; and Zenklusen, R. 2014. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SICOMP* 43(6):1831–1879.

Chen, R. S.; Lucier, B.; Singer, Y.; and Syrgkanis, V. 2017. Robust optimization for non-convex objectives. In *Advances in Neural Information Processing Systems*, 4708–4717.

Cunningham, W. H. 1984. Testing membership in matroid polyhedra. *JCTB* 36(2):161–188.

Feige, U.; Mirrokni, V. S.; and Vondrák, J. 2011. Maximizing non-monotone submodular functions. *SICOMP* 40(4):1133–1153.

Filmus, Y., and Ward, J. 2012. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *Proc. of FOCS*, 659–668. IEEE.

Freund, Y., and Schapire, R. E. 1999. Adaptive game playing using multiplicative weights. *GEB* 29(1-2):79–103.

Fujishige, S. 2005. *Submodular functions and optimization*, volume 58. Elsevier.

Fujita, R.; Kobayashi, Y.; and Makino, K. 2013. Robust matchings and matroid intersections. *SIDMA* 27:1234–1256.

Fujito, T. 2000. Approximation algorithms for submodular set cover with applications. *IEICE Transactions on Information and Systems* 83(3):480–487.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman New York.

Grötschel, M.; Lovász, L.; and Schrijver, A. 2012. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media.

Hassin, R., and Rubinstein, S. 2002. Robust matchings. *SIDMA* 15(4):530–537.

Hellerstein, L.; Lidbetter, T.; and Pirutinsky, D. 2018. Solving zero-sum games using best response oracles with applications to search games. *arXiv:1704.02657v4*.

Kakimura, N., and Makino, K. 2013. Robust independence systems. *SIDMA* 27(3):1257–1273.

Kakimura, N.; Makino, K.; and Seimi, K. 2012. Computing knapsack solutions with cardinality robustness. *Japan Journal of Industrial and Applied Mathematics* 29(3):469–483.

Kale, S. 2007. *Efficient algorithms using the multiplicative weights update method*. Ph.D. Dissertation, Princeton University.

Kasperski, A., and Zieliński, P. 2016. *Robust Discrete Optimization Under Discrete and Interval Uncertainty: A Survey*. Springer International Publishing. 113–143.

Kawase, Y., and Sumita, H. 2018. Randomized strategies for robust combinatorial optimization. *arXiv:1805.07809*.

Kellerer, H.; Mansini, R.; and Speranza, M. G. 2000. Two linear approximation algorithms for the subset-sum problem. *EJOR* 120:289–296.

Kobayashi, Y., and Takazawa, K. 2016. Randomized strategies for cardinality robustness in the knapsack problem. *Theoretical Computer Science*.

Krause, A., and Golovin, D. 2014. Submodular function maximization. In *In Tractability: Practical Approaches to Hard Problems (to appear)*. Cambridge University Press.

Krause, A.; Roper, A.; and Golovin, D. 2011. Randomized sensing in adversarial environments. In *Proc. of IJCAI*, volume 22, 2133–2139.

Lee, J.; Sviridenko, M.; and Vondrák, J. 2010. Submodular maximization over multiple matroids via generalized exchange properties. *Math. Oper. Res.* 35(4):795–806.

Lovász, L. 1983. Submodular functions and convexity. In *Mathematical Programming The State of the Art*. Springer. 235–257.

Matuschke, J.; Skutella, M.; and Soto, J. A. 2018. Robust randomized matchings. *Math. Oper. Res.* 43(2):675–692.

Nisan, N.; Roughgarden, T.; Tardos, É.; and Vazirani, V. V. 2007. *Algorithmic Game Theory*. Cambridge University Press.

Sviridenko, M. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Let.* 32(1):41–43.

Tambe, M. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press.