MWSR over an Uplink Gaussian Channel with Box Constraints: A Polymatroidal Approach

Peng-Jun Wan*
Dongguan University of Technology
Guandong, China

Zhu Wang
SUNY at Oneonta
New York, USA

Huaqiang Yuan
Dongguan University of Technology
Guandong, China

Jiliang Wang
Tsinghua University
Beijing, China

Jinling Zhang
Renmin University of China
Beijing, China

ABSTRACT

The rate capacity region of an uplink Gaussian channel is a generalized symmetric polymatroid. Practical applications impose additional lower and upper bounds on the rate allocations, which are represented by box constraints. A fundamental scheduling problem over an uplink Gaussian channel is to seek a rate allocation maximizing the weighted sum-rate (MWSR) subject to the box constraints. The best-known algorithm for this problem has time complexity $O\left(n^5 \ln^{O(1)} n\right)$. In this paper, we take a polymatroidal approach to developing a quadratic-time greedy algorithm and a linearithmic-time divide-and-conquer algorithm. A key ingredient of these two algorithms is a linear-time algorithm for minimizing the difference between a generalized symmetric rank function and a modular function after a linearithmic-time ordering.

CCS CONCEPTS

• **Networks** → **Network algorithms**; • **Theory of computation** → **Algorithm design techniques**.

KEYWORDS

non-orthogonal multiple access, successive interference cancellation, polymatroid

ACM Reference Format:
Peng-Jun Wan, Zhu Wang, Huaqiang Yuan, Jiliang Wang, and Jinling Zhang. 2019. MWSR over an Uplink Gaussian Channel with Box Constraints: A Polymatroidal Approach. In *The Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '19), July 2–5, 2019, Catania, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3323679.3326519

*Also with Illinois Institute of Technology.

Mobihoc '19, July 2–5, 2019, Catania, Italy

ACM ISBN 978-1-4503-6764-6/19/07...$15.00
https://doi.org/10.1145/3323679.3326519

1 INTRODUCTION

Non-orthogonal multiple access (NOMA) [17] enhances bandwidth efficiency significantly by accommodating multiple users within the same orthogonal resource block, and is a promising technology of meeting the dramatically increasing demand for wireless capacity in the fifth generation (5G) networks. A basic technique of NOMA is the successive interference cancellation (SIC) [1, 2, 19, 30] at the receiver. Rather than decoding every user treating the interference from other users as noise, the receiver with SIC receiver decodes all users sequentially. After one user is decoded, its signal is stripped away from the aggregate received signal before the next user is decoded.

For an uplink Gaussian channel between a base station and a set E of n users, the rate-capacity region is well characterized (e.g. [29]). For each user $e \in E$, let $p(e)$ be the signal to noise ratio (SNR) of the signal from user e perceived by the base station. With SIC by the base station, the rate-capacity region of these users consists of all vectors $x \in \mathbb{R}_+^E$ (the set of non-negative vectors indexed by E) satisfying that

$$\sum_{e \in S} x(e) \leq \log\left(1 + \sum_{e \in S} p(e)\right)$$

for each $S \subseteq E$. For practical applications, additional upper and/or lower bounds have to be imposed on the rate allocations or requirements by the users. In many scenarios, serving a user at a rate below its minimum requirement is futile, while serving a user at a rate above its maximum requirement is wasteful. Furthermore, the maximum rate limits can be used to cap the rates of any set of users such as those that have subscribed to a lower tier of service. In general, the lower bound on rate allocation is specified by a vector a in the rate-capacity region, and the upper bound on rate allocation is specified by a vector $b \in \mathbb{R}_+^E$ with $b \geq a$. A vector x is said to be a feasible rate allocation if x lies in the rate capacity region and $a \leq x \leq b$. Suppose that each user $e \in E$ has a positive reward $w(e)$ per rate unit. The total reward accrued by a rate allocation x is $\sum_{e \in E} w(e) x(e)$. The problem **MWSR** over an uplink Gaussian channel with box constraints seeks a feasible rate allocation x with maximum total reward.

The problem **MWSR** over an uplink Gaussian channel with box constraints is a special case of linear optimization over a bi-truncated generalized symmetric polymatroid. A set-function r on

2^E (the collection of subsets of E) is said to be a *(polymatroidal) rank* [3] if it satisfies the following three properties:

- submodularity: for any $A, B \subseteq E$,
$$r(A) + r(B) \geq r(A \cup B) + r(A \cap B);$$
- monotonicity: for any $A \subset B \subseteq E$, $r(A) \leq r(B)$;
- normalization: $r(\emptyset) = 0$.

A special type of rank is the generalized symmetric rank [5]: r is said to be a *generalized symmetric rank* [5] if there exist an increasing and strictly concave function ϕ on $\mathbb{R}_+$ with $\phi(0) = 0$ and a positive vector $p \in \mathbb{R}_+^E$ such that

$$r(S) = \phi\left(\sum_{e \in S} p(e)\right) \tag{1}$$

for each $S \subseteq E$. Moreover, if $p(e) = 1$ for each $e \in E$, then r is known as a *symmetric rank* [15]. For the problem **MWSR** over an uplink Gaussian channel with box constraints, the function ϕ is defined by $\phi(t) = \log(1 + t)$ for each $t \in \mathbb{R}_+$. The *polymatroid* [3] of a rank r is the polytope

$$\Omega := \left\{x \in \mathbb{R}_+^E : \sum_{e \in S} x(e) \leq r(S), \forall S \subseteq E\right\}.$$

Given a vector $a \in \Omega$ and a vector $b \in \mathbb{R}_+^E$ with $b \geq a$, the *bi-truncated polymatroid* of r by a from below and b from above is the polytope

$$\Omega' := \Omega \cap \left\{x \in \mathbb{R}_+^E : a \leq x \leq b\right\}.$$

If r is a generalized symmetric (resp., symmetric) rank, then Ω is referred to as a *generalized symmetric (resp., symmetric) polymatroid*, and Ω' is referred to as a *bi-truncated generalized symmetric (resp., symmetric) polymatroid*. Suppose that each user $e \in E$ has a positive weight $w(e)$. The problem

$$\max_{x \in \Omega} \sum_{e \in E} w(e)\, x(e)$$

is referred to as the linear optimization problem over the polymatroid Ω; and the problem

$$\max_{x \in \Omega'} \sum_{e \in E} w(e)\, x(e)$$

is referred to as the linear optimization problem over the bi-truncated polymatroid Ω'. Thus, the problem **MWSR** over an uplink Gaussian channel with box constraints is a linear optimization over a bi-truncated generalized symmetric polymatroid.

The linear optimization over a polymatroid can be solved by a simple polymatroid greedy method [3]. This method requires n rank evaluations, and an ordering of the users in the decreasing order of weight, and thus takes at least linearithmic time. A bi-truncated polymatroid is not a polymatroid in general, and hence the polymatroid greedy method [3] cannot be directly applicable to the linear optimization over a bi-truncated polymatroid. However, the maximum total reward can only be achieved at maximal elements in the bi-truncated polymatroid, which also form the set of maximal elements of the polymatroid of a "bi-truncated" rank function [8, 9, 22]. Thus, the polymatroid greedy method [3] can be applied to this polymatroid. But the evaluation of the "bi-truncated" rank becomes quite extensive in general as it involves minimizing a submodular function. A number of polynomial-time but very expensive algorithms [7, 11, 13, 14, 16, 18, 20] have been proposed for minimizing a general submodular function. Among them, the fastest one is that of [16] that requires $O\left(n^4 \ln^{O(1)} n\right)$ running time. The overall running time is then $O\left(n^5 \ln^{O(1)} n\right)$. On the other hand, for linear optimization over some special *bi-truncated symmetric* polymatroids, efficient algorithms have been developed recently in [24–26]. A natural question is whether linear optimization over the broader class of *bi-truncated generalized symmetric* polymatroids can also be solved efficiently.

This paper takes a polymatroidal approach to the algorithmic study of the problem **MWSR** over an uplink Gaussian channel with box constraints. For the *broader* problem of linear optimization over a bi-truncated generalized symmetric polymatroid, we develop two efficient algorithms:

- A quadratic-time greedy algorithm: This algorithm is an application of the classic polymatroid greedy method [3].
- A linearithmic-time divide-and-conquer algorithm: This algorithm is an implementation of a refined decomposing method with proper data structures.

A key ingredient of these algorithms is the linear-time procedure for minimizing the difference between a generalized symmetric rank function and a modular function after a linearithmic-time ordering. The second algorithm above has a running time matching the best-known (polymatroid greedy) algorithm for linear optimization over a generalized symmetric polymatroid. In addition to the problem **MWSR** over an uplink Gaussian channel with box constraints, the above algorithms can also be directly applied to other networking scheduling problems such as those in [6, 23].

The following standard notations are used in this paper. A vector $x \in \mathbb{R}^E$ (the set of real vectors indexed by E) is often treated as a modular set function on 2^E defined by $x(S) = \sum_{e \in S} x(e)$ for each $S \subseteq E$. For two vectors $x, y \in \mathbb{R}^E$, we write $x \leq y$ to denote that $x(e) \leq y(e)$ for each $e \in E$, and $x < y$ to denote that $x \leq y$ and $x \neq y$. For a subset Φ of $\mathbb{R}^E$, a vector $x \in \mathbb{R}^E$ is called *maximal* in Φ if $x \in \Phi$ and there is no $y \in \Phi$ such that $x < y$. Suppose that A and B are disjoint subsets of E. For any $y \in \mathbb{R}^A$ and $z \in \mathbb{R}^B$, the *direct sum* of y and z, denoted by $y \oplus z$, is the vector $x \in \mathbb{R}^{A \cup B}$ with $x(e) = y(e)$ for each $e \in A$ and $x(e) = z(e)$ for each $e \in B$. For any $x \in \mathbb{R}^E$ and any $S \subseteq E$, let $x^S \in \mathbb{R}^S$ be the restriction of x on S. For each $e \in E$, we denote by χ^e the characteristic vector of e.

The rest of paper is organized as follows. In Section 2, some fundamental properties of polymatroids are introduced. In Section 3, we provide a full characterization of minimizers of rank-modulus difference. In Section 4 and Section 5 we present the quadratic-time greedy algorithm and the linearithmic-time divide-and-conquer algorithm respectively. Finally, we conclude this paper in Section 6.

2 FUNDAMENTALS OF POLYMATROID

In this section, we introduce some fundamental properties of polymatroids that are relevant to this paper. We further refer the interested reader to [9] for background on submodular set functions,

[21] for background on polymatroid and submodular function minimization.

Consider a finite ground set E, and a rank function r on 2^E. A vector $x \in \mathbb{R}^E$ is a *subbase* of r if $x \in \mathbb{R}_+^E$ and $x(S) \le r(S)$ for each $S \subseteq E$. The set of all subbases of r is called the *polymatroid* of r. Let x be a subbase of r. A subset $T \subseteq E$ is said to be *x-tight* w.r.t. r if $x(T) = r(T)$. The collection of all x-tight subsets w.r.t. r is closed under taking union and intersection [9, 21]. In particular, the union of all x-tight sets w.r.t. r is also x-tight w.r.t. r, and hence it is the unique *maximal x-tight set* w.r.t. r. Clearly, for any $e \in E$, e is not in the maximal x-tight set w.r.t. r if and only if $x + \varepsilon\chi^e$ is also a subbase of r for some $\varepsilon > 0$ [9].

A maximal subbase of r is called is a *base* of r. Equivalently, a subbase x of r is a base of r if and only if $x(E) = r(E)$ [9, 21]. The set of all bases of r is also a polytope, which is called the *base polytope* of r. Its vertices can be characterized as follows [3]. For each ordering $\langle e_1, e_2, \cdots, e_n\rangle$ of E where $n = |E|$, let v be the vector defined by $v(e_1) = r(\{e_1\})$ and for each $1 < j \le n$,

$$v(e_j) = r(\{e_1, e_2, \cdots, e_j\}) - r(\{e_1, e_2, \cdots, e_{j-1}\}).$$

Then, v is a vertex of the base polytope of r. For this reason, v is referred to as the *extreme base of r induced* by the ordering $\langle e_1, e_2, \cdots, e_n\rangle$. The vertices of the base polytope of r are exactly the extreme bases of r induced by all possible orderings of E. A vector $x \in \mathbb{R}^n$ is a *superbase* of r if there exists a base y of r such that $y \le x$.

Suppose that each $e \in E$ has a positive weight $w(e)$. The total weight of a subbase x of r is $\sum_{e\in E} w(e)\,x(e)$. Then a subbase of r with maximum weight must be a base of r. So, the problem **MWSR** over the polymatroid of r is equivalent to finding a maximum-weighed base of r. By the polymatroid greedy method [3], the extreme base of r induced by the ordering of E in the decreasing order of weight is an optimal solution.

2.1 Restriction And Contraction

Let A be a subset of E. The *restriction* of r on A, denoted by r^A, is the set function on 2^A defined by $r^A(S) = r(S)$ for any $S \subseteq A$. The *contraction* of r on A, denoted by r_A, is the set function on $2^{E\setminus A}$ defined by $r_A(S) = r(S \cup A) - r(A)$ for any $S \subseteq E \setminus A$. Clearly, r^A and r_A are both polymatroidal rank functions. In addition, the following "composition" property [9] holds.

Lemma 2.1. *Suppose y is a subbase of r^A and z is a subbase of r_A. Then $x := y \oplus z$ is a subbase of r. Moreover, x is a base of r if and only if both y is a base of r^A and z is a base of r_A.*

Conversely, the following "decomposition" property [9] holds.

Lemma 2.2. *Let x be a subbase of r, and y and z be the restrictions of x to A and $E \setminus A$ respectively. Then, y is also a subbase of r^A. Furthermore, if y is a base of r^A, then z is a also subbase of r_A.*

In general, for any two disjoint subsets A and B of E, r_A^B denotes the set function on 2^B defined by $r_B^A(S) = r(S \cup A) - r(A)$ for any $S \subseteq B$. Then, r_B^A can be regarded as the restriction of r_B on A.

Finally, we remark that the generalized symmetry is preserved by restriction and contraction.

2.2 Truncations of Polymatroid

Consider a vector $b \in \mathbb{R}_+^E$. The *upper truncation* of r by b [9], denoted by r^b, is the rank function on 2^E defined by for any $S \subseteq E$,

$$r^b(S) = \min_{T\subseteq S}\{r(T) + b(S \setminus T)\}.$$

For a vector $x \in \mathbb{R}_+^E$, the following properties holds [9]:

- x is a subbase of r^b iff $x \le b$ and x is a subbase of r;
- x is a base of r^b iff x is a maximal subbase of r satisfying that $x \le b$;

In particular, when b is a subbase of r, b is unique base of r^b.

Consider a subbase a of r. The *lower truncation* of r by a [9], denoted by r_a, is a rank function on E defined by for any $S \subseteq E$,

$$r_a(S) = \min_{S\subseteq T\subseteq E}\{r(T) - a(T \setminus S)\}.$$

For a vector $x \in \mathbb{R}_+^E$, x is a base of r_a if and only if $x \ge a$ and x is a base of r.

Consider a subbase a of r and a vector $b \in \mathbb{R}_+^E$ with $b \ge a$. The *bi-truncation* of r by a from below and b from above [8, 9, 22], denoted by r_a^b, is the rank function on 2^E defined by for any $S \subseteq E$,

$$r_a^b(S) = \min_{T\subseteq E}\{r(T) + b(S \setminus T) - a(T \setminus S)\}.$$

r_a^b is also the upper truncation of r_a by b, and the lower truncation of r^b by a (observing that a is also a subbase of r^b). A vector $x \in \mathbb{R}_+^E$ is a base of r_a^b if and only if x is a maximal subbase of r satisfying that $a \le x \le b$. In particular, when b is a subbase of r, b is unique base of r_a^b. The maximal elements in the bi-truncated polymatroid

$$\left\{x \in \mathbb{R}_+^E : x(S) \le r(S), \forall S \subseteq E; a \le x \le b\right\}.$$

is exactly the base polytope of r_a^b. Then the problem **MWSR** over the bi-truncated polymatroid is equivalent to finding a maximum-weighed base of r_a^b, which can be solved by the polymatroid greedy method [3]. However, the evaluation of the bi-truncated rank is generally expensive as remarked at the beginning of this paper.

3 MINIMIZERS OF RANK-MODULUS DIFFERENCE

Consider a finite ground set E, and a polymatroid function r on 2^E, and a vector $u \in \mathbb{R}_+^E$. By treating u as a modular function on 2^E, the difference $r - u$ is referred to as a *rank-modulus difference*. The collection of minimizers of $r-u$ is closed under taking intersections and unions [21]. In particular, the union of all minimizers of $r - u$ is also a minimizer of $r - u$, and hence it is the unique *maximal minimizer* of $r - u$. When r is *strictly submodular*, i.e., for any two subsets A and B of E neither of which is a subset of each other,

$$r(A) + r(B) > r(A \cup B) + r(A \cap B),$$

then the collection of minimizers of $r - u$ is a chain [27, 28], i.e., for any two minimizers T_1 and T_2 of $r - u$, either $T_1 \subseteq T_2$ or $T_2 \subseteq T_1$.

Rank-modulus difference and its minimizers play essential roles in algorithmic applications including membership test of a polymatroid, computing the tight sets of a subbase, and evaluation of the truncated ranks. In Subsection 3.1, we present fundamental properties of minimizers of the difference between an arbitrary rank and a modulus. In Subsection 3.2, we develop efficient algorithms for computing minimizers of the difference between a generalized symmetric rank and a modulus.

3.1 Arbitrary Rank

In this subsection, we first provide the full characterization of minimizers of the difference between an arbitrary rank r and a non-negative modulus u.

Lemma 3.1. *Let T be a subset of E. The following statements are equivalent:*

1. *T is a minimizer of $r-u$.*
2. *For each base v of r^u, $v(T)=r(T)$ and $v(e)=u(e)$ for any $e\in E\setminus T$.*
3. *For some base v of r^u, $v(T)=r(T)$ and $v(e)=u(e)$ for any $e\in E\setminus T$.*

Proof. (1) $\Rightarrow$ (2). Suppose that T is a minimizer of $r-u$, and v is a base of r^u. Then, $v(E)=r^u(E)=r(T)+u(E\setminus T)$ and $v(T)\le r^u(T)=r(T)$. So,

$$\begin{aligned} v(E)&=v(T)+v(E\setminus T)\le r(T)+u(E\setminus T)\\ &=r^u(E)=v(E). \end{aligned}$$

Hence, we must have $v(T)=r(T)$ and $v(E\setminus T)=u(E\setminus T)$, which implies $v(e)=u(e)$ for any $e\in E\setminus T$.

(2) $\Rightarrow$ (3): Trivial.

(3) $\Rightarrow$ (1): Suppose that for some base v of r^u, $v(T)=r(T)$ and $v(e)=u(e)$ for any $e\in E\setminus T$. For any $S\subseteq T$,

$$\begin{aligned} r(S)+u(T)-u(S)&=r(S)+u(T\setminus S)\\ &\ge v(S)+v(T\setminus S)=v(T)=r(T), \end{aligned}$$

and hence $r(S)-u(S)\ge r(T)-u(T)$.

For any $S\supseteq T$,

$$\begin{aligned} r(S)\ge v(S)&=v(T)+v(S\setminus T)\\ &=r(T)+u(S\setminus T)\\ &=r(T)+u(S)-u(T) \end{aligned}$$

So, $r(S)-u(S)\ge r(T)-u(T)$.

Finally, for any $S\subseteq E$,

$$\begin{aligned} &[r(S)-u(S)]+[r(T)-u(T)]\\ &\ge[r(S\cap T)-u(S\cap T)]+[r(S\cup T)-u(S\cup T)]\\ &\ge[r(T)-u(T)]+[r(T)-u(T)] \end{aligned}$$

which implies $r(S)-u(S)\ge r(T)-u(T)$. Thus, T is a minimizer of $r-u$. □

Corollary 3.2. *The following statements hold:*

- *u is a subbase of r iff $\emptyset$ is a minimizer of $r-u$.*
- *u is a superbase of r iff E is a minimizer of $r-u$.*
- *u is a base of r iff both $\emptyset$ and E are minimizers of $r-u$.*

Next, we give the full characterization of the maximal minimizer of $r-u$.

Lemma 3.3. *Let T be a subset of E. The following statements are equivalent:*

1. *T is the maximal minimizer of $r-u$.*
2. *For each base v of r^u, T is the maximal v-tight set w.r.t. r.*
3. *For some base v of r^u, T is the maximal v-tight set w.r.t. r.*

Proof. (1) $\Rightarrow$ (2). By Lemma 3.1, T is a v-tight set, and $v(e)=u(e)$ for any $e\in E\setminus T$. For any *proper superset* S of T, by the maximality of T, $r(S)-u(S)>r(T)-u(T)$. So,

$$\begin{aligned} r(S)&>r(T)+u(S\setminus T)\\ &=v(T)+v(S\setminus T)\\ &=r(T)+v(S)-v(T)\\ &=v(S). \end{aligned}$$

Thus, S is not a v-tight set w.r.t. r.

(2) $\Rightarrow$ (3): Trivial.

(3) $\Rightarrow$ (1): We first claim that $v(e)=u(e)$ for any $e\in E\setminus T$. Assume to the contrary that $v(e)<u(e)$ for some $e\in E\setminus T$. Since T is the maximal v-tight set w.r.t. r, for some $\varepsilon>0$, $v+\varepsilon\chi^e$ is still a subbase of r and $v+\varepsilon\chi^e\le u$. Thus, $v+\varepsilon\chi^e$ is also a subbase of r^u. This contradicts to that v is a base of r^u. So, the claim holds.

By Lemma 3.1, T is a minimizer of $r-u$. For any *proper superset* S of T, by the maximality of T,

$$\begin{aligned} r(S)>v(S)&=v(T)+v(S\setminus T)\\ &=r(T)+u(S\setminus T)\\ &=r(T)+u(S)-u(T). \end{aligned}$$

So, $r(S)-u(S)>r(T)-u(T)$. Thus, S is not a minimizer of $r-u$. □

Next, we consider a special modulus arising from the definition of bi-truncated rank. Consider a subbase a of r and a vector $b\in\mathbb{R}_+^E$ with $b\ge a$. For any subset S of E, let $u_S\in\mathbb{R}_+^E$ be the vector defined by

$$u_S(e)=\begin{cases} b(e), & \text{if } e\in S;\\ a(e), & \text{if } e\in E\setminus S. \end{cases} \tag{2}$$

Then, for any $T\subseteq E$,

$$u_S(T)=u_S(T\cap S)+u_S(T\setminus S)=b(T\cap S)+a(T\setminus S).$$

So,

$$r_a^b(S)=b(S)+\min_{T\subseteq E}\{r(T)-u_S(T)\}.$$

If T is a minimizer of $r-u_S$ over all subsets of E, then

$$\begin{aligned} r_a^b(S)&=b(S)+r(T)-u_S(T)\\ &=r(T)+b(S\setminus T)-a(T\setminus S). \end{aligned}$$

Each minimizer of $r-u_S$ has the following property.

Lemma 3.4. *Suppose that S is a non-empty subset of E, and T is a minimizer of $r-u_S$. Then for each base v of r_a^b s.t. $v(S)=r_a^b(S)$, we have $v(T)=r(T)$ and $v(e)=u_S(e)$ for each $e\in(T\setminus S)\cup(S\setminus T)$.*

Proof. Since v is a base of r_a^b, we have $a \le v \le b$ and $v(T) \le r_a^b(T) \le r(T)$. As

$$v(S) + v(T \setminus S) = v(S \cup T) = v(T) + v(S \setminus T),$$

we have

$$\begin{aligned} v(S) &= v(T) + v(S \setminus T) - v(T \setminus S) \\ &\le r(T) + b(S \setminus T) - a(T \setminus S). \end{aligned}$$

On the other hand, since S is v-tight w.r.t. r_a^b and T is a minimizer of $r - u_S$, we have

$$v(S) = r_a^b(S) = r(T) + b(S \setminus T) - a(T \setminus S).$$

Thus, $v(T) = r(T)$, $v(S \setminus T) = b(S \setminus T)$, and $v(T \setminus S) = a(T \setminus S)$. So, for any $e \in S \setminus T$, $v(e) = b(e) = u_S(e)$; for any $e \in T \setminus S$, $v(e) = a(e) = u_S(e)$. □

Finally, a pair of minimizers of $r - u_S$ have the following property.

Lemma 3.5. *Suppose that S is a non-empty subset of E, and T_1 and T_2 are two minimizers of $r - u_S$ over all subsets of E with $T_1 \subseteq T_2$. Then for each base v of r_a^b s.t. $v(S) = r_a^b(S)$,*

- *the restriction of v on T_1 is a base of r^{T_1}, and $a(e) \le v(e) \le u_S(e)$ for each $e \in T_1$;*
- *the restriction of v on $T_2 \setminus T_1$ is a base of $r_{T_1}^{T_2 \setminus T_1}$, and $v(e) = u_S(e)$ for each $e \in T_2 \setminus T_1$;*
- *the restriction of v on $E \setminus T_2$ is a subbase of $r_{T_2}^{E \setminus T_2}$, and $u_S(e) \le v(e) \le b(e)$ for each $e \in E \setminus T_2$.*

Proof. Since v is a base of r_a^b, v is also a subbase of r. By Lemma 3.4, $v(T_2) = r(T_2)$; hence the restriction of v on T_2 is a base of r^{T_2}. By Lemma 2.2, the restriction of v on $E \setminus T_2$ is a subbase of $r_{T_2}^{E \setminus T_2}$. By Lemma 3.4, $v(T_1) = r(T_1)$; hence the restriction of v on T_1 is a base of r^{T_1}. Again, by Lemma 2.2, the restriction of v on $T_2 \setminus T_1$ is a base of $r_{T_1}^{T_2 \setminus T_1}$.

By Lemma 3.4, $v(e) = u_S(e)$ for each $e \in (S \setminus T_1) \cup (T_2 \setminus S)$. As

$$T_2 \setminus T_1 \subseteq (S \setminus T_1) \cup (T_2 \setminus S),$$

$v(e) = u_S(e)$ for each $e \in T_2 \setminus T_1$.

Consider any $e \in T_1$. Clearly, $v(e) \ge a(e)$. If $e \in S$, then $v(e) \le b(e) = u_S(e)$. If $e \notin S$, then $v(e) = u_S(e)$. Thus, $v(e) \le u_S(e)$ in any case.

Consider any $e \in E \setminus T_2$. Clearly, $v(e) \le b(e)$. If $e \notin S$, then $v(e) \ge a(e) = u_S(e)$. If $e \in S$, then $v(e) = u_S(e)$. Thus, $v(e) \ge u_S(e)$ in any case. □

The above lemma together with Corollary 3.2 implies the following corollary.

Corollary 3.6. *Suppose that S is a non-empty subset of E, and u_S is a base of r. Then u_S is the unique base of r_a^b s.t. $v(S) = r_a^b(S)$.*

3.2 Generalized Symmetric Rank

In this subsection, the rank r is assumed to be generalized symmetric as defined in equation (1). We show that the maximal minimizer of $r - u$ can be computed by a linear search.

A crucial property of the generalized symmetric rank r is a simple test of whether a given vector $u \in \mathbb{R}_+^E$ is a subbase of not. Specifically, suppose that $\langle e_1, e_2, \cdots, e_n \rangle$ is an ordering of E in which $u(e_j)/p(e_j)$ decreases with j. For each $0 \le i \le n$, let E_i be the subset $\{e_j : 0 \le j \le i\}$. Then, the following test of whether u is a subbase of r was given in [10].

Lemma 3.7. *u is a subbase of r if and only if $u(E_i) \le r(E_i)$ for each $1 \le i \le n$.*

Based upon the above test and the full characterization of the maximal minimizer of $r - u$ described in Lemma 3.3, we establish the following linear search for the maximal minimizer of $r - u$.

Theorem 3.8. *The maximal minimizer of $r - u$ is E_k for some $0 \le k \le n$.*

Proof. By Lemma 3.3, it is sufficient to show that for some base v of r^u, the maximal v-tight set w.r.t. r is E_k for some $0 \le k \le n$. We consider the following $v \in \mathbb{R}_+^n$ defined in [10]: For each $1 \le i \le n$,

$$v(e_i) = \min\{u(e_i), r(E_i) - v(E_{i-1})\}$$

Then, $v \le u$ and $v(E_i) \le r(E_i)$ for each $1 \le i \le n$. Let

$$k = \max\{0 \le i \le n : v(E_i) = r(E_i)\},$$

which is is well-defined as $v(E_0) = r(E_0) = 0$. We will show that v is a base of r^u and E_k is the maximal v-tight set w.r.t. r.

The property that v is a base of r^u was shown in [10]. For the sake of completeness, we include the detailed proof here. We first show that v is a subbase of r. By Lemma 3.7, it is sufficient to show that $v(e_i)/p(e_i)$ decreases with i. We prove that for each $1 \le i \le n-1$, $v(e_i)/p(e_i) \ge v(e_{i+1})/p(e_{i+1})$ in two cases.

Case 1: $v(e_i) = u(e_i)$. Then,

$$\frac{v(e_i)}{p(e_i)} = \frac{u(e_i)}{p(e_i)} \ge \frac{u(e_{i+1})}{p(e_{i+1})} \ge \frac{v(e_{i+1})}{p(e_{i+1})}.$$

Case 2: $v(e_i) < u(e_i)$. Then, $v(e_i) = r(E_i) - v(E_{i-1})$ and hence $v(E_i) = r(E_i)$. So,

$$\begin{aligned} \frac{v(e_i)}{p(e_i)} &= \frac{r(E_i) - v(E_{i-1})}{p(e_i)} \\ &\ge \frac{r(E_i) - r(E_{i-1})}{p(e_i)} \\ &\ge \frac{r(E_{i+1}) - r(E_i)}{p(e_{i+1})} \\ &= \frac{r(E_{i+1}) - v(E_i)}{p(e_{i+1})} \\ &= \frac{v(e_{i+1})}{p(e_{i+1})}. \end{aligned}$$

where the second inequality follows from that r is a generalized symmetric rank.

Since $v \le u$, v is also subbase of r^u. By the choice of k, $v(E_k) = r(E_k)$ and $v(e_j) = u(e_j)$ for each $j > k$. Thus, for any $\varepsilon > 0$ and any $1 \le j \le n$, $v + \varepsilon \chi^{e_j}$ is not a subbase of r^u. So, v is a base of r^u.

Next, we show that E_k is the maximal v-tight set w.r.t. r. This holds trivially if $k = n$; so we assume that $k < n$. Then $v(e_k)/p(e_k) > v(e_{k+1})/p(e_{k+1})$ as

$$\begin{aligned}\frac{v(e_k)}{p(e_k)} &= \frac{v(E_k) - v(E_{k-1})}{p(e_k)} \\ &= \frac{r(E_k) - v(E_{k-1})}{p(e_k)} \\ &\geq \frac{r(E_k) - r(E_{k-1})}{p(e_k)} \\ &\geq \frac{r(E_{k+1}) - r(E_k)}{p(e_{k+1})} \\ &= \frac{r(E_{k+1}) - v(E_k)}{p(e_{k+1})} \\ &> \frac{v(E_{k+1}) - v(E_k)}{p(e_{k+1})} \\ &= \frac{v(e_{k+1})}{p(e_{k+1})},\end{aligned}$$

where the strict inequality follows from that $v(E_{k+1}) < r(E_{k+1})$. Let

$$\varepsilon = \min\left\{\frac{v(e_k)}{p(e_k)} - \frac{v(e_{k+1})}{p(e_{k+1})}, \min_{k+1\leq j\leq n} \frac{r(E_j) - v(E_j)}{p(E_j) - p(E_k)}\right\}.$$

Then $\varepsilon > 0$. Let $\overline{v}$ be such that $\overline{v}(e_j) = v(e_j)$ if $1 \leq j \leq k$ and $\overline{v}(e_j) = v(e_j) + \varepsilon p(e_j)$ for each $k+1 \leq j \leq n$. We claim that $\overline{v}$ is a subbase of r. Such claim implies that any strict superset of E_k is not v-tight w.r.t. r, and hence E_k is the maximal v-tight set w.r.t. r. We prove the claim below using Lemma 3.7.

First, we show that $\overline{v}(e_j)/p(e_j) \geq v(e_{j+1})/p(e_{j+1})$ for each $1 \leq j < n$ in three cases; in other words, $\overline{v}(e_j)/p(e_j)$ decreases with j.

Case 1: $1 \leq j < k$. Then,

$$\frac{\overline{v}(e_j)}{p(e_j)} = \frac{v(e_j)}{p(e_j)} \geq \frac{v(e_{j+1})}{p(e_{j+1})} = \frac{\overline{v}(e_{j+1})}{p(e_{j+1})}.$$

Case 2: $j = k$. Then,

$$\frac{\overline{v}(e_k)}{p(e_k)} = \frac{v(e_k)}{p(e_k)} \geq \frac{v(e_{k+1})}{p(e_{k+1})} + \varepsilon = \frac{\overline{v}(e_{k+1})}{p(e_{k+1})}.$$

Case 3: $k+1 \leq j < n$,

$$\frac{\overline{v}(e_j)}{p(e_j)} = \frac{v(e_j)}{p(e_j)} + \varepsilon \geq \frac{v(e_{j+1})}{p(e_{j+1})} + \varepsilon = \frac{\overline{v}(e_{j+1})}{p(e_{j+1})}.$$

Secondly, we show that $\overline{v}(E_j) \leq r(E_j)$ for each $1 \leq j \leq n$ in two cases.

Case 1: $1 \leq j \leq k$. Then,

$$\overline{v}(E_j) = v(E_j) \leq r(E_j);$$

Case 2: $k+1 \leq j \leq n$,

$$\begin{aligned}\overline{v}(E_j) &= v(E_j) + \varepsilon(p(E_j) - p(E_k)) \\ &\leq v(E_j) + (r(E_j) - v(E_j)) \\ &= r(E_j).\end{aligned}$$

By Lemma 3.7, $\overline{v}$ is a subbase of r. This completes the proof of the theorem. □

Suppose that the ordering $\langle e_1, e_2, \cdots, e_n\rangle$ is represented by a list L. Among all the prefix sublists of L, the shortest (respectively, longest) one which is a minimizer of $r - u$ is referred to as the shortest (respectively, longest) prefix minimizer of $r - u$ in L. The above theorem says that the longest prefix minimizer of $r-u$ in L is also the maximal minimizer of $r-u$. We describe a linear-time procedure **MinDiff**, which takes L as input and produces minimum value δ of $r - u$, and the length k_1 (respectively, k_2) shortest (respectively, longest) prefix minimizer of $r - u$ in L. The vectors p and u are represented by arrays globally accessible by the procedure **MinDiff**. During the process of linear search along the list L, three working variables i, q, and σ are used to store $|T|$, $p(T)$, and $u(T)$ of the present prefix sublist T of L respectively, and maintain the following *invariant* property: among all the prefix sublists of length at most i, the prefix sublist of length k_1 (respectively, k_2) is the shortest (respectively, longest) one achieving the minimum value δ of $r - u$.

The main body of the procedure is outlined in Table 1. Initially, all the three output variables and the three working variables are initialized to 0. Then, the procedure scans the list L from the beginning to the end. When an element $e \in L$ is scanned, the three working variables are updated accordingly, and subsequently the three output variables are updated as follows. Note that $\phi(q) - \sigma$ is the value of $r - u$ at the present prefix list of length i, and δ is the minimum value of $r - u$ at all all shorter prefix sublists. Thus, if $\phi(q) - \sigma < \delta$, then δ is replaced by $\phi(q) - \sigma$ and both k_1 and k_2 are replaced by i; if $\phi(q) - \sigma = \delta$, only k_2 is replaced by i; if $\phi(q) - \sigma > \delta$, then all the three output variables remain unchanged. After completing the scanning of the entire list L, the values of the three output variables are returned. As each scanning step (i.e., each iteration in the for-loop) takes constant time, the procedure **MinDiff** has linear running time.

```
MinDiff(L):
δ ← 0, q ← 0, σ ← 0;
k1 ← 0, k2 ← 0, i ← 0;
for each e ∈ L sequentially do
    i ← i + 1;
    q ← q + p(e);
    σ ← σ + u(e);
    if φ(q) − σ < δ then
        δ ← φ(q) − σ, k1 ← i, k2 ← i;
    if φ(q) − σ = δ then k2 ← i;
return δ, k1, and k2
```

Table 1: Outline of the procedure MinDiff.

Finally, we introduce a slight extension to the procedure **MinDiff**, which will be utilized in the latter part of this paper. Let q_0 be a positive parameter and A is a non-empty subset of E. Then the function $\widetilde{r}$ defined by

$$\widetilde{r}(S) = \phi(q_0 + p(S)) - \phi(q_0)$$

for each $S \subseteq A$ is also a generalized symmetric rank function on the subsets of A. Denote the restriction of u on A by u^A. The procedure **MinDiff-2** includes q_0 in the input in addition to L, and produces the minimum value δ of $\widetilde{r} - u^A$ and the length k_1 (respectively, k_2) shortest (respectively, longest) prefix minimizer of $\widetilde{r} - u$ in L. It makes the following two simple modifications from **MinDiff**,

- In the initialization, the two statements $\delta \leftarrow 0, q \leftarrow 0$ are replaced with $\delta \leftarrow \phi(q_0), q \leftarrow q_0$
- A statement $\delta \leftarrow \delta - \phi(q_0)$ is added right before the return statement.

Thus, its running time is also linear.

4 A GREEDY ALGORITHM

In this section, we present a greedy algorithm for linear optimization over a bi-truncated generalized symmetric polymatroid.

Let

$$E^{=} = \{e \in E : a(e) = b(e)\},$$
$$E^{<} = \{e \in E : a(e) < b(e)\}.$$

Then, for any base v of r_a^b and any $e \in E^{=}$, $v(e) = a(e) = b(e)$. This means that the users in $E^{=}$ are always allocated fixed rates in any base of r_a^b, regardless of their weights (rewards). Therefore, we can change their weights to an arbitrary positive value which is strictly smaller than $\min_{e \in E^<} w(e)$. Such change is optimality-preserving. Thus, we assume $\min_{e \in E^<} w(e) > \max_{e \in E^=} w(e)$.

Suppose that $e_1, e_2, \cdots, e_k$ is an ordering of $E^<$ in which $w(e_j)$ decreases with j. Let $E_0 = \emptyset$, and $E_i = \{e_1, e_2, \cdots, e_i\}$ for $1 \le i \le k$. Let $x \in \mathbb{R}_+^E$ be the vector defined by $x(e_j) = r_a^b(E_j) - r_a^b(E_{j-1})$ for $1 \le j \le k$, and $x(e) = a(e)$ for each $e \in E^=$. By the polymatroid greedy method [3], x is an optimal solution. We derive an alternate expression of $x(e_j)$ for $1 \le j \le k$ below. Let u_{E_j} be as defined in equation (2), and denote

$$\delta_j = \min_{T \subseteq E} \left[r(T) - u_{E_j}(T) \right].$$

In addition, denote $\delta_0 = 0$. Then, $r_a^b(E_j) = b(E_j) + \delta_j$. So,

$$x(e_j) = \left[b(E_j) + \delta_j\right] - \left[b(E_{j-1}) + \delta_{j-1}\right]$$
$$= b(e_j) + \delta_j - \delta_{j-1}.$$

Next, we describe an efficient implementation of the above design when r is a generalized symmetric rank as defined in equation (1). The input vectors a, b, p, w are represented by arrays of size n, so is the output vector x. We also use an array u to represent u_{E_j}, which is initialized to a, and be updated iteratively. The ordering of $E^<$ in the decreasing order of weight is represented by a list L^w. The ordering of E in the decreasing order of $u(e)/p(e)$ is represented by another list L^{up}. In addition, we use two scalar variables δ and δ' to represent δ_j and δ_{j-1} respectively.

Initialization: For each $e \in E^=$, $x(e)$ is set of $a(e)$. u is set to a; δ' is set to 0. Then, L^{up} and L^w are computed by the merge-sorting algorithm. The total running time is $O(n \log n)$.

Greedy computation of $x(e)$ for each $e \in E^<$: The algorithm scans the list L^w sequentially. In the iteration in which $e \in L^w$ is scanned, $x(e)$ is computed as follows.

- $u(e)$ is reset to $b(e)$, and the list L^{up} is updated accordingly in linear time.
- The minimum δ of $r - u$ over all prefix sublists of L^{up} is computed by applying the algorithm **MinDiff**.
- $x(e)$ is set to $b(e) + \delta - \delta'$, and δ' is reset to δ.

Thus, each iteration takes $O(n)$ time. As there are $|E^<| = O(n)$ iterations, the total running time is $O(n^2)$.

In summary, we have the following theorem.

Theorem 4.1. *The algorithm Greedy-GS produces an optimal solution in quadratic time.*

5 A DIVIDE-AND-CONQUER ALGORITHM

In this section, we develop a divide-and-conquer algorithm for linear optimization over a bi-truncated generalized symmetric polymatroid with linearithmic time complexity. Underlying this algorithm is a refined decomposing method for linear optimization over a bi-truncated polymatroid, which is presented in Subsection 5.1. A divide-and-conquer implementation of this method for linear optimization over a bi-truncated generalized symmetric polymatroid is presented in Subsection 5.2.

5.1 A Refined Decomposing Method

Suppose that r is a polymatroidal rank on 2^E, and each user $e \in E$ has a positive weight $w(e)$. We define an extended problem **MWSR**(A, A_0, a, b) where

- A is a subset of E, and is referred as the ground set,
- A_0 is a subset of E disjoint from A,
- a is a subbase of $r_{A_0}^A$ representing the lower bound, and
- b is a vector in $\mathbb{R}_+^A$ satisfying that $b \ge a$, representing upper bound.

For simplicity of treatment, the definition allows for the *degenerate* instance that $A = \emptyset$, for which *null* is the only feasible solution with 0 total weight. For the non-degenerate instance, a feasible solution to this extended problem is a subbases x of $r_{A_0}^A$ satisfying that $a \le x \le b$, and the objective of this extended problem is to compute a feasible solution x maximizing $\sum_{e \in A} w(e) x(e)$. Then, the original problem can be represented by **MWSR**$(E, \emptyset, a, b)$.

Consider a non-degenerate problem **MWSR**(A, A_0, a, b). Define

$$A^{=} := \{e \in A : a(e) = b(e)\},$$
$$A^{<} := \{e \in A : a(e) < b(e)\}.$$

Clearly, for each feasible solution x, we must have $x(e) = b(e)$ for any $e \in A^=$. Thus, users in $A^=$ are said to be *fixed*, and users in $A^<$ is said to be *free*. In the *trivial* instance that $A^< = \emptyset$, $b = a$ is the unique optimal solution. So, we assume that $A^< \ne \emptyset$. In general, $|A^<|$ is a key measure of the effort required to solve the problem **MWSR**(A, A_0, a, b). There are two *simple* cases of the instance in which the optimal solution is unique and directly computable:

- Case 1: b is a subbase of $r_{A_0}^A$. In this case, b is also the unique optimal solution.
- Case 2: $A^<$ is a singleton $\{e\}$. In this case, the unique optimal solution x is identical to b except the value of $x(e)$, which is equal to

$$b(e) + \min_{T \subseteq A} \left(r_{A_0}(T) - b(T)\right).$$

Both cases can tested or solved by computing the minimum value of $r_{A_0} - b$. For the instance which is of neither of the above cases (i.e., $\left|A^<\right| \geq 2$ and b is not a subbase), we derive a recursive relation subsequently.

Let S be a subset of $A^<$ satisfying that

$$|S| = \left\lceil \left|A^<\right| /2 \right\rceil$$

and

$$\min_{e \in S} w(e) \geq \max_{e \in A^< \setminus S} w(e).$$

Then, neither S nor $A^< \setminus S$ is empty. Let $u \in \mathbb{R}_+^A$ be the vector

$$u(e) = \begin{cases} b(e), & \text{if } e \in S; \\ a(e), & \text{if } e \in A \setminus S. \end{cases} \tag{3}$$

The following lemma is an immediate consequence of the Corollary 3.6 and the polymatroid greedy method.

Lemma 5.1. *If u is a base of $r_{A_0}^A$, then u is an optimal solution to the problem* $\textbf{MWSR}(A, A_0, a, b)$.

In general, suppose that A_1 and A_2 be two minimizers of $r_{A_0}^A - u$ with $A_1 \subseteq A_2$. Denote $\overline{A_2} = A \setminus A_2$. Let u_1 (respectively, u_2, u_3) be the restriction of u on A_1 (respectively, $\overline{A_2}$, $A_2 \setminus A_1$), a_1 be the restriction of a on A_1. Then, $a_1 \leq u_1$ and a_1 is a subbase of $r_{A_0}^{A_1}$; hence the problem $\textbf{MWSR}(A_1, A_0, a_1, u_1)$ is well-defined. Similarly, $u_2 \leq b_2$ and u_2 is a subbase of $r_{A_0 \cup A_2}^{\overline{A_2}}$ by Lemma 3.5; hence the problem $\textbf{MWSR}\left(\overline{A_2}, A_0 \cup A_2, u_2, b_2\right)$ is also well-defined. These two problems are referred as the "child" problems of the "parent" problem $\textbf{MWSR}(A, A_0, a, b)$. They have the following composition relations.

Theorem 5.2. *Consider any optimal solution x_1 to the problem $\textbf{MWSR}(A_1, A_0, a_1, u_1)$ and any optimal solution x_2 to the problem $\textbf{MWSR}\left(\overline{A_2}, A_0 \cup A_2, u_2, b_2\right)$. Let $x = x_1 \oplus x_2 \oplus u_3$. Then, x is an optimal solution to the problem $\textbf{MWSR}(A, A_0, a, b)$.*

Proof. Since $a \leq u \leq b$, it holds that $a \leq x \leq b$. Note that x_1 is a subbase of $r_{A_0}^{A_1}$, u_3 is a subbase of $r_{A_0 \cup A_1}^{A_2 \setminus A_1}$ by Lemma 3.5, and x_2 is a subbase of $r_{A_0 \cup A_2}^{\overline{A_2}}$. By Lemma 2.1, x is a subbase of $r_{A_0}^A$. Thus, the feasibility of x follows. Let v be the optimal solution to the problem $\textbf{MWSR}(A, A_0, a, b)$ produced by the polymatroid greedy method, as described in previous section, in a weight-decreasing order of E in which S is a prefix. Then, S is v-tight w.r.t. the bi-truncated rank. Let v_1 (respectively, v_2, v_3) be the restriction of v on A_1 (respectively, $\overline{A_2}$, $A_2 \setminus A_1$). By Lemma 3.5, $v_3 = u_3$, v_1 is an optimal solution to the problem $\textbf{MWSR}(A_1, A_0, a_1, u_1)$, and v_2 is an optimal solution to the problem $\textbf{MWSR}\left(\overline{A_2}, A_0 \cup A_2, u_2, b_2\right)$. By the optimalities of x_1 and x_2, we have

$$\sum_{e \in A_1} w(e) x_1(e) \geq \sum_{e \in A_1} w(e) v_1(e),$$
$$\sum_{e \in \overline{A_2}} w(e) x_2(e) \geq \sum_{e \in \overline{A_2}} w(e) v_2(e).$$

Since

$$\sum_{e \in A_2 \setminus A_1} w(e) u_3(e) = \sum_{e \in A_2 \setminus A_1} w(e) v_3(e).$$

The above three inequalities imply that

$$\sum_{e \in A} w(e) x(e) \geq \sum_{e \in A} w(e) v(e).$$

So, the optimality of x follows. □

The above theorem implies that the users in $A_2 \setminus A_1$ can be fixed to a rate allocation u_3. Now, we identify the set of free users in each of the two new problems. Let

$$A_1^< := \{e \in A_1 : a_1(e) < u_1(e)\},$$
$$\overline{A_2}^< := \left\{e \in \overline{A_2} : u_2(e) < b_1(e)\right\}.$$

Then,

$$A_1^< = \{e \in A_1 : a(e) < u(e)\} = A_1 \cap S,$$
$$\overline{A_2}^< = \left\{e \in \overline{A_2} : u(e) < b(e)\right\} = A^< \setminus (A_2 \cup S).$$

So,

$$\left|A_1^<\right| \leq |S| = \left\lceil \left|A^<\right| /2 \right\rceil,$$
$$\left|A_1^<\right| \leq \left|A^<\right| - |S| \leq \left\lfloor \left|A^<\right| /2 \right\rfloor.$$

Thus, the number of free users in each of the two new problems is at most halved. In the trivial case that $A_1^<$ is empty, the problem $\textbf{MWSR}(A_1, A_0, a_1, u_1)$ has a unique feasible solution u_1. Similarly, in trivial case that $\overline{A_2}^<$ is empty, the problem $\textbf{MWSR}\left(\overline{A_2}, A_0 \cup A_2, u_2, b_2\right)$ has a unique feasible solution $b_2 = u_2$.

We remark that the decomposition method [24–26] is a special case of the above decomposition method in which $A_1 = A_2$. The advantage of employing dual minimizers is obvious: more users may become fixed. In particular, the entire subset $A_2 \setminus A_1$ becomes fixed. The child problems may have smaller size. For certain rank r such as the generalized symmetric rank, the dual minimizers can be computed at no additional cost.

5.2 A Linearithmic Implementation

In this subsection, we assume that the rank r is a generalized symmetric as defined in equation (1), and present an implementation of the generalized decomposition method with linearithmic time complexity.

We begin with the introduction of the data structures to be used. First, there are *seven* arrays indexed by E which are globally accessible to all the procedure calls. Among them, two static arrays p and w are used for storing the vector p in the definition of r and the weight vector w respectively. Two arrays a and b are used for storing and updating the lower bound a and upper bound b respectively. At the end of the algorithm, b is returned as the final solution. With such trick, many duplication operations, parameter passing, as well as certain degeneracy can be avoided. The array u is to store the vector u defined in equation (3). The array col is to indicate the membership of S, $A^< \setminus S$, and $A^=$ in a problem with ground set A by

$$\mathrm{col}(e) = \begin{cases} 0, & \text{if } e \in A^=; \\ 1, & \text{if } e \in S; \\ 2, & \text{if } e \in A^< \setminus S. \end{cases}$$

The array *class* is to indicate the membership of A_1, $A_2 \setminus A_1$, and $\overline{A_2}$ in a problem with ground set A by

$$class(e) = \begin{cases} 0, & \text{if } e \in A_2 \setminus A_1; \\ 1, & \text{if } e \in A_1; \\ 2, & \text{if } e \in \overline{A_2}. \end{cases}$$

Second, there are three lists used locally by each problem with ground set A: a list $L^{a/p}$ (respectively, $L^{b/p}$) of A in the decreasing order of $a(e)/p(e)$ (respectively, $b(e)/p(e)$), and a list L^w of $A^<$ in the decreasing order of $w(e)$.

With the above data structures, we describe a proper representation of a problem **MWSR**(A, A_0, a, b). First, a and b can be skipped as they are globally accessible and updated throughout the execution of algorithm. Second, A_0 can be replaced by a single parameter $q_0 = p(A_0)$ as

$$r^A_{A_0}(S) = \phi(q_0 + p(S)) - \phi(q_0)$$

for each $S \subseteq A$. Third, for the achieving linear-time computation, A is represented by the three lists $L^{a/p}$, $L^{b/p}$, and L^w for A. Thus, the problem **MWSR**(A, A_0, a, b) is represented by the tuple $\left(L^{a/p}, L^{b/p}, L^w, q_0\right)$. Accordingly, we name the recursive procedure to solve the problem **MWSR**(A, A_0, a, b) by **DC**$\left(L^{a/p}, L^{b/p}, L^w, q_0\right)$.

The main algorithm for the problem **MWSR**$(E, \emptyset, a, b)$ is named **DC-Main**. It simply prepares the seven arrays p, w, a, b, u, col, *class* and the three lists $L^{a/p}$, $L^{b/p}$, L^w of E, calls the procedure **DC**$\left(L^{a/p}, L^{b/p}, L^w, 0\right)$, and finally returns b. The preparation takes linearithmic time. In the remaining of this subsection, we elaborate on the design of the procedure **DC**$\left(L^{a/p}, L^{b/p}, L^w, q_0\right)$ for the problem **MWSR**(A, A_0, a, b). Note that the **Combination** part is not needed.

Direct computation: First, the minimum δ of $r^A_{A_0} - b^A$, where b^A denotes the restriction of b on A, is computed along the list $L^{b/p}$ with parameter q_0 by applying the procedure **MinDiff-2**. If $\delta = 0$, then b is a subbase of $r^A_{A_0}$, and the procedure stops and returns. Then the length m of L^w, which is $\left|A^<\right|$, is computed. If $m = 1$, then for the unique $e \in L^w$, $b(e)$ is reset to $b(e) + \delta$, and the procedure stops and returns. Clearly, this part takes linear time.

Division: This part is logically split into two steps. Step 1 essentially computes S, u, and the two minimizers A_1 and A_2. First, the two arrays col and u are updated as follows:

- Along the list $L^{b/p}$, $\mathrm{col}(e)$ is initialized to 0 and $u(e)$ is initialized to $a(e)$ for each $e \in A$.
- Along the first half of the list L^w (which is chosen as S), $\mathrm{col}(e)$ is reset to 0 and $u(e)$ is set to $b(e)$ for each $e \in S$.
- Along the second half of the list L^w (which is chosen as $A^< \setminus S$), $\mathrm{col}(e)$ is reset to 2 for each $e \in A^< \setminus S$.

Then, a list $L^{u/p}$ of A in the decreasing order of $u(e)/p(e)$ is created as follows.

- The sublist $\widetilde{L}^{b/p}$ of $L^{b/p}$ consisting of the users in S is extracted from $L^{b/p}$. Note that a user $e \in S$ if and only if $\mathrm{col}(e) = 1$.
- The sublist $\widetilde{L}^{a/p}$ of $L^{a/p}$ consisting of the users not in S is extracted from $L^{a/p}$.
- The two lists $\widetilde{L}^{b/p}$ and $\widetilde{L}^{a/p}$ are then merged into the list $L^{u/p}$, as in the classic Merge-Sorting algorithm, in linear time.

Finally, the length k_1 (respectively, k_2) of the shortest (respectively, longest) prefix minimizer A_1 (respectively, A_2) in $L^{u/p}$ of $r^A_{A_0} - u^A$ is computed by applying the procedure **MinDiff-2**$\left(L^{u/p}, q_0\right)$. If $k_1 = 0$ and $k_2 = |A|$, then b^A is reset to u^A, and and the procedure stops and returns. Clearly, this step takes linear time.

Step 2 of the **Division** part resets a and b and builds the inputs to the two recursive procedure calls: $\left(L_1^{a/p}, L_1^{b/p}, L_1^w, q_0\right)$ associated with A_1, and $\left(L_2^{a/p}, L_2^{b/p}, L_2^w, q_0 + q_2\right)$ associated with $\overline{A_2}$ where $q_2 = p(A_2)$. Note that $L_1^{b/p}$ consists of the first k_1 elements in $L^{u/p}$, and $L_2^{a/p}$ consists of the last $|A| - k_2$ elements in $L^{u/p}$. Thus, by a single scanning of $L^{u/p}$, the two lists $L_1^{b/p}$ and $L_2^{a/p}$, the three arrays *class*, a and b, and the scalar variable q_2 can be computed and updated. Then, assisted by the array *class*, the list $L_1^{a/p}$ (respectively, $L_2^{b/p}$) is extracted from $L^{a/p}$ (respectively, $L^{b/p}$), and the two lists L_1^w and L_2^w are extracted from the list L^w. Clearly, this step also takes linear time.

Conquer: This part is trivial. It simply makes a call to

$$\mathbf{DC}\left(L_1^{a/p}, L_1^{b/p}, L_1^w, q_0\right)$$

if $L_1^w \neq \emptyset$, a call to

$$\mathbf{DC}\left(L_2^{a/p}, L_2^{b/p}, L_2^w, q_0 + q_2\right),$$

if $L_2^w \neq \emptyset$, and then returns.

All the problems generated in the recursive computation can be organized as a rooted binary tree according the parent-child relations, known as the recurrence tree. The depth of the recurrence tree is known as the recurrence depth. Since the number of free users in a child problem is at most half of that of its parent problem, the recurrence depth is $O(\log n)$. As the **Direct computation** part and **Division** part takes linear time, all the problems at the same level (or depth) have $O(n)$ time complexity since they have disjoint ground sets. So, the overall running time of the algorithm **DC-Main** is $O(n \log n)$. In summary, we have the following result.

Theorem 5.3. *The algorithm **DC-Main** produces an optimal solution in linearithmic time.*

6 CONCLUSION

For the uplink Gaussian channel with box constraints as well as some other communication systems, the feasible rate-capacity region can be represented by a bi-truncated generalized symmetric polymatroid. The best-known algorithm for maximizing weighted sum rate over such rate-capacity region has time complexity $O\left(n^5 \ln^{O(1)} n\right)$. In this paper, we develop a quadratic-time greedy algorithm and a linearithmic-time divide-and-conquer algorithm. Underlying the divide-and-conquer algorithm is a refined decomposing method for linear optimization over a bi-truncated polymatroid. A key ingredient of both algorithms is a linear-time algorithm for minimizing the difference between a generalized symmetric rank function and a modular function after a linearithmic-time ordering.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of P. R. China under grants 61502505, 61529202, 61572131, and 61872081, and by SUNY Oneonta Faculty Development Funds.

REFERENCES

[1] J. G. Andrews, Interference cancellation for cellular systems: a contemporary overview, *IEEE Trans. Wireless Commun.* 12(2): 19–29, 2005.
[2] T. M. Cover and J. A. Thomas, *Elements of information theory.* John Wiley & Sons, 2012.
[3] J. Edmonds, Submodular functions, matroids and certain polyhedra. In *Combinatorial structures and their applications*, eds. R. Guy, H. Hanani, N. Sauer and J. Schonheim, Pages 69-87, 1970.
[4] W. H. Cunningham, On submodular function minimization. *Combinatorica* 5(3): 185–192, 1985.
[5] A. Federgruen and H. Groenevelt, The greedy procedure for resource allocation problems: Necessary and sufficient conditions for optimality. *Operations Research* 34: 909-918, 1986.
[6] A. Federgruen and H. Groenevelt, Characterization and optimization of achievable performance in general queueing systems, *Operations Research* 36(5): 733 - 741, 1988.
[7] L. Fleischer and S. Iwata, A push-relabel framework for submodular function minimization and applications to parametric optimization. *Discrete Applied Mathematics* 131(2): 311–322, 2003.
[8] A. Frank and E. Tardos, Generalized polymatroids and submodular flows, *Mathematical Programming* 42: 489–563, 1988.
[9] S. Fujishige, *Submodular Functions and Optimization*, 2nd ed. Annals of Discrete Mathematics vol. 58, Elsevier, Amsterdam, 2005.
[10] H. Groenevelt, Two algorithms for maximizing a separable concave function over a polymatroid feasible region. *European Journal of Operational Research* 54(2): 227–236, 1991.
[11] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* 1: 169–197, 1981.
[12] S. Iwata, Submodular function minimization. *Mathematical Programming* 112(1): 45–64, 2008.
[13] S. Iwata and L. Fleischer and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Proc. of 32nd Symposium on Theory of Computing* (STOC'00), pp. 97-106, 2000.
[14] S. Iwata and J. B. Orlin, A simple combinatorial algorithm for submodular function minimization. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA'09), pages 1230–1237, 2009.
[15] E. L. Lawler and C. U. Martel, Computing maximal "polymatroidal" network flows. *Mathematical of Operation Research* 7: 334-347, 1982.
[16] Y. T. Lee, A. Sidford, and S. C. Wong, A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science* (FOCS'15), pages 1049–1065, 2015.
[17] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan, and L. Hanzo, Nonorthogonal Multiple Access for 5G and Beyond, *Proceedings of the IEEE* 105(12): 2347–2381, 2017.
[18] J.B. Orlin, A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming* 118(2): 237–251, 2009.
[19] P. Patel and J. Holtzman, Analysis of a simple successive interference cancellation scheme in a DS/CDMA system, *IEEE J. Sel. Areas Commun.* 12(5): 796–807, 1994.
[20] A. Schrijver, A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory*, Series B, 80(2): 346–355, 2000.
[21] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*, Springer, 2003.
[22] N. V. Shakhlevich, A. Shioura, and V. A. Strusevich, Single machine scheduling with controllable processing times by submodular optimization. *Int. J. Found. Comput. Sci.* 20: 247–269, 2009.
[23] J. G. Shanthikumar and D. D. Yao, Multiclass queueing systems: polymatroidal structure and optimal scheduling control, *Operations Research* 40(S2): 293-299, 1992.
[24] A. Shioura, N. V. Shakhlevich, and V. A. Strusevic, Decomposition algorithms for submodular optimization with applications to parallel machine scheduling with controllable processing times. *Math. Program., Ser. A* 153: 495–534, 2015.
[25] A. Shioura, N. V. Shakhlevich, and V. A. Strusevich, Application of submodular optimization to single machine scheduling with controllable processing times subject to release dates and deadlines, *INFORMS Journal on Computing* 28: 148-161, 2016.
[26] A. Shioura, N. V. Shakhlevich, and V. A. Strusevich, Preemptive models of scheduling with controllable processing times and of scheduling with imprecise computation: A review of solution approaches, European Journal of Operational Research 266(3): 795-818, 2018.
[27] D. M. Topkis. Minimizing a submodular function on a lattice. *Operations Research* 26(2): 305– 321, 1978.
[28] D. M. Topkis. *Supermodularity and complementarity*, Princeton University Press, 2011.
[29] D. Tse and S. V. Hanly. Multiaccess fading channels - part I: polymatroid structure, optimal resource allocation and throughput capacities. *IEEE Transactions on Information Theory* 44(7): 2796–2994, 1998.
[30] D. Tse and P. Viswanath, *Fundamentals of wireless communication.* Cambridge university press, 2005.