Approximating Minimum Linear Ordering Problems

Satoru Iwata[1], Prasad Tetali[2,3], and Pushkar Tripathi[3]

[1] Research Institute for Mathematical Sciences, Kyoto University
[2] School of Mathematics, Georgia Institute of Technology ⋆⋆
[3] School of Computer Science, Georgia Institute of Technology

Abstract. This paper addresses the Minimum Linear Ordering Problem (MLOP): Given a nonnegative set function f on a finite set V, find a linear ordering on V such that the sum of the function values for all the suffixes is minimized. This problem generalizes well-known problems such as the Minimum Linear Arrangement, Min Sum Set Cover, Minimum Latency Set Cover, and Multiple Intents Ranking. Extending a result of Feige, Lovász, and Tetali (2004) on Min Sum Set Cover, we show that the greedy algorithm provides a factor 4 approximate optimal solution when the cost function f is supermodular. We also present a factor 2 rounding algorithm for MLOP with a monotone submodular cost function, using the convexity of the Lovász extension. These are among very few constant factor approximation algorithms for NP-hard minimization problems formulated in terms of submodular/supermodular functions. In contrast, when f is a symmetric submodular function, the problem has an information theoretic lower bound of 2 on the approximability.
Feige, Lovász, and Tetali (2004) also devised a factor 2 LP-rounding algorithm for the Min Sum Vertex Cover. In this paper, we present an improved approximation algorithm with ratio 1.79. The algorithm performs multi-stage randomized rounding based on the same LP relaxation, which provides an answer to their open question on the integrality gap.

1 Introduction

In this paper we introduce the Minimum Linear Ordering Problem (MLOP), which generalizes several known problems such as the Minimum Linear Arrangement (MLA) and Min Sum Set Cover problem (MSSC) problems. Each of these problems has been extensively studied in isolation. In this paper we initiate a systematic study of these problems under the general umbrella of submodular and supermodular set functions.

An instance of the MLOP consists of a ground set V of cardinality n and a cost function $f : 2^V \to \mathbb{R}_+$. The objective is to find a linear ordering (bijection) $\sigma : V \to \{1, \ldots, n\}$ such that $\sum_i f(S_i)$ is minimized, where for any linear ordering σ of V, we define $S_i = S_i(\sigma) = \{v \mid \sigma(v) \geq i\}$. We consider three broad classes of cost functions f: supermodular, monotone submodular, and symmetric submodular functions.

⋆⋆ Supported in part by NSF DMS-1101447 and CCR 0910584

1.1 Results and Techniques

For the case when the cost function is a *supermodular* function we establish the following theorem.

Theorem 1. *For any instance of MLOP with a supermodular cost function, the greedy algorithm yields a factor* 4 *approximation to the optimal linear ordering.*

The proof is based on a scaling technique that is similar to dual fitting. We use a histogram to represent any solution to the problem and show that the histogram corresponding to the greedy solution when scaled appropriately fits within the histogram for the optimal solution.

We also consider a special case of this problem, the min-sum vertex cover problem. In this problem we are given a graph $G(V, E)$ and the objective is to arrange the vertices of G in a linear ordering σ, such that $\sum_{(u,v)\in E} \min\{\sigma(u), \sigma(v)\}$ is minimized. We achieve the following result with regard to this problem.

Theorem 2. *There exists a Las Vegas algorithm that approximates the min-sum vertex cover to within a factor of* 1.79*.*

This algorithm is based on a multi-stage rounding of the natural linear programming relaxation. In our rounding technique, we randomly and independently round each of the variables in the optimal linear programming solution. However we do not perform this rounding simultaneously for all variables. Instead, we round the variables in several stages. Using this technique, we are able to achieve an approximation factor better than 2. In doing so we also answer an open question posed by [10], showing that the integrality gap of the natural LP relaxation is indeed less than 2.

We also consider the MLOP when the cost function is submodular. For *monotone* submodular functions, we obtain the following result.

Theorem 3. *For any monotone submodular function defined over a ground set of size n, there exists a deterministic algorithm for the corresponding MLOP that achieves a factor of* $2 - \frac{2}{n+1}$*.*

The algorithm is based on the Lovász extension for the given submodular function. The Lovász extensions provide a means of extending the techniques of linear programing to discrete set functions such as the ones considered here. We use the Lovász extension for the given submodular function to define a convex program that is solvable using the ellipsoid method. We then give a *deterministic* procedure to round the optimal solution to this program to get the desired integral solution.

Finally, we also consider the case when the cost function is a general submodular function, which may not be monotone. For this setting, we show an information theoretic lower bound of 2. In particular, we have the following result.

Theorem 4. *For any constant $\epsilon > 0$ there exists a family of instances of the MLOP with symmetric submodular cost functions such that no algorithm making polynomially many queries achieve a factor better than $2 - \epsilon$, even if it is given infinite computational time.*

We achieve this by constructing two families of submodular cost functions that are indistinguishable, with high probability, after polynomial number of value queries, but have different optimal objective values. The ratio of the optimal values under these functions gives the desired factor. Note that the bound is information theoretic, i.e., it holds even if the algorithm is given infinite computational time, but is constrained to make only polynomially many value queries.

1.2 Prior Work

Submodular functions have been the subject of intense study over the last four decades with regards to combinatorial optimization. Special instances of the above mentioned problems have received considerable attention from the point of view of approximation algorithms. However, we are not aware of any work that has studied these problems under a unified framework of submodular/supermodular functions. We will now review some of the problems that have previously been studied in this area.

Supermodular MLOP: Feige, Lovász, and Tetali [10] introduced the min-sum set cover (MSSC), which is a special instance of MLOP with supermodular cost functions. In this problem, we are given a hyper-graph $H = (V, E)$. For a linear ordering $\sigma : V \to \{1, \dots, n\}$ and a hyper-edge $e \in E$, let $\hat{\sigma}(e)$ denote the minimum of $\sigma(v)$ among all the vertices v in e. The goal of the min sum set cover problem is to find an linear ordering σ that minimizes $\sum_{e \in E} \hat{\sigma}(e)$. They gave a factor 4 approximation algorithm for this problem and showed that the factor was essentially tight.

They also considered the min-sum vertex cover problem described in Section 3, and gave an LP rounding based factor 2 approximation algorithm for the problem. This result was not provably tight and the integrality gap of the LP-relaxation was left as an open question. In subsequent work, Barenholz, Feige, and Peleg [5] provided a small improvement (with a rather technically involved analysis) by way of obtaining a 1.99995 factor approximation, and raised the question of further improving the bound. Answering this question, Theorem 2 below provides a substantial improvement, with an alternative rounding and a much simpler analysis, in giving a 1.79 factor approximation.

A recent paper of Azar, Gamzu, and Yin [2] discusses a generalization of the MSSC problem in the context of reranking of search results by a search engine. In this so-called Multiple Intents Ranking (MIR) problem, we are given a hyper-graph $H = (V, E)$ with each hyper-edge $e \in E$ having a vector of nonnegative reals $w(e) = \langle w_1(e), \dots, w_{r(e)}(e) \rangle$, where $r(e)$ denotes the number of vertices contained in e. For a linear ordering $\sigma : V \to \{1, \dots, n\}$ and a hyper-edge $e \in E$, let $\hat{\sigma}_i(e)$ denote the i-th smallest $\sigma(v)$ among all the vertices v in e. Then the objective is to find a linear ordering σ that minimizes $\sum_{e \in E} \sum_{i=1}^{r(e)} w_i(e)\hat{\sigma}_i(e)$. Apart from MSSC this problem also generalizes the minimum latency set cover problem introduced by Hassin and Levin [14]. We refer the reader to [1] and [3] for recent developments on the MIR problem. We were also informed of a recent (unpublished), further generalization due to Im, Nagarajan, and van der Zwaan [16]; these authors study a so-called Minimum Latency Submodular Cover (MLSC), which generalizes the submodular ranking work of Azar and Gamzu on one hand and the Latency Covering Steiner Tree on the other. See the manuscript [16] on the arxiv, for details.

Submodular MLOP: In the Minimum Linear Arrangement (MLA) problem which is a special case of submodular MLOP, we are asked to arrange the vertices of a given graph $G(V, E)$ in a linear ordering σ, so that $\sum_{(u,v) \in E} |\sigma(v) - \sigma(u)|$ is minimized. Rao and Richa [20] gave a $O(\log n)$ factor algorithms for this problem which was later improved to a $O(\sqrt{\log n} \log \log n)$ factor algorithm by Feige and Lee [9] and Charikar et. al. in [6]. The problem has also been studied on special instances and polynomial time algorithms are known for some special graphs; refer to [15] for a detailed exposition.

On the hardness front, Devanur, Khot, Saket, and Vishnoi [8] showed that the problem is hard to approximate to within any constant factor under the Unique Games Conjecture and proved that the integrality gap for the SDP relaxation of this problem is bounded from below by $O(\log \log n)$. The problem has also received considerable attention from the point of view of experimental analysis and heuristics (refer [4] [19]).

Finally, there has been recent interest in studying minimization problems with submodular cost functions [13, 11, 17]. However almost all the problems previously considered turn out to be quite intractable and have large polynomial lower bounds. Exceptions include the submodular vertex cover problem [13, 17] and the submodular multiway partition [7].

1.3 Preliminaries

A set function f is said to be submodular if $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$ holds for every $X, Y \subseteq V$. Supermodular functions are defined in a similar way; f is supermodular if $f(X) + f(Y) \leq f(X \cup Y) + f(X \cap Y)$ for every $X, Y \subseteq V$. We define a function f to be monotone if $f(X) \leq f(Y)$ for $X, Y \subseteq V$ with $X \subseteq Y$. It is called symmetric if $f(X) = f(V \setminus X)$ for every $X \subseteq V$. We assume that f is normalized, i.e., $f(\emptyset) = 0$. Note that a normalized nonnegative supermodular function is monotone, as $X \subseteq Y$ implies $f(X) \leq f(X) + f(Y \setminus X) \leq f(Y) + f(\emptyset) = f(Y)$.

Finally, a word is in order about the representation of the cost function f, since it is defined over an exponentially large domain. We use the standard *value oracle model*: that the cost function is given by a value oracle that when queried with a set S returns the value $f(S)$.

2 Supermodular Linear Ordering

This section is devoted to the linear ordering problem with supermodular cost function f. In this setting, we are given a supermodular set function f over a ground set V of size n and we are required to arrange the elements of V in a linear ordering σ such that $\sum_i f(S_i)$ is minimized, where $S_i = \{v \mid \sigma(v) \geq i\}$.

We first claim that the min sum set cover (MSSC) problem considered in [10] is a special instance of this problem. For each $X \subseteq V$ in the hyper-graph $H(V, E)$, let $f(X)$ denote the number of edges included in X. Then f is a supermodular function. Note that $e \in E$ is included in S_i if and only if $i \leq \hat{\sigma}(e)$. Therefore, we have $\sum_{i=1}^{n} f(S_i) = \sum_{e \in E} \hat{\sigma}(e)$. Thus the min sum set cover problem is a very special case of our setting with f being a nonnegative supermodular function.

More generally, the multiple intents ranking problem is a special case of MLOP where each hyper-edge $e \in E$ has a weight $w(e) = \langle w_1(e), \ldots, w_{r(e)}(e) \rangle$. For each $X \subseteq V$ and e, let $f_e(X)$ denote the sum of the last $|X \cap e|$ components of $w(e)$, and put $f(X) = \sum_{e \in E} f_e(X)$. Then $\sum_{i=1}^{n} f(S_i) = \sum_{e \in E} \sum_{j=1}^{r(e)} w_j(e) \hat{\sigma}_j(e)$ holds for any linear ordering σ. If the weight vector $w(e)$ is monotone non-increasing, i.e., $w_1(e) \geq \cdots \geq w_{r(e)}(e)$, then f_e is a supermodular function. Thus the multiple intents ranking problem with non-increasing weight vectors is a special case of MLOP with supermodular cost functions.

In contrast, if $w(e)$ is monotone non-decreasing, then f_e is monotone submodular. Thus the multiple intents ranking problem with non-decreasing weight vectors reduces to the MLOP with monotone submodular cost functions, which will be discussed in Section 4.1.

2.1 Greedy Algorithm

We will consider the following greedy algorithm for this problem. We try to iteratively build the ordering by augmenting the current solution with the element such that the cost of the remaining elements is the smallest possible. The greedy algorithm for the supermodular linear ordering problem begins by setting $T_1 = V$. Then for $i = 1, \ldots, n$, select $v \in T_i$ that minimizes $f(T_i \setminus \{v\})$ and set $\sigma(v) = i$ and $T_{i+1} = T_i \setminus \{v\}$.

2.2 Analysis

We now intend to prove that the greedy algorithm provides an approximate solution within a ratio of 4. Let $S_1, \ldots, S_n$ be the subsets given by $S_i = \{v \mid \sigma(v) \geq i\}$ with an optimal solution σ. Consider a histogram that consists of n columns. The i-th column has width $f(S_i) - f(S_{i+1})$, corresponding to the interval between $f(S_{i+1})$ and $f(S_i)$, and height i. The area of this diagram is equal to the optimal value of the problem denoted by OPT.

With reference to the subsets $T_1, \ldots, T_n$ generated by the greedy algorithm, construct another histogram that also consists of n columns. The i-th column has width $f(T_i) - f(T_{i+1})$, corresponding to the interval between $f(T_{i+1})$ and

$f(T_i)$, and height $p_i = \frac{f(T_i)}{f(T_i)-f(T_{i+1})}$. The area under this histogram is equal to the objective value of the greedy solution denoted by $GREEDY$.

Shrink the second diagram by a factor of 2. We now intend to show that this shrunk version of the second diagram is completely included in the first diagram. To see this, it suffices to check that $(f(T_i)/2, p_i/2)$ lies in the first histogram for each $i \in [n] = 1, 2, \ldots, n$.

For each fixed i, put $k = \lceil p_i/2 \rceil$. Then we now claim that $f(S_k) \geq f(T_i)/2$. In fact, by the procedure of the greedy algorithm, we have $f(T_i \setminus \{v\}) \geq f(T_{i+1})$ for each $v \in V \setminus T_i$. This implies that

$$\begin{aligned} f(S_k) \geq f(T_i \cap S_k) &\geq f(T_i) - \sum_{v \in T_i \setminus S_k} [f(T_i) - f(T_i \setminus \{v\})] \\ &\geq f(T_i) - |T_i \setminus S_k| \cdot [f(T_i) - f(T_{i+1})] \\ &\geq f(T_i) - (k-1)[f(T_i) - f(T_{i+1})] \\ &\geq f(T_i) - \frac{p_i}{2}[f(T_i) - f(T_{i+1})] = \frac{f(T_i)}{2}, \end{aligned}$$

The second inequality follows from the supermodularity of f. Thus, the second histogram is contained in the first one, which implies $GREEDY/4 \leq OPT$.

3 Min Sum Vertex Cover Problem

The Min Sum Vertex Cover (MSVC) problem is a special instance of the Min Sum Set Cover (MSSC) problem in which the given hyper-graph is a graph. We are given a graph $G(V, E)$ and the objective is to arrange the vertices of G in a linear order σ, such that the following sum is minimized, $\sum_{(u,v) \in E} \min \{\sigma(u), \sigma(v)\}$. We present a factor 1.79 approximation algorithm for MSVC.

3.1 Randomized Rounding Algorithm

We begin with the following LP relaxation of the problem. We will use $t \in [n]$ to index the positions in the ordering. Let $x_v(t)$ denote whether vertex v is present at position t and let $y_{uv}(t)$ depict if edge (u, v) is *not* covered by the vertices in the first t positions.

$$\text{Minimize} \quad \sum_{(uv) \in E} \sum_{t=1}^{n} y_{uv}(t)$$

$$\text{subject to} \quad 1 - \sum_{s \leq t} x_u(s) - \sum_{s \leq t} x_v(s) \leq y_{uv}(t) \qquad (u, v) \in E,\ \forall t, \qquad \text{(1a)}$$

$$\sum_{s=1}^{n} x_u(s) + \sum_{s=1}^{n} x_v(s) \geq 1 \qquad \forall (u, v) \in E, \qquad \text{(1b)}$$

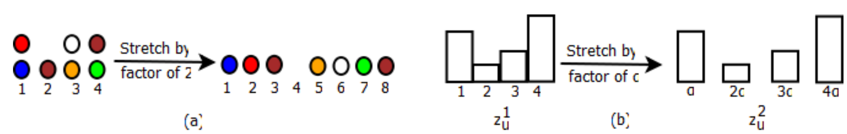
$$x_u(t) \geq 0 \qquad \forall u \in V,\ \forall t \qquad \text{(1c)}$$

$$y_{uv}(t) \geq 0 \qquad \forall (u, v) \in E,\ \forall t. \qquad \text{(1d)}$$

The LP can be solved by standard means and let (x^*, y^*) be the optimal solution to the LP. Next, we will define a rounding scheme to round (x^*, y^*) to an integer solution by assigning every vertex a unique integer value (position) on the real line.

Overview of the Algorithm: Note that in the above linear program, if we (independently and) randomly round every vertex by interpreting $x_v^*(.)$ as a probability distribution, we are not even guaranteed a feasible solution. In [10], the authors fix this problem by scaling the solution x^* *prior* to rounding as follows. Let t_v be the largest value of t for which $\sum_{s<t} x_v(t) < 1/2$. For every vertex v, they introduce a new variable $z_v(t)$ where $z_v(t) = 2x_v(t)$ for $t < t_v$, and $z_v(t_v) = 1 - \sum_{t<t_v} z_v(t)$, and $z_v(t) = 0$ for $t > t_v$. By equation (1b), for every edge uv there exists $w \in \{u, v\}$ is such that $\sum_{s=1}^{n} x_w^*(s) \geq 1/2$, which implies $\sum_{s=1}^{n} z_w^*(s) = 1$. Therefore rounding independently, according to z, will surely yield a feasible solution. One can show that the expected value of the rounded solution is at most the optimal value of the objective.

The only shortcoming of the above rounding scheme is that owing to the scaling step, on an average 2 vertices can get rounded to the same position. Intuitively, this can be rectified by *stretching* the real-axis by a factor of 2 to accommodate the extra vertices as shown in Figure 3.1(a). This gives us one a 2 approximate algorithm.



We beat this factor by interleaving the rounding and stretching subroutines in multiple phases. In each phase our algorithm independently rounds on a subset of vertices and assigns them positions on the real-line. Then we stretch the real-line by a constant factor before starting the next phase.

Algorithm Description: Let us fix some notation that would be useful in describing the algorithm. The algorithm proceeds in several phases indexed by r and in each phase it assigns positions for some vertices on the real line. We will use $p(v)$ to denote position of vertex v. For any positive integer r let $z_v^r(t) = z_v(t')$, if $t = \alpha^{(r-1)}t'$ for some constant $1 < \alpha \leq 2$ to be determined later; else, let $z_v^r(t) = 0$. That is, for $r > 1$, z^r is obtained from z^1 by stretching the real-line by a factor of α^{r-1}. Refer to Figure 3.1(b).

At the start of the algorithm assign a phase number β_v to every vertex v according to the distribution $\Pr[\beta_v = r] = 2^{-r}$ and let $S_r = \{v \mid \beta_v = r\}$. In the r-th phase all vertices in S_r are assigned positions as follows - for every vertex $v \in S_r$, randomly (and independently) assign v to position t with probability $z_v^r(t)$ i.e. set $p(v)$ to t. The algorithm terminates when all vertices are assigned a position on the real-line.

For position t, let $n_t = \sum_v p(v)$. To recover the ordering of vertices, replace position t by n_t time slots and allocate the vertices v for which $p(v) = t$ to these time slots in a random order.

3.2 Analysis

For the sake of conciseness, for any edge $uv \in E$ for a given execution of the algorithm, let us use Γ_{uv} to denote $|\{w \mid p(w) < \min\{p(u), p(v)\}\}| = \sum_{s<t} n_s$ i.e. Γ_{uv} is the number of vertices that are placed to the left of $argmin\{p(u), p(v)\}$. Thus the expected contribution of edge uv to the objective value is $\mathbb{E}[1 + \Gamma_{uv}]$. This can be further simplified using conditional expectation as shown below.

$$\mathbb{E}[1+\Gamma_{uv}] = \sum_t \Pr[\min\{p(u),p(v)\} = t] \times \mathbb{E}[1+\Gamma_{uv} | \min\{p(u),p(v)\} = t]$$

$$= \sum_t t \Pr[\min\{p(u),p(v)\} = t] \times \frac{\mathbb{E}[1+\Gamma_{uv} \mid \min\{p(u),p(v)\} = t]}{t}$$

$$\leq \sum_t t \Pr[\min\{p(u),p(v)\} = t] \times \max_t \left\{ \frac{\mathbb{E}[1+\Gamma_{uv} \mid \min\{p(u),p(v)\} = t]}{t} \right\}$$

$$\leq \mathbb{E}[\min\{p(u),p(v)\}] \times \max_t \left\{ \frac{\mathbb{E}[1+\Gamma_{uv} \mid \min\{p(u),p(v)\} = t]}{t} \right\} \tag{2a}$$

In Lemmas 1 and 2, we bound both quantities in (2a).

Lemma 1. *For any edge* $(u, v) \in E$*,*

$$\mathbb{E}[\min\{p(u),p(v)\}] \leq \frac{3}{4-\alpha} \sum_t y_{uv}(t) \,.$$

Proof. As a warm up let us begin by calculating the expected value of $\min\{p(u), p(v)\}$ given that the edge is covered during the first phase. For any $u \in V$, define $F_u(t) = \sum_{s<t} z_u^1(s)$, i.e., $F_u(t)$ is the probability that u is placed at a position to the left of t given that $\beta_u = 1$.

$\mathbb{E}[\min\{p(u), p(v)\} \mid (u, v)$ is covered in phase 1$]$

$$= \frac{\sum_t (1-F_u(t))}{3} + \frac{\sum_t (1-F_v(t))}{3} + \frac{\sum_t (1-F_u(t))(1-F_v(t))}{3} \tag{3a}$$

$$= 1/3 \sum_t 3 - 2F_u(t) - 2F_v(t) + F_u(t)F_v(t) \tag{3b}$$

$$\leq 1/3 \sum_t 3 - 2F_u(t) - 2F_v(t) + \frac{F_u(t)+F_v(t)}{2} \tag{3c}$$

$$= \sum_t 1 - \frac{F_u(t)+F_v(t)}{2} \tag{3d}$$

The first term in (3a) corresponds to the case when $r_u = 1$ and $r_v > 1$; similarly the second term corresponds to the case when $r_v = 1$ and $r_u > 1$, and the third term is the case when both $r_u = 1$ and $r_v = 1$. We get (3c) from (3b) since both F_u and F_v are bounded by 1. Finally, since $\Pr[uv \text{ is covered in phase } 1] = 3/4$, we have $\Pr[\min\{p(u), p(v)\} \,\&(u,v) \text{ is covered in phase } 1] = 3/4 \sum_t 1 - \frac{F_u(t)+F_v(t)}{2}$.

Edge (u, v) is not covered by the start of the r-th phase if both $\beta_u \geq r$ and $\beta_v \geq r$. This happens with probability $2^{-2(r-1)}$. Also since by the start of the r-th phase z_u^1 and z_v^1 have been stretched by a factor of α^{r-1}, the expected position where edge (u, v) is covered if it is covered in the r-th phase is $\alpha^{r-1} \sum_t 1 - \frac{F_u(t)+F_v(t)}{2}$. Once again, as above, the probability that uv is covered in the r^{th} phase given that it was not covered in the first $r - 1$ phases is 3/4. Combining these two facts we get,

$$
\begin{aligned}
\mathrm{E}\left[\min\{\bar{x}_u, \bar{x}_v\}\right] &= \sum_{r=1}^{\infty} 4^{-(r-1)} \frac{3\alpha^{r-1}}{4} \sum_t 1 - \frac{F_u(t) + F_v(t)}{2} \\
&= 3/4 \sum_{r=1}^{\infty} (\alpha/4)^{r-1} \sum_t \left\{ 1 - \sum_{s<t} (x_u(t) + x_v(t)) \right\} \\
&\leq 3/4 \sum_{r=0}^{\infty} (\alpha/4)^r \sum_t y_{uv}(t) \leq 3/4 \sum_t y_{uv}(t) \sum_{r=0}^{\infty} (\alpha/4)^r = \frac{3}{4-\alpha} \sum_t y_{uv}(t)\,.
\end{aligned}
$$

The last equation follows from the linear programming constraint (1a).

Lemma 2. $\max_t \left\{ \frac{\mathbb{E}\left[1 + \Gamma_{uv} \mid \min\{p(u), p(v)\} = t\right]}{t} \right\} \leq 2\alpha/(2\alpha - 1)\,.$

Proof. For any position t,

$$
\begin{aligned}
\mathbb{E}\left[1 + \Gamma_{uv} \mid \min\{p(u), p(v)\} = t\right] &= 1 + \sum_{r=1}^{\infty} \sum_{w \notin \{u,v\}} \sum_{s<t} \Pr\left[\beta_w = r\right] z_w^r(s) \\
&= 1 + \sum_{r=1}^{\infty} 2^{-r} \sum_{w \notin \{u,v\}} \sum_{s<t} z_w^r(s) \leq 1 + \sum_{r=1}^{\infty} 2^{-r} \frac{2(t-1)}{\alpha^{r-1}} \qquad \text{(5a)} \\
&\leq t \sum_{r=0}^{\infty} (2\alpha)^{-r} = 2t\alpha/(2\alpha - 1)\,. \qquad \text{(5b)}
\end{aligned}
$$

The first part of (5a) follows from the distribution from which we choose β_w and the second part is derived from the definition of z^r. Finally (5b) holds for $\alpha < 2$ and dividing throughout by t gives the desired result.

Substituting the results from Lemmas 1 and 2 in to (2a) we find that for an arbitrary edge uv, the expected contribution to the objective is at most $\frac{6\alpha}{(4-\alpha)(2\alpha-1)} \sum_t y_{uv}(t)$. For $\alpha = \sqrt{2}$, this is approximately equal to $1.79 \sum_t y_{uv}(t)$. Summing over all edges and noting that $\sum_{(uv)\in E} \sum_t y_{uv}(t)$ is a lower bound on the optimal solution, we conclude that the above algorithm approximates min sum vertex cover to within a factor of at most 1.79.

4 Submodular Linear Ordering

4.1 Monotone Submodular Functions

In this section, we discuss the minimum linear ordering problem with f being a monotone submodular function. The algorithm is based on a continuous extension for the submodular function called the Lovász extension defined below.

Definition 1. *For a set function $f : 2^V \to \mathbb{R}$ with $f(\emptyset) = 0$, its extension $\hat{f} : {\mathbb{R}_+}^V \to \mathbb{R}$ is defined by $\hat{f}(x) = \sum_{i=1}^{n} \lambda_i f(S_i)$, where $V = S_1 \supsetneq S_2 \supsetneq \cdots \supsetneq S_n \supsetneq \emptyset$ is a chain such that $\sum \lambda_i 1_{S_i} = x$ and $\lambda_i \geq 0$.*

Alternatively, one can define $\hat{f}$ by $\hat{f}(x) = \mathbb{E}\left[f(\{i : x_i > \lambda\}\right]$, where λ is uniformly random in $[0, 1]$. Note that the value $\hat{f}(x)$ is easy to compute, provided that an oracle access to f is available. Lovász [18] showed that $\hat{f}$ is convex if and only if f is submodular.

Consider the following convex optimization problem, which can be solved in polynomial time by the ellipsoid method.

$$\begin{array}{lll} \langle \text{CP} \rangle & \text{Minimize} & \hat{f}(x) \\ & \text{subject to} & \displaystyle\sum_{v \in S} x(v) \geq |S|(|S|+1)/2, \qquad \forall S \subseteq V. \end{array}$$

For a linear ordering σ, let x^σ denote a vector defined by $x^\sigma(v) = \sigma(v)$. Then x^σ is a feasible solution, and its objective value is $\hat{f}(x^\sigma) = \sum_{i=1}^{n} f(S_i)$. Thus $\langle \text{CP} \rangle$ serves as a relaxation problem. Let x^* be an optimal solution of $\langle \text{CP} \rangle$. We now consider a deterministic rounding procedure that returns a linear ordering σ so that $x^*(u) \leq x^*(v)$ implies $\sigma(u) \leq \sigma(v)$.

Lemma 3. *For each $v \in V$, we have $k \leq (2 - \frac{2}{k+1})x^*(v)$, where $k = \sigma(v)$.*

Proof. Consider the subset $S = \{u \mid \sigma(u) \leq k\}$. By the feasibility of x^*, we have $\sum_{v \in S} x^*(v) \geq k(k+1)/2$. Since $x^*(u) \leq x^*(v)$ for every $u \in S$, this implies $x^*(v) \geq (k+1)/2$. Hence we obtain $k \leq (2 - \frac{2}{k+1})x^*(v)$.

Theorem 5. *The algorithm constructs a linear ordering whose objective value is no more than $2 - \frac{2}{n+1}$ times the optimal one.*

Proof. Recall that $x^\sigma(v) = \sigma(v)$ for each $v \in V$. It follows from Lemma 3 that $x^\sigma(v) \leq \left(2 - \frac{2}{k+1}\right) x^*(v) \leq \left(2 - \frac{2}{n+1}\right) x^*(v)$, where $k = \sigma(v)$. Since $\hat{f}$ is monotone non-decreasing and $\hat{f}(\alpha x) = \alpha \hat{f}(x)$ holds for any $\alpha > 0$, this implies $\hat{f}(x^\sigma) \leq \left(2 - \frac{2}{n+1}\right) \hat{f}(x^*)$. Therefore, $\sum_{i=1}^{n} f(S_i)$ is at most $2 - \frac{2}{n+1}$ times the optimal value.

4.2 Symmetric Submodular Functions

We now focus on the linear ordering problem with f being a symmetric submodular function which includes the minimum linear arrangement problem over graphs. Given a graph $G(V, E)$ with edge capacity $c : E \to \mathbb{R}_+$, the minimum linear arrangement problem asks for finding a linear ordering $\sigma : V \to \{1, \ldots, n\}$ that minimizes $\sum_{(u,v)\in E} c(u, v)|\sigma(u) - \sigma(v)|$. Let $\kappa : 2^V \to \mathbb{R}_+$ denote the cut capacity function. For a linear ordering $\sigma : V \to \{1, \ldots, n\}$, we have

$$\sum_{i=1}^{n} \kappa(S_i) = \sum_{i=1}^{n} \sum_{(u,v)\in E,\, u\in S_i,\, v\notin S_i} c(u, v) = \sum_{(u,v)\in E} c(u, v)\, |\sigma(u) - \sigma(v)|.$$

Thus the minimum linear arrangement problem is a special case of MLOP with f being a cut function of a graph which is a symmetric submodular function.

Next, we show an unconditional information theoretic lower bound on the approximation factor for MLOP with symmetric submodular functions. This is done by defining two symmetric submodular functions f_1 and f_2 such that they achieve the same value on 'most' of the queries but have different optimal values.

Concretely, let $\delta > 0$ such that $\delta^2 = \frac{1}{n}\omega(\log n)$ and $\beta = \frac{n}{4}(1 + \delta)$. Let R be a subset of V of size $\frac{n}{2}$ then for any $S \subseteq V$, define $f_1(S) = \min\left(|S|, \frac{n}{2}\right) - \frac{|S|}{2}$ and $f_2(S) = \min\left(|S|, \frac{n}{2}, \beta + |S \cap R|, \beta + |S \cap \bar{R}|\right) - \frac{|S|}{2}$.

It can be shown that both f_1 and f_2 are nonnegative and submodular and using a result by Svitkina and Fleischer [11] we can bound the probability of distinguishing them using polynomially many value queries.

Lemma 4 ([12]). *For R chosen uniformly at random from among all subsets of V of size $\frac{n}{2}$, any algorithm that makes a polynomial number of oracle queries has probability at most $n^{\omega(1)}$ of distinguishing the functions f_1 and f_2.*

It can be shown the ratio of the optimal values of the linear arrangements under f_1 and f_2 is $2 - o(1)$, which coupled with Lemma 4 yields the following theorem. We defer the details of the proof to the full version of the paper.

Theorem 6. *For every constant $\epsilon > 0$ there exists a family of instances of the NM-MLOP such that no (computationally unbounded) algorithm making polynomially many queries to the cost function can achieve a factor better than $2 - \epsilon$.*

Acknowledgment. We thank Zoya Svitkina for suggesting Theorem 4.

References

1. Azar, Y., Gamzu, I.: Ranking with submodular valuations. In: Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '11, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (2011) 1070–1079
2. Azar, Y., Gamzu, I., Yin, X.: Multiple intents re-ranking. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing. STOC '09, New York, NY, USA, ACM (2009) 669–678

3. Bansal, N., Gupta, A., Krishnaswamy, R.: A constant factor approximation algorithm for generalized min-sum set cover. In: Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '10, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (2010) 1539–1545
4. Bar-Yehuda, R., Even, G., Feldman, J., Naor, J.S.: Computing an optimal orientation of a balanced decomposition tree for linear arrangement problems. J. Graph Algorithms Appl. **5** (2001) 1–27
5. Barenholz, U., Feige, U., Peleg, D.: Improved approximation for min-sum vertex cover. (2006)
6. Charikar, M., Hajiaghayi, M.T., Karloff, H., Rao, S.: l22 spreading metrics for vertex ordering problems. Algorithmica **56**(4) (January 2010) 577–604
7. Chekuri, C., Ene, A.: Approximation algorithms for submodular multiway partition. In: Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science. FOCS '11, Washington, DC, USA, IEEE Computer Society (2011) 807–816
8. Devanur, N.R., Khot, S.A., Saket, R., Vishnoi, N.K.: Integrality gaps for sparsest cut and minimum linear arrangement problems. In: Proceedings of the 38th Annual ACM Symposium on Theory of Computing. STOC '06, New York, NY, USA, ACM (2006) 537–546
9. Feige, U., Lee, J.R.: An improved approximation ratio for the minimum linear arrangement problem. Inf. Process. Lett. **101** (January 2007) 26–29
10. Feige, U., Lovász, L., Tetali, P.: Approximating min sum set cover. Algorithmica **40** (2004) 219–234
11. Fleischer, L., Svitkina, Z.: Submodular approximation: Sampling-based algorithms and lower bounds. In: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, Washington, DC, USA, IEEE Computer Society (2008) 697–706
12. Fleischer, L., Svitkina, Z.: Submodular approximation: Sampling-based algorithms and lower bounds. Unpublished manuscript (May 2010)
13. Goel, G., Karande, C., Tripathi, P., Wang, L.: Approximability of combinatorial problems with multi-agent submodular cost functions. In: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science. FOCS '09, Washington, DC, USA, IEEE Computer Society (2009) 755–764
14. Hassin, R., Levin, A.: An approximation algorithm for the minimum latency set cover problem. In: Proceedings of the 13th Annual European Symposium on Algorithms. ESA '05, LNCS 3669, Springer-Verlag (2005) 726–733
15. Horton, S.B.: The Optimal Linear Arrangement Problem: Algorithms and Approximation. PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA (1997) AAI9735901.
16. Im, S., Nagarajan, V., Zwaan, R.v.d.: Minimum latency submodular cover, manuscript. (2012)
17. Iwata, S., Nagano, K.: Submodular function minimization under covering constraints. In: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science. FOCS '09, Washington, DC, USA, IEEE Computer Society (2009) 671–680
18. Lovász, L.: Submodular functions and convexity. In: Mathematical Programming — The State of the Art, Springer-Verlag (1983) 235–257
19. Petit, J.: Experiments on the minimum linear arrangement problem. J. Exp. Algorithmics **8** (December 2003)

20. Rao, S., Richa, A.W.: New approximation techniques for some ordering problems. In: Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '98, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (1998) 211–218

5 Third Reviewer comments

The first result is when f is a supermodular function (this generalizes min-sum set cover). They give a 4 approximation for it. The techniques is basically same as the original Feige et al paper. Now, as a motivation for the supermodular setting, they say that this captures the multiple intents reranking problem when then weight function there is non-increasing (I am not defining this problem here). This is the only motivation they gave and so they claim credit for getting a factor 4 approximation for this problem. However, they do not seem to know that a 4-approximation was already shown by Gamzu et al in their STOC09 paper which introduced the reranking problem.

Pushkar : We do mention this paper in the related work section on page 3, final paragraph. should we highlight this result even more ?

Similarly, they show a factor 2 approximation when f is monotone submodular. However, they do not seem to know that such a result was known already for an important special case (when f corresponds to the coverage function). This reduces to a special case of precdence constrainted scheduling (and this fact is also well known among people who work on these ordering problems). This omission is also quite disturbing.

Pushkar : I am not sure which result he is referring to.

On the other hand, I must say that their technique of using Lovasz extension seems new and interesting.

Finally, I found their section 3 the most interesting result. Though I found the writing here quite poor. There are several typos, and they could have given much more intuition and explanation.