2007－AL－110 (10)
2007／1／23

劣モジュラ罰則を含む組合せ最適化問題の緩和問題に対する Lovász 拡張とノンスムーズ凸最適化手法を用いた効率的解法

Fabián A. Chudak
ETH Zurich
chudak@ifor.math.ethz.ch

永野 清仁
東京大学
kiyohito_nagano@mist.i.u-tokyo.ac.jp

概要

本研究では劣モジュラ性を持つペナルティを含む組合せ最適化問題を扱う．劣モジュラ関数の Lovász 拡張を用いることで非常に自然な緩和問題が得られる．我々はそれらを近似的に解くのに，Nesterov により近年与えられたシンプルなノンスムーズ凸最適化手法を用いた手法を提案する．

Efficient solutions to relaxations of combinatorial problems with submodular penalties via the Lovász extension and non-smooth convex optimization

Fabián A. Chudak
ETH Zurich

Kiyohito Nagano
University of Tokyo

Abstract

We consider convex relaxations for combinatorial optimization problems with submodular penalties. The relaxations are obtained very naturally through a novel use of the Lovász extension of a submodular function. We also propose the use of simple and recent algorithms for non-smooth convex optimization due to Nesterov to approximately solve them.

1 Introduction

The purpose of this paper is twofold. On one hand, we introduce the use of the Lovász extension [14] to relax submodularity constraints. On the other hand, we show how the recent nonlinear programming results of Nesterov [15, 16, 18, 19] on minimization of non-smooth convex functions can be used to design fast approximation schemes for complex convex relaxations of combinatorial optimization problems.

In [14] Lovász made a simple and clever connection between convex optimization and submodular function minimization. More precisely, he showed how to extend a submodular set function defined on the subsets of a ground set V, $|V| = n$, to a convex function defined on $\mathbb{R}^n$. This convex function is the maximum of linear functions defined over the quite non-trivial base polytope of the submodular function; nevertheless, its evaluation at any $z \in \mathbb{R}^n$ can be computed very efficiently using what is called the greedy algorithm. Because of its simple structure, this same procedure provides a subgradient and, hence, it can be used for minimizing the extended function using the ellipsoid algorithm. These observations were at the core of the first polynomial (and strongly polynomial) time algorithms for submodular function minimization (see [8]). Combinatorial algorithms (which in particular do not use the ellipsoid algorithm) for submodular function minimization were discovered later [12, 21].

Motivated by the stark contrast between the algorithmic solutions to single-commodity and multicommodity flow problems, a great deal of research has been devoted to finding approximation schemes for specially structured linear programs (see [1]). On the one hand, these algorithms are inspired by and possess a combinatorial structure. On the other hand, these algorithms usually can also be interpreted as subgradient algorithms for certain nonlinear non-smooth Lagrangian relaxation. In fact, it seems that this latter way of thinking provides a greater variety of applications and allows also the treatment of more general convex problems. We adopt this approach in this paper.

Subgradient algorithms run in $\Theta(\frac{1}{\epsilon^2})$ iterations (see, e.g., [20]) where $\epsilon > 0$ is the desired absolute accuracy, though it must be stressed that they are not necessarily polynomial on other parameters of the problem. These algorithms compute projections only on the primal space. In standard form (e.g., [20]) the run time depends on the size of the subgradients. However, it was shown in [19, 17] that this quantity can be replaced by a bound on the variation of the subgradients, which, in particular, allows one to ignore linear terms. If, in addition to primal projections, dual projections can also be computed efficiently, the substantially stronger dependence on ϵ of $O(\frac{1}{\epsilon})$ iterations can be achieved (see [15, 16]). This is accomplished by making the objective function differentiable and applying very fast methods for smooth convex optimization (thus we refer to it as in

[15] as smooth minimization of non-smooth functions). As remarked before, in this case also the dependence on other parameters of the problem is typically not polynomial.

In this paper we consider a generalization of the uncapacitated facility location problem (UFL). In the standard variant of UFL, we have a set of clients and a set of sites in which we may open facilities and the goal is to minimize the facility opening costs plus the assignment costs of all the clients to the open facilities (see [24] for a survey). Here we study the variant of UFL that has submodular penalties (FLwP). In this problem not all clients need to necessarily be serviced, instead there is a monotone submodular function that penalizes the set of unserviced clients. This problem was introduced by Hayrapetyan, Swamy and Tardos [9] which, in turn, generalized the work of Charikar et al [4] who considered a linear penalty function. They proposed a $1+\alpha$ approximation algorithm for the case on which the assignment costs are a metric, where α is the approximation factor of an LP-based approximation algorithm for the corresponding metric case of UFL. Their algorithm uses the ellipsoid algorithm to solve a linear programming relaxation of FLwP. In this paper, we consider an equivalent relaxation for the problem. In contrast with [9] we arrive to a compact convex formulation of the relaxation fairly naturally using the Lovász extension. Furthermore we propose more efficient algorithms to solve it using the algorithms for non-smooth convex optimization mentioned above. Using the rounding algorithm of [9], we obtain efficient $(1+\varepsilon)(1+\alpha)$ approximation algorithms for FLwP.

We point out that our reformulation of FLwP is nontrivial and that, even though we are using convex optimization algorithms and some of its terminology, the algorithms themselves are very simple (and even have a somewhat combinatorial flavor, see [3]). To transform these non-polynomial algorithms into a FPTAS we prove a characterization of optimal solutions to the relaxation using elementary properties of submodular functions and the Lovász extension. We then combine this characterization with binary search as in [3] (see also [1, 2, 25]). Thus we obtain $O(\frac{1}{\epsilon^2})$ and $O(\frac{1}{\epsilon})$ FPTAS for the problem. The $O(\frac{1}{\epsilon^2})$ algorithms are very simple and can be applied to any submodular function h. The $O(\frac{1}{\epsilon})$ algorithms could also be applied to any h, though the dual projections need a number of calls to a generic submodular function minimization routine, hence affecting the practicality of such algorithms. However, for important special cases, these dual projections can be computed efficiently, thus providing very fast and practical algorithms. The use of smooth minimization of non-smooth functions to design FPTAS was introduced by Bienstock & Iyengar [2] and later more in the flavor of this paper by Chudak & Eleuterio [3]. To the best of our knowledge the use of subgradient optimization results of the type [19, 20] seems to be new. As a by-product of our methods we obtain a very simple pseudo-polynomial time algorithm for integer-valued submodular function minimization, which, in contrast with the algorithms of [5, 12, 21], does not need to keep track of extreme bases.

We also consider a set covering problem with submodular costs, also introduced in [9] which is a quite general problem. It generalizes the set covering problem, FLwP and facility location problems involving set-based facility costs [22, 23]. For the relaxation of this problem using the Lovász extension, we show that the integrality gap is best possible and design efficient FPTAS by extending the subgradient and the smooth minimization methods.

2 Preliminaries

2.1 The Lovász extension of a submodular function.

We briefly review known facts about submodular functions (see, e.g., [7]) that we will need later. Let V be a finite set with $|V| = n$. A set function $\rho : 2^V \to \mathbb{R}$ is *submodular* if $\rho(X \cup \{v\}) - \rho(X) \geq \rho(Y \cup \{v\}) - \rho(Y)$ for each $X \subseteq Y \subseteq V$ and $v \in V \setminus Y$. The interpretation of this definition is that submodular functions naturally model economies of scale. Furthermore ρ is *monotone* if $\rho(X) \leq \rho(Y)$ for each $X \subseteq Y \subseteq V$. In the following, we assume that ρ is a submodular function with $\rho(\varnothing) = 0$. The *base polytope* of ρ, $\mathbf{B}(\rho)$, is defined by $\mathbf{B}(\rho) = \{x \in \mathbb{R}^n \mid \sum_{v\in X} x_v \leq \rho(X)\ (\forall X \subseteq V), \sum_{v\in V} x_v = \rho(V)\}$. The set $\mathbf{B}(\rho)$ is nonempty and bounded. Furthermore, ρ is monotone if and only if $\mathbf{B}(\rho) \subseteq \mathbb{R}^n_+$. All the vertices of the base polytope, the *extreme bases*, can be obtained as follows. Let $L = (v_1, \ldots, v_n)$ be a linear ordering of V. Define $L(v_k) = \{v_1, v_2, \ldots, v_k\}$ for each $k = 1, \ldots, n$. Then, the vector $b = (b_v)_{v\in V} \in \mathbb{R}^n$ defined by $b_v = \rho(L(v)) - \rho(L(v) \setminus \{v\})$ (for each $v \in V$) is an extreme base of $\mathbf{B}(\rho)$. In this case we will say that b is the *extreme base generated by* L.

The the Lovász extension [14] of a submodular set function ρ with $\rho(\varnothing) = 0$ is a function $\widehat{\rho} : \mathbb{R}^n_+ \to \mathbb{R}$ with the property that for each subset $X \subseteq V$, $\widehat{\rho}(I_X) = \rho(X)$, where $I_X \in \mathbb{R}^n_+$ is the indicator function of set X (i.e., $z_v = 1$ if $v \in X$ and 0 otherwise). For $z = (z_v)_{v\in V} \in \mathbb{R}^n_+$, $\widehat{\rho}(z)$ is given by the value of the following LP

$$\widehat{\rho}(z) = \max \sum_{v\in V} z_v b_v \ \text{ s.t. } \ b \in \mathbf{B}(\rho). \tag{1}$$

Note that the number of constraints of this LP is exponential. Nevertheless, its value can be easily computed. If $L = (v_1, \ldots, v_n)$ is a linear ordering *compatible* with z, that is, $z_{v_1} \geq \cdots \geq z_{v_n}$, then the extreme base generated by L is an optimal solution of (1). This procedure is known as the *greedy algorithm* and takes $O(n \log n + n\gamma)$ time where γ denotes the time of one function evaluation. For $b \in \mathbf{B}(\rho)$ and $X \subseteq V$, X is *b-tight* with respect to ρ if $\sum_{v\in X} b_v = \rho(X)$. Note that if b is an extreme base generated by linear ordering L, $L(v) \subseteq V$ is b-tight for each $v \in V$. As we introduced it, clearly the function $\widehat{\rho}$ is convex (as the max of linear functions) and positively homogeneous. In addition, if ρ is monotone, $\widehat{\rho}$ is also monotone in $\mathbb{R}^n_+$.

Let z be an arbitrary vector in $\mathbb{R}^n_+$, $L = (v_1, \ldots, v_n)$ a linear ordering of V compatible with z and $b \in \mathbf{B}(\rho)$ be an extreme base generated by L. Then from above, the value of the Lovász extension $\widehat{\rho}(z)$ is equal to $\widehat{\rho}(z) = z_{v_n}\rho(L(v_n)) + \sum_{k=1}^{n-1}(z_{v_k} - z_{v_{k+1}})\rho(L(v_k))$. In addition, the following representation will also be useful. Let $p_1 > \cdots > p_d$ be the positive distinct values of z and $p_{d+1} = 0$. For $1 \leq a \leq d$, define $\lambda_a = p_a - p_{a+1}$ and $X_a = \{v \in V \mid z_v \geq p_a\}$. Then, we have that the sets X_a are b-tight and $\widehat{\rho}(z) = \sum_{a=1}^{d} \lambda_a \rho(X_a)$.

2.2 Non-smooth convex optimization.

In this section we review basic and simple algorithms for minimizing a convex function over a convex set. The exposition covers recent results of Nesterov [15, 16, 18, 19]. Suppose that $Q_1 \subseteq \mathbb{R}^n$, $Q_2 \subseteq \mathbb{R}^m$ are bounded, closed and convex sets and $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$ and $a \in \mathbb{R}^m$. Consider the pair of convex and concave functions respectively ($x \in \mathbb{R}^n, u \in \mathbb{R}^m$)

$$(2) \qquad \begin{aligned} f(x) &= \max_{u \in Q_2}\{c^t x + \langle Ax, u\rangle + a^t u\} \text{ and} \\ \phi(u) &= \min_{x \in Q_1}\{c^t x + \langle Ax, u\rangle + a^t u\}, \end{aligned}$$

where $\langle Ax, u\rangle = (Ax)^t u$. Note that $f(x) \geq \phi(u)$ (weak-duality) for any $x \in Q_1$, $u \in Q_2$. In fact, it is a simple exercise of convex analysis to show that $\min_{x \in Q_1} f(x) = \max_{u \in Q_2} \phi(u)$. Next we review general algorithms from convex optimization to approximately solve this min-max problem.

2.2.1 Subgradient algorithms.

We assume that the primal space $\mathbb{R}^n$ is endowed with a given norm $\|.\|$. The linear–algebra dual norm is given by $\|s\|^* = \max\{\langle s, x\rangle : \|x\| = 1\}$, for $s \in \mathbb{R}^n$ as usually. Furthermore we have a strongly convex function $d : Q_1 \longrightarrow \mathbb{R}^+$ (called prox-function), with strong convexity parameter σ. If $x_o \in Q_1$ is the minimizer of d, we also assume that $d(x_o) = 0$, so that $d(x) \geq \frac{\sigma}{2}\|x - x_o\|^2$. Next, note that in our case, for a given $\overline{x} \in Q_1$, a subgradient of f at $\overline{x}$ is any vector of the form $c + A^t\overline{u}$, where $\overline{u} \in \mathbb{R}^m$ is an optimal solution of the max problem defining $f(\overline{x})$. For $\overline{x} \in Q_1$ we use $\partial f(\overline{x})$ to denote the set of subgradients of f at $\overline{x}$. Let $D = \max_{x \in Q_1} d(x)$. Suppose that for $M > 0$ the variation of subgradients is bounded by M (i.e., $\|\mathcal{G}(x) - \mathcal{G}(y)\|^* \leq M$ for all $x, y \in Q_1, \mathcal{G}(x) \in \partial f(x), \mathcal{G}(y) \in \partial f(y)$).

Theorem 2.1 ([18, 19]) *There is an algorithm that after k iterations finds $\widehat{x}_k \in Q_1$ and $\widehat{u}_k \in Q_2$ satisfying $f(\widehat{x}_k) - \phi(\widehat{u}_k) \leq \frac{M}{\sqrt{k}}\sqrt{\frac{8D}{\sigma}}$. At each iteration, we need to compute a subgradient of f at some $x \in Q_1$ and solve a subproblem of the form $\min\{p^\top x + d(x) \mid x \in Q_1\}$ for some $p \in \mathbb{R}^n$.*

Notice that M and D are not necessarily polynomial in general. Furthermore to run a subgradient algorithm we must be able to efficiently compute subgradients and solve the *projections* $\min\{p^\top x + d(x) \mid x \in Q_1\}$. If $\epsilon > 0$ is the desired absolute accuracy, these algorithms run in $O(1/\epsilon^2)$ iterations.

2.2.2 Smooth minimization of non-smooth functions.

Substantially stronger convergence results can be obtained if we also assume that we can compute projections on the space Q_2. First, to ease the notation, we use subindex 1 for the norm and prox-function in the x-space, so that $\|.\|$, d, σ and D become, respectively, $\|.\|_{①}$, d_1, σ_1 and D_1. Similarly as in the x-space, in the u-space we fix a norm and a strongly convex function, and define the objects $\|.\|_{②}$, d_2, u_o, σ_2 and D_2, respectively. The norm of A as a bilinear operator is defined as usual: $\|A\| = \max\{\langle Ax, u\rangle \mid \|x\|_{①} = 1,\ \|u\|_{②} = 1\}$.

Theorem 2.2 ([15, 16]) *There is an algorithm that after k iterations finds $\widehat{x}_k \in Q_1$ and $\widehat{u} \in Q_2$ satisfying $f(\widehat{x}_k) - \phi(\widehat{u}_k) \leq \frac{4\|A\|}{k+1}\sqrt{\frac{D_1 D_2}{\sigma_1\sigma_2}}$. At each iteration, we have to find the exact solutions of three problems of the form $\min\{p^\top x + d_1(x) \mid x \in Q_1\}$ or $\min\{q^\top u + d_2(u) \mid u \in Q_2\}$ for some $p \in \mathbb{R}^n$, $q \in \mathbb{R}^m$ and perform a matrix-vector multiplication (either Ax or $A^t u$).*

As remarked for the subgradient algorithm, the algorithm of the theorem is generally not polynomial (all of $\|A\|, D_1, D_2$ need not be polynomial). Now if $\epsilon > 0$ is the desired absolute accuracy, the algorithm of the theorem runs in $O(1/\epsilon)$ iterations.

3 Uncapacitated facility location with submodular penalties

The uncapacitated facility location problem with submodular penalties (FLwP) can be described as follows. We are given a set of potential facility sites $\mathcal{F}$ with $|\mathcal{F}| = m$, and a set of clients $\mathcal{D}$ with $|\mathcal{D}| = n$. If we open a facility at site $i \in \mathcal{F}$, we incur an opening cost $f_i \geq 0$, and it can provide service to any number of clients. Each client may be serviced by an open facility. If client $j \in \mathcal{D}$ is serviced by an open facility at location $i \in \mathcal{F}$, the cost is c_{ij}. If X is the set of clients that are not serviced by any facility, there is a submodular penalty cost of $h(X)$, where $h : 2^{\mathcal{D}} \to \mathbb{R}$ is a monotone submodular function with $h(\varnothing) = 0$ and $W := h(\mathcal{D}) > 0$. Let γ denote the time for one evaluation of h. For simplicity, we assume that γ is at least linear in $\log n$. The objective is to determine which facilities to open and which clients to service so as to minimize the total cost.

3.1 A convex relaxation.

Hayrapetyan *et al.* [9] gave a linear programming relaxation for FLwP that has exponentially many variables and argued that it can be solved with the ellipsoid algorithm. In this section, we introduce the same relaxation somewhat more naturally using the Lovász extension of h. We use variable y_i to indicate whether we open a facility at location $i \in \mathcal{F}$; variable x_{ij} indicates whether client j is assigned to an open facility at location $i \in \mathcal{F}$; and variable z_j indicates whether client j is not serviced

by any facility. Next we can write the following convex relaxation.

$$
\begin{aligned}
\min \quad & \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{D}} c_{ij}x_{ij} + \sum_{i\in\mathcal{F}} f_i y_i + \widehat{h}(z) \\
\text{s. t.} \quad & \sum_{i\in\mathcal{F}} x_{ij} + z_j = 1, \quad \text{for client } j \in \mathcal{D}, \\
& x_{ij} \le y_i, \quad \text{for } i \in \mathcal{F},\ j \in \mathcal{D}, \\
& x_{ij}, y_i, z_j \ge 0, \quad \text{for } i \in \mathcal{F},\ j \in \mathcal{D},
\end{aligned} \tag{3}
$$

where $\widehat{h}$ is the Lovász extension of h. To see that (3) is indeed a relaxation of FLwP, suppose that we have an integer solution (x, y, z). Notice that for client $j \in \mathcal{D}$, the equation guarantees that it is either serviced by some facility (one x_{ij} is 1) or not serviced (z_j is 1). The following inequalities make sure that a client is assigned to an open facility. The first two terms in the objective function take into account service and facility opening costs. Finally if X is the set of clients not serviced by any facility, clearly z is the indicator function of X and $h(X) = \widehat{h}(z)$ follows from Section 2.1. It is not hard to show using the dual formulation of the LP (1) that (3) is equivalent to the formulation given in [9].

We will use $\mathsf{OPT}_{\mathrm{RX}}$ to denote the optimal value of (3). We introduce the notation Δ_n for the n-dimensional simplex, that is, $\Delta_n = \{p \in \mathbb{R}^n : \sum_{i=1}^n p_i = 1, p_i \ge 0\}$. Also we will think of x as a matrix in $\mathbb{R}^{m\times n}$. We will use x' to denote the matrix x with the additional row z at the bottom, so that $x' \in \mathbb{R}^{(m+1)\times n}$. Now the equality constraints in (3) can be summarized by saying that each column of x' is in Δ_{m+1}; for brevity we will simply write $x' \in \Delta_{m+1}^n$. Next we change the form of (3) to better fit the algorithms of Section 2.2. As in [3], it is clear that any feasible solution to (3) has smaller value if we take $y_i = \max_{j\in\mathcal{D}} x_{ij}$ for each $i \in \mathcal{F}$. Furthermore, if $(u_{ij}) \in \mathbb{R}^{m\times n}$, and for each $i \in \mathcal{F}$ we let $u_i = (u_{ij})_j \in \mathbb{R}^n$, we can also write $y_i = \max_{u_i\in\Delta_n} \sum_{j=1}^n u_{ij}x_{ij}$; similarly as before, if $u \in \mathbb{R}^{m\times n}$, $u_i \in \Delta_n$ for each $i \in \mathcal{F}$, we simply write $u \in \Delta_n^m$. Finally, it will be advantageous to work with a scaled version of h. To this end, we define $\rho : 2^{\mathcal{D}} \to \mathbb{R}$ by $\rho(X) = \frac{h(X)}{W}$ $(X \subseteq \mathcal{D})$. As $\rho(\mathcal{D}) = 1$ and $\mathbf{B}(h) \subseteq \mathbb{R}_+^n$, we have $\mathbf{B}(\rho) \subseteq \Delta_n$. Combining these notations and observations together with (1), we have the following reformulation of (3)

$$
\begin{aligned}
\mathsf{OPT}_{\mathrm{RX}} = \min_{x'\in\Delta_{m+1}^n} \max_{\substack{u\in\Delta_n^m \\ b\in\mathbf{B}(\rho)}} \Big\{ & \sum_{\substack{i\in\mathcal{F} \\ j\in\mathcal{D}}} c_{ij}x_{ij} + \sum_{\substack{i\in\mathcal{F} \\ j\in\mathcal{D}}} f_i u_{ij} x_{ij} \\
& + \sum_{j\in\mathcal{D}} W z_j b_j \Big\},
\end{aligned} \tag{4}
$$

which is exactly of the needed form (2). Define function $f : \mathbb{R}^{(m+1)n} \to \mathbb{R}$ so that $f(x')$ is exactly the max inside the min-max formula (4). Thus, if $Q := \Delta_{m+1}^n$, the relaxation (4) reduces to the non-smooth convex optimization problem $\mathsf{OPT}_{\mathrm{RX}} = \min_{x'\in Q} f(x')$.

3.2 Solving the relaxation.

We apply the two methods described in Section 2.2 to find an approximate solution to (4). First we show that the direct application of these methods leads to pseudo-polynomial time algorithms. In the next section we show how to design FPTAS.

3.2.1 Subgradient optmization.

For fixed $\overline{x}' \in Q = \Delta_{m+1}^n$, let $(\overline{u}, \overline{b})$ be an optimal solution that defines $f(\overline{x}')$. Then the vector $\mathcal{G}(\overline{x}') \in \mathbb{R}^{(m+1)\times n}$ with $c_{ij} + f_i\overline{u}_{ij}$ in the (linear-algebra) dual coordinate of x_{ij} and $W\overline{b}_j$ in the (linear-algebra) dual coordinate of z_j is a subgradient of f at $\overline{x}'$. Such a vector can be easily obtained in $O(mn + n\gamma + n\log n)$ time using the results of Section 2.1.

For $x' \in Q$, we choose the Euclidean norm $\|x'\|^2 = \sum_{i,j} x_{ij}^2 + \sum_j z_j^2$. Thus $\|\cdot\|^*$ is also the Euclidean norm. Let $F = \max\{\max_{i\in\mathcal{F}} f_i, W\}$. We define the convex set $\widehat{\Delta}_n := \Delta_n - \Delta_n = \{p - q \in \mathbb{R}^n : p, q \in \Delta_n\}$. Clearly $\widehat{\Delta}_n \subseteq V_0(2) = \{p \in \mathbb{R}^n \mid \sum_{j=1}^n |p_j|^2 \le 2\}$, the ball of radius 2 centered at 0. Since $\mathbf{B}(\rho) \subseteq \Delta_n$ and $0 \in \widehat{\Delta}_n$, it is straightforward to verify that $\mathcal{G}(x_1') - \mathcal{G}(x_2') \in F\widehat{\Delta}_n^{m+1}$ for any $x_1', x_2' \in Q$ and thus the variation of subgradients of f is bounded as follows

$$
\begin{aligned}
\|\mathcal{G}(x_1') - \mathcal{G}(x_2')\| &\le \max_{w\in F\widehat{\Delta}_n^{m+1}} \|w\| \\
&\le F\sqrt{m+1}\max_{w\in V_0(2)} \|w\| = \sqrt{2(m+1)}F\,.
\end{aligned}
$$

We define $d(x') = \frac{1}{2}\|x' - x_\circ'\|^2$ where $x_\circ'$ is chosen so as to minimize the constant D in Theorem 2.1, thus $(x_\circ)_{ij} = (z_\circ)_j = \frac{1}{m+1}$ for each i and j. Then function d is strongly convex on Q with respect to $\|\cdot\|$ and $\sigma = 1$. Let $D := \max_{x'\in Q} d(x') = \frac{1}{2}n\frac{m}{m+1}$. To apply the algorithm of Theorem 2.1 we have to solve projections of the form

$$
\min \ p^\top x' + d(x') \ \text{ s.t. } \ x' \in Q = \Delta_{m+1}^n \tag{5}
$$

for some $p \in \mathbb{R}^{(m+1)\times n}$. It is easy to see that these projections decompose into n problems of the form $\min_{w\in\Delta_{m+1}} q^t w + \sum_j w_j^2$ which, in turn, can be solved in linear time (see, e. g., [3, §2]). Thus (5) can be solved in $\mathrm{O}(mn)$. Hence, from Theorem 2.1, we obtain the following.

Theorem 3.1 *There is an algorithm that after k iterations finds a solution $\widehat{x} \in Q$ of relaxation (4) with $f(\widehat{x}') - \mathsf{OPT}_{\mathrm{RX}} \le \frac{\sqrt{8mn}}{\sqrt{k}}F$. Each iteration can be implemented in $\mathrm{O}(n(m + \log n + \gamma))$ time.*

3.2.2 Smooth minimization of non-smooth functions.

In this section we solve the min-max problem (4) using the techniques of Section 2.2. Here we have that $Q_1 = Q = \Delta_{m+1}^n$ and $Q_2 := \Delta_n^m \times \mathbf{B}(\rho) \subseteq \Delta_n^{m+1}$. If $u \in \Delta_n^m$ and $b \in \mathbf{B}(\rho)$, we will use $u' := (u, b) \in Q_2$. Note that $\langle Ax', u'\rangle = \sum_{i,j} f_i x_{ij} u_{ij} + \sum_j W z_j b_j$. We choose the Euclidean norms in each space, that is, $\|x'\|_{①}^2 = \sum_{i,j} x_{ij}^2 + \sum_j z_j^2$ and $\|u'\|_{②}^2 = \sum_{i,j} u_{ij}^2 + \sum_j b_j^2$; to ease the notation we drop the indices ① and ②. As in the previous section, we take $d_1(x') = \frac{1}{2}\|x' - x_\circ'\|^2$, and similarly for space Q_2 , $d_2(u') = \frac{1}{2}\|u' - u_\circ'\|^2$. Clearly $\sigma_1 = \sigma_2 = 1$. Here $x_\circ'$ and $u_\circ'$ are chosen so as to minimize D_1 and D_2 of Theorem 2.2; thus, $(x_\circ)_{ij} = (z_\circ)_j = \frac{1}{m+1}$ and $u_\circ' = (u_\circ, b_\circ) = \arg\min_{u'\in Q_2} \|u'\|^2$ where $(u_\circ)_{ij} = \frac{1}{n}$ and the computation of $b_\circ \in \mathbf{B}(\rho)$ will be

discussed later. Then $D_1 = \max_{x' \in Q_1} d_1(x') = \frac{1}{2} n \frac{m}{m+1}$ and for $D_2 = \max_{u' \in Q_2} d_2(u')$ we use the upper bound $\frac{1}{2}(m+1)$. Finally we bound the operator norm $\|A\|$ using the Cauchy-Schwartz inequality as follows

$$\begin{aligned}\|A\| &\leq \max_{\|x'\|=\|u'\|=1} \Big\{ \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{D}} f_i |x_{ij} u_{ij}| + \sum_{j \in \mathcal{D}} W |z_j b_j| \Big\} \\ &\leq \max\{\max_i f_i, W\} = F.\end{aligned}$$

Next we elaborate on how to solve the subproblems of Theorem 2.2. These are of the form $\min\{p^\top x + d_1(x) \,|\, x \in Q_1\}$ and $\min\{q^\top u + d_2(u) \,|\, u \in Q_2\}$. First notice that the former are exactly as (5) thus can be solved in $O(mn)$ time. For the latter, notice that they decompose into $m+1$ subproblems in which the first m are quadratic projections onto the simplex Δ_n which exactly as we argued before can be solve in $O(nm)$ and the last one is of the form

$$\min \Big\{ \sum_{j \in \mathcal{D}} \big(q_j b_j + \tfrac{1}{2} b_j^2\big) \,:\, b \in \mathbf{B}(\rho) \Big\}. \tag{6}$$

Let QMB denote the time required to solve this quadratic optimization problem. Then each iteration of the algorithm of Theorem 2.2 can be implemented in $O(mn + \mathsf{QMB})$ (notice that A is sparse, so Ax' and $A^t u'$ can be computed in $O(mn)$ time). Note that the computation of $b_\circ$ is a special case of (6) (actually any $b_\circ \in \mathbf{B}(\rho)$ would work). Applying Theorem 2.2, we obtain the following.

Theorem 3.2 *There is an iterative algorithm that after k iterations finds a feasible solution $\widehat{x}' \in Q$ of (3) such that $f(\widehat{x}') - \mathsf{OPT}_{\mathrm{RX}} \leq \frac{2\sqrt{mn}}{k+1} F$. Each iteration can be implemented in $O(nm + \mathsf{QMB})$.*

In general, problem (6) can be solved by performing submodular function minimization at most n times (see [6], [7, §§7–9]). To get the greatest possible speed up with respect to ϵ of Theorem 2.2 we would need to solve it in strongly polynomial time. Alternatively, using the approximate version of the theorem from [3] we could settle for a weakly polynomial time algorithm at the expense of an additional $O(\log(1/\epsilon))$ factor. Due to the results of [12, 21, 8], both alternatives are possible (with the results of [12, 21] we would have an overall "combinatorial algorithm"). The running time, though, for the general case would make the subgradient algorithms of the previous section more attractive (and much simpler). On the other hand, in many important special cases problem (6) can be solved efficiently.

Functions arising from concave cardinality functions. Let $\theta : \{0, 1, \ldots, n\} \to \mathbb{R}$ be a nondecreasing concave function with $\theta(0) = 0$. Then, a function ρ_{C} defined by $\rho_{\mathrm{C}}(X) = \theta(|X|)$ for each $X \subseteq V$ is monotone and submodular. One can show that problem (6) can be solved in $\mathrm{O}(n \log n)$ time using Fujishige's decomposition algorithm [6].

Hierarchical functions. We are given a rooted undirected tree $T = (U, E)$. The leaves of T correspond to V. Each vertex $v \in U$ has a nonnegative cost a_v. For each subset $X \subseteq V$, let N_X be the set of vertices of the minimum subtree including the root s and leaves X. It is easy to see that the set function $\rho_{\mathrm{H}} : 2^V \to \mathbb{R}$ with $h(\varnothing) = 0$ defined by $\rho_{\mathrm{H}}(X) = \sum_v \{a_v : v \in N_X\}$ for each subset $X \subseteq V$ is monotone and submodular. This function can model the costs of multiple levels of service installation. Iwata and Zuiki [13] showed that (6) can be solved in $\mathrm{O}(n \log n)$ time using the algorithm proposed by Hochbaum and Hong [11]. As pointed out by an anonymous reviewer, in this special case it is possible to write down a compact formulation for this case of FLwP. However, though polynomial, this formulation may be difficult to solve in practice for large instances (the simplest case of the LP without penalties is already difficult, see [3]). In contrast, our algorithms are very simple to implement and scalable.

3.3 Binary search.

First of all we state a well-known and efficient procedure to transform absolute error algorithms into relative error algorithms. Let (P) be a minimization problem with objective function f_{P} with positive optimal value $\mathsf{OPT}_{\mathrm{P}}$. Suppose that given a target value $R > 0$ and a relative error $0 < \eta \leq 1$ and that we have an algorithm $\mathsf{Feas}_{\mathrm{P}}(R, \eta)$ which either decides that $\mathsf{OPT}_{\mathrm{P}} > R$ or finds a feasible solution x such that $f_{\mathrm{P}}(x)$ is less than or equal to $(1+\eta)R$. We assume further that the running time of $\mathsf{Feas}_{\mathrm{P}}(R, \eta)$ does not depend on R and is bounded by a polynomial on $\frac{1}{\eta}$ of the form $\mathrm{T}(\eta) = \mathrm{T}_1(\eta) + \mathrm{T}_2 = \frac{1}{\eta} q(\frac{1}{\eta}) + \mathrm{T}_2$ (here T_2 does not depend on η). The following lemma is from Young [25].

Lemma 3.3 ([25]) *Suppose that x is a feasible solution of (P) and, $\mathsf{LB} > 0$ and $t > 0$ are such that $\mathsf{LB} \leq \mathsf{OPT}_{\mathrm{P}} \leq f_{\mathrm{P}}(x) \leq t\,\mathsf{LB}$. Then we can find a feasible solution $\widehat{x}$ of (P) such that $f_{\mathrm{P}}(\widehat{x}) \leq (1+\varepsilon)\mathsf{OPT}_{\mathrm{P}}$ in $\mathrm{O}\big(\log\log t \cdot \mathrm{T}(\frac{1}{2}) + \mathrm{T}_1(\varepsilon) + \log\frac{1}{\varepsilon} \cdot \mathrm{T}_2\big)$ time.*

We can easily compute a feasible integer solution to (4), and a lower bound LB with $t = n$ as follows. For each client $j \in \mathcal{D}$, let $\mathsf{LB}_j = \min\{h(\{j\}), \min_i\{c_{ij} + f_i\}\}$, and let $\mathsf{LB} = \max_j \mathsf{LB}_j$. For each $j \in \mathcal{D}$, if $\mathsf{LB}_j = h(\{j\})$, we set $\overline{z}_j = 1$; otherwise we set $\overline{x}_{ij}$ to 1 for a facility i with $\mathsf{LB}_j = c_{ij} + f_i$. The following lemma is straightforward.

Lemma 3.4 *A feasible solution $\overline{x}' \in Q$ with $\mathsf{LB} \leq \mathsf{OPT}_{\mathrm{RX}} \leq f(\overline{x}') \leq n\mathsf{LB}$ can be computed in $O(mn + n\gamma)$ time.*

The design of the feasibility procedure $\mathsf{Feas}_{\mathrm{P}}(R, \eta)$ is more delicate and it requires the following structural lemma about the optimal solutions to problem (3). For a finite set I and a vector $p = \{p_i \in \mathbb{R} \,|\, i \in I\}$, we denote $\mathrm{supp}(p) = \{i \in I \,|\, p_i \neq 0\} \subseteq I$.

Lemma 3.5 *Given a feasible solution $\overline{x}' = (\overline{x}, \overline{z}) \in Q$, there is another feasible solution $\widehat{x}' = (\widehat{x}, \widehat{z}) \in Q$ which satisfies properties: (i) $f(\widehat{x}') \leq f(\overline{x}')$, (ii) $f_i \leq f(\overline{x}')$, if $\widehat{x}_{ij} > 0$ ($j \in \mathcal{D}$, $i \in \mathcal{F}$), and (iii) $h(\mathrm{supp}(\widehat{z})) \leq f(\overline{x}')$.*

Proof: Define $g(x) = \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{D}} c_{ij}x_{ij} + \sum_{i\in\mathcal{F}} f_i(\max_{j\in\mathcal{D}} x_{ij})$ ($x \in \mathbb{R}^{mn}$). For $x' \in Q$, we have $f(x') = g(x) + \widehat{h}(z)$. First we give a vector $\widetilde{x}' = (\widetilde{x}, \widetilde{z}) \in Q$ such that $f(\widetilde{x}') \le f(\overline{x}')$ and $f_i \le f(\overline{x}')$, if $\widetilde{x}_{ij} > 0$ ($j \in \mathcal{D}, i \in \mathcal{F}$). After that, we give a vector $\widehat{x}' \in Q$ satisfying (i), (ii) and (iii).

Let $F^\circ = \{i \in \mathcal{F} \mid f_i > f(\overline{x}')\}$. If $F^\circ \neq \varnothing$, we set $\widetilde{x}$ and $\widetilde{z}$ by eliminating facilities F° as follows

$$(7)\qquad \widetilde{x}_{ij} = \begin{cases} 0 & \text{for } i \in F^\circ,\ j \in \mathcal{D} \\ \dfrac{\overline{x}_{ij}}{1-\sum_{i'\in F^\circ}\overline{x}_{i'j}} & \text{for } i \in \mathcal{F}\setminus F^\circ,\ j \in \mathcal{D}, \end{cases}$$
$$\widetilde{z}_j = \frac{\overline{z}_j}{1-\sum_{i'\in F^\circ}\overline{x}_{i'j}} \quad \text{for } j \in \mathcal{D}.$$

We can easily see that $\widetilde{x}' = (\widetilde{x}, \widetilde{z}) \in Q$. Let $y_{F^\circ} = \sum_{i\in F^\circ}(\max_{j\in\mathcal{D}} x_{ij}) \in \mathbb{R}_+$; by the definition of F°, $y_{F^\circ} < 1$ and also $\widetilde{z} \le \frac{\overline{z}}{1-y_{F^\circ}}$. As $\widehat{h}$ is monotone and positively homogeneous, we have $\widehat{h}(\widetilde{z}) \le \widehat{h}(\frac{\overline{z}}{1-y_{F^\circ}}) = \frac{1}{1-y_{F^\circ}}\cdot\widehat{h}(\overline{z})$. Furthermore, since $g(\widetilde{x}) \le \frac{1}{1-y_{F^\circ}}\big(g(\overline{x}) - \sum_{i'\in F^\circ} f_{i'}\cdot(\max_{j\in\mathcal{D}}\overline{x}_{i'j})\big) < \frac{1}{1-y_{F^\circ}}\big(g(\overline{x}) - y_{F^\circ}f(\overline{x}')\big)$, we obtain $f(\widetilde{x}') < \frac{1}{1-y_{F^\circ}}(g(\overline{x}) + \widehat{h}(\overline{z}) - y_{F^\circ}f(\overline{x}')) = f(\overline{x}')$.

If $h(\mathrm{supp}(\widetilde{z})) \le f(\overline{x}')$, then set $\widehat{x}' = \widetilde{x}'$. So we assume otherwise. Let $p_1 > \cdots > p_d$ be the distinct values of $\widetilde{z}$ other than value 0 and $p_{d+1} = 0$. For $1 \le a \le d$, define $\lambda_a = p_a - p_{a+1}$ and $X_a = \{j \in \mathcal{D} \mid z_j \ge p_a\}$. From Section 2.1, $\widehat{h}(\widetilde{z}) = \sum_{a=1}^{d}\lambda_a h(X_a)$. Let s be the minimum index such that $h(X_s) > f(\overline{x}')$. We iteratively generate $(\widehat{x}^{(k)}, \widehat{z}^{(k)}) \in Q$ for $0 \le k \le d-s+1$. We maintain $f(\widehat{x}^{(k)}, \widehat{z}^{(k)}) \le f(\overline{x}')$ and $\mathrm{supp}(\widehat{z}^{(k)}) = X_{d-k}$ for each k. Initially we set $\widehat{x}^{(0)} = \widetilde{x}$ and $\widehat{z}^{(0)} = \widetilde{z}$. For $0 \le k \le d-s$, we set $\widehat{x}^{(k+1)}$ and $\widehat{z}^{(k+1)}$ by reducing $\mathrm{supp}(\widehat{z}^{(k)})$ as follows

$$(8)\qquad \widehat{x}_{ij}^{(k+1)} = \begin{cases} \dfrac{x_{ij}^{(k)}}{1-\lambda_{d-k}} & \text{for } i \in \mathcal{F},\ j \in X_{d-k} \\ x_{ij}^{(k)} & \text{for } i \in \mathcal{F},\ j \in \mathcal{D}\setminus X_{d-k}, \end{cases}$$
$$\widehat{z}_j^{(k+1)} = \begin{cases} \dfrac{\widehat{z}_j^{(k)}-\lambda_{d-k}}{1-\lambda_{d-k}} & \text{for } j \in X_{d-k} \\ 0 & \text{for } j \in \mathcal{D}\setminus X_{d-k}, \end{cases}$$

and update $\lambda_a := \frac{\lambda_a}{1-\lambda_{k-d}}$ for $1 \le a \le d-k-1$. By induction, we can see that $(\widehat{x}^{(k)}, \widehat{z}^{(k)}) \in Q$ and $\mathrm{supp}(\widehat{z}^{(k)}) = X_{d-k}$ for each k. Observe that $g(\widehat{x}^{(k+1)}) \le \frac{1}{1-\lambda_{d-k}}g(\widehat{x}^{(k)})$ and that $\widehat{h}(\widehat{z}^{(k+1)}) = \frac{1}{1-\lambda_{d-k}}(\widehat{h}(\widehat{z}^{(k)}) - \lambda_{d-k}h(X_{d-k}))$. So we have

$$f(\widehat{x}^{(k+1)}, \widehat{z}^{(k+1)}) \le \tfrac{1}{1-\lambda_{d-k}}\big(f(\widehat{x}^{(k)}, \widehat{z}^{(k)}) - \lambda_{d-k}h(X_{d-k})\big) < f(\overline{x}').$$

Finally we set $\widehat{x} = \widehat{x}^{(d-s+1)}$ and $\widehat{z} = \widehat{z}^{(d-s+1)}$. Clearly $\widehat{x}'$ satisfies (i), (ii) and (iii). □ □

Corollary 3.6 *Given $\overline{x}' \in Q$, we can find $\widehat{x}' \in Q$ as in Lemma 3.5 in $O(n(\gamma+m))$ time.*

Next let $R > 0$ and $\eta \in (0, 1]$. A simple consequence of Lemma 3.5 is that if $f_{i_\circ} > R$ we can remove facility $i_\circ$ (and, thus, all the variables $x_{i_\circ j}$ disappear from the formulation). Similarly, using the monotonicity of h, if $h(\{j_\circ\}) > R$ we can set $z_{j_\circ} = 0$ (i.e., eliminate variable $z_{j_\circ}$). The reason is that if $\mathsf{OPT}_{\mathrm{RX}} \le R$, the lemma ensures that the optimum does not change in either case. These observations already allows us to design a feasibility procedure for our problems (using that $h(S) \le \sum_{s\in S} h(\{s\})$ and that the subproblems can only become easier). However, we can obtain better bounds if we further *truncate* the submodular function h. Let $R' = (1+\eta)R$, $\mathcal{D}_R = \{j \in \mathcal{D} \mid h(\{j\}) \le R\}$, and $\mathcal{F}_R = \{i \in \mathcal{F} : f_i \le R\}$. We define the truncation of function h, $h_{R'} : 2^{\mathcal{D}_R} \to \mathbb{R}$ as $h_{R'}(X) = \min\{R', h(X)\}$. The following lemma is easy to prove.

Lemma 3.7 *The truncation $h_{R'}$ is submodular and monotone with $h_{R'}(\varnothing) = 0$.*

Note that the truncation of a submodular function need not be submodular. Also $h_{R'}$ may have a different structure than that of h; thus, problem (6) may become difficult for $h_{R'}$, while it was easy for h. For example, if h arises from some cardinality concave function, $h_{R'}$ always has the same structure. However, if h is a hierarchical function, the restriction of h to $\mathcal{D}' \subseteq \mathcal{D}$ is also hierarchical but the truncation $h_{R'}$ need not be. This makes a difference only for smoothing-type algorithms (Section 2.2.2.), while subgradient algorithms are always better off using truncation.

To design the feasibility procedure $\mathsf{Feas}_{\mathrm{RX}}(R, \eta)$, we apply either Theorem 3.1 or 3.2 to the instance of the problem in which all facilities with $f_i > R$ are removed, all the variables z_j are set to 0 whenever $h(\{j\}) > R$ and replace h by $h_{R'}$; let Q_1' be the feasible region of this problem, f' be the objective function value and $\mathsf{OPT}_{\mathrm{RX}}{}'$ be the optimal solution value of this problem instance. It is easy to see that we can take F to be R'. We will run the algorithms so that in the formulas of the theorems the bound on the optimality gap becomes strictly less than ηR. Hence using either algorithm, we obtain $\overline{x}' \in Q_1'$ such that $f'(\overline{x}') < \eta R + \mathsf{OPT}_{\mathrm{RX}}{}'$. Next we apply Corollary 3.6 to obtain $\hat{x}' \in Q_1'$ with $f'(\hat{x}') \le f(\overline{x}')$ and satisfying that $h_{R'}(\mathrm{supp}(\hat{z})) \le f(\overline{x}')$. Next let $\widetilde{x}' \in Q_1$ is the lifting of x' by adding zeros and check whether $f(\widetilde{x}') \le (1+\eta R)$. If the inequality holds, we found the solution needed; if not, we decide that $\mathsf{OPT}_{\mathrm{RX}} > R$.

To prove the validity of $\mathsf{Feas}_{\mathrm{RX}}(R, \eta)$ suppose that $\mathsf{OPT}_{\mathrm{RX}} \le R$. Then using Theorem 3.5, we know that eliminating the variables does not change the optimum. Furthermore, using the definition of truncation, $\mathsf{OPT}_{\mathrm{RX}}{}' \le \mathsf{OPT}_{\mathrm{RX}}$. Thus, we have that

$$f'(\hat{x}') \le f'(\overline{x}') < \eta R + \mathsf{OPT}_{\mathrm{RX}}{}' \le \eta R + \mathsf{OPT}_{\mathrm{RX}} \le (1+\eta)R = R'.$$

Finally, since $h_{R'}(\mathrm{supp}(\hat{z})) < R'$, it follows that $h_{R'}(\mathrm{supp}(\hat{z})) = h(\mathrm{supp}(\hat{z}))$ which, using that the Lovász extension can be computed via the greedy algorithm, implies that $f(\widetilde{x}') = f'(\hat{x}')$.

We fill in the details about the number of iterations that $\mathsf{Feas}_{\mathrm{RX}}$ has to run. The subgradient algorithm needs $O(\frac{mn}{\eta^2})$ iterations; the smoothing algorithm needs

$O(\frac{\sqrt{mn}}{\eta})$ iterations using truncation and $O(\frac{\sqrt{(m+n)n}}{\eta})$ eliminating variables only.

Theorem 3.8 *Given $\varepsilon > 0$, we can find a feasible solution $\widehat{x}' \in Q$ satisfying $f(\widehat{x}') \leq (1+\varepsilon)\mathsf{OPT}_{\mathsf{RX}}$ in $O(mn^2(m + \log n + \gamma)(\log\log n + \frac{1}{\varepsilon^2}))$ time using as a subroutine the algorithm of Theorem 2.1. Alternatively if we use as a subroutine the algorithm of Theorem 2.2, the running time is $O(\sqrt{mn}(mn + \mathsf{QMB})(\frac{1}{\varepsilon} + \log\log n) + n\gamma(\log(\frac{1}{\varepsilon}) + \log\log n))$.*

We conclude this section by noting that the algorithms of Theorem 3.8 can be used to design an approximation algorithm for FLwP when the service costs are symmetric and satisfy the triangle inequality using exactly the same algorithm as in [9]. In short, after approximately solving the relaxation (4), we obtain $\widehat{x}' = (\widehat{x}, \widehat{z})$. Let $b \in \mathbf{B}(h)$ an extreme base generated by a linear ordering compatible with $\widehat{z}$. Let $\mathcal{A}_{\mathsf{UFL}}$ be an LP-based α_{UFL}-approximation algorithm for the metric uncapacitated facility location problem. Then the clients that are serviced, $N \subseteq \mathcal{D}$, are those for which $\widehat{z}_j \leq \frac{1}{1+\alpha_{\mathrm{UFL}}}$, and apply $\mathcal{A}_{\mathsf{UFL}}$ to the resulting instance of UFL. The only observation needed to complete the arguments of [9] is that $h(\mathcal{D} - N) \leq (1 + \alpha_{\mathrm{UFL}})\widehat{h}(\widehat{z})$, but this follows immediately from the fact that $\mathcal{D} - N$ is b-tight (from Section 2.1). Thus, we can provide a $(1+\epsilon)(1+\alpha_{\mathrm{UFL}})$-approximation algorithm as in [9] without invoking the ellipsoid algorithm.

3.4 Submodular function minimzation.

Let V be a finite set with $|V| = n$, and $r : 2^V \to \mathbb{R}$ be a submodular function; wlog $r(\varnothing) = 0$. In this section we consider the submodular function minimization problem, i.e., $\min_{X \subseteq V} r(X)$ by reducing it to FLwP as follows. The set of clients $\mathcal{D} = V$ and there is only one facility ($m = 1$) with opening cost equal to 0. We define the service costs as $c(v) = \max\{0, r(V \setminus \{v\}) - r(V)\}$ ($v \in V$). For $X \subseteq V$, we use $c(X) = \sum_{v \in X} c_v$; let $C = c(V)$. Define $h : 2^V \to \mathbb{R}$ as $h(X) = r(X) + c(X)$. Clearly h is submodular, $h(\varnothing) = 0$ and it is not hard to show that it is also monotone. Furthermore, $c(V \setminus X) + h(X) = r(X) + C$ for any $X \subseteq V$ and $\sum_{v \in V} c_v(1 - z_v) + \widehat{h}(z) = \widehat{r}(z) + C$ for any $z \in \mathbb{R}^n_+$.

From [14] it is known that $\min_{X \subseteq V} r(X) = \min_{z \in [0,1]^n} \widehat{r}(z)$ and if $z^* \in [0,1]^n$ is an optimal fractional solution, one can round it in $O(n^2 + n\gamma)$ time. Thus, from Theorem 3.1 we get a very simple algorithm for submodular minimzation that for $\epsilon > 0$ finds a set X with $r(X)$ within absolute error of ϵ in $O(\frac{1}{\epsilon^2}(r(\mathcal{D}) + C)^2 n^2(\gamma + \log n))$. In particular, if r is integer valued, we get a pseudo-polynomial time exact algorithm for submodular function minimization that, in contrast with previous algorithms such as [5, 12, 21], does not need to keep track of extreme bases. We end this section by pointing out that we can extend Lemma 3.5 so that it can handle negative service costs, provided that the objective function of the problem is sufficiently positive: in general, if we assume that $r(\varnothing) > 0$, we have $r(X) = h(X) - c(X) + r(\varnothing)$, we need the condition that $r(\varnothing) \geq C$. Thus, we can obtain FPTAS for certain special cases of submodular function minimization with running time $O(\frac{1}{\epsilon^2}n^2(\gamma + \log n))$ which may have practical advantages compared to current exact algorithms for submodular minimzation.

4 Set covering with submodular costs

The set covering problem with submodular costs (SCwS) [9] is described as follows. We are given a finite ground set V with $|V| = n$ and a collection of subsets of V, $\mathcal{S} = \{S_1, \ldots, S_m\}$ with $\cup_{i=1}^m S_i = V$ and $\sum_{i=1}^m |S_i| = K$ ($\leq mn$). We would like to find a partition of V into m sets $T_i \subseteq S_i$, some of which might be empty. For each $1 \leq i \leq m$, the cost of T_i is given by a monotone submodular set function $h_i : 2^{S_i} \to \mathbb{R}$ with $h_i(\varnothing) = 0$ and $W_i := h_i(S_i) > 0$. Let γ denote the upper bound on the time needed to evaluate any h_i. The cost of a partition $T_1, \ldots, T_m$ is $\sum_{i=1}^m h_i(T_i)$ and the goal is to find a minimum cost partition. This problem is a generalization of set covering, FLwP and other facility location problems involving set-based facility costs [22, 23] in which costs are given by hierarchical functions. If $m = 2$, this problem can be solved in polynomial time by performing submodular function minimization. If $m \geq 3$, however, it is NP-hard because of the hardness of multiway cuts.

As in Section 3, we can obtain a convex relaxation using the Lovász extension as follows. For $v \in V$, let $A_v = \{i : v \in S_i\} \subseteq \{1, \ldots, m\}$. Let $z \in \mathbb{R}^K$, $z = (z_{iv})_{i \in A_v}$. We use the notation $z_i = \{z_{iv} \,|\, v \in S_i\} \in \mathbb{R}^{|S_i|}$ and $z_v = \{z_{iv} \,|\, i \in A_v\} \in \mathbb{R}^{|A_v|}$. In a binary solution z_{iv} indicates whether $v \in S_i$ and $T_i = \{v : z_{iv} = 1\}$. Thus, the following is a relaxation of SCwS:

$$(9) \quad \min \sum_{i=1}^m \widehat{h}_i(z_i) \;\; \text{s.t.} \sum_{i \in A_v} z_{iv} = 1 \text{ for } v \in V, \; z \geq \mathbf{0}.$$

We denote by $\mathsf{OPT}_{\mathsf{RX}}$ the optimal value of (9) and define $f(z) = \sum_i \widehat{h}_i(z_i)$. For each $i = 1, \ldots, m$, let $\rho_i := \frac{h_i}{W_i}$, which implies $\mathbf{B}(\rho_i) \subseteq \Delta_{|S_i|}$. Using the scaled base polytopes, (9) becomes

$$\mathsf{OPT}_{\mathsf{RX}} = \min_{z \in \prod_v \Delta_{|A_v|}} \max_{b \in \prod_i \mathbf{B}(\rho_i)} \{\sum_{i=1}^m \sum_{v \in S_i} W_i z_{iv} b_{iv}\},$$

which is of the form (2). Using the transformation of SCwS into an exponentially large set covering problem, we can show that the quality of the relaxation is best possible, that is, within a factor of $H(n) = \sum_{k=1}^n \frac{1}{k} \leq \ln n + 1$. We can also extend Lemma 3.5 and Theorem 3.8 as follows.

Lemma 4.1 *For a solution $\overline{z}$ of (9), there exists a solution $\widehat{z}$ satisfying (i) $f(\widehat{z}) \leq f(\overline{z})$ and (ii) $h_i(\mathrm{supp}(\widehat{z}_i)) \leq f(\overline{z})$ for each $i = 1, \ldots, m$. Such a solution $\widehat{z}$ can be found in $O(K(\gamma + m))$ time.*

Theorem 4.2 *Given $\varepsilon > 0$, we can find a feasible solution $\widehat{z}$ of (9) such that $f(\widehat{z}) \leq (1+\varepsilon)\mathsf{OPT}_{\mathrm{RX}}$ in $\mathrm{O}(mnK(\gamma + \log n)(\frac{1}{\varepsilon^2} + \log\log n))$ time using the algorithm of Theorem 2.1 as a subroutine. Alternatively using Theorem 2.2, the running time is $\mathrm{O}^*(\frac{1}{\varepsilon}\sqrt{mn}(K + m\mathsf{QMB}) + \log(\frac{1}{\varepsilon})Km)$.*

We denote by μ the maximum frequency of any element, $\max_{v\in V}|A_v|$. Let $\widehat{z}$ be a $1+\varepsilon$ approximation solution of (9). By setting $\hat{T}_i = \{v \in S_i \setminus \cup_{i'=1}^{i-1}\hat{T}_{i'} : \widehat{z}_{iv} \geq \frac{1}{\mu}\}$ for each $1 \leq i \leq m$, we can also provide a $(1+\varepsilon)\mu$ approximation algorithm for SCwC using techniques similar to [10].

We conclude this section by pointing out that we can design algorithms for FLwP when the penalty function has the more complex form $h^*(X) = \min\{\sum_{i=1}^m h_i(T_i) : (T_1, \ldots, T_m)$ partition of X, $T_i \subseteq S_i\}$. Note that h^* is not necessarily submodular. This problem can be cast into a special case of SCwS, but can be solved more efficiently if treated as we did FLwP. For the case in which the assigment costs form a metric, a similar rounding procedure as in [9] leads to a $(1+\varepsilon)(\alpha_{\mathrm{UFL}} + \mu)$ approximation algorithm.

Acknowledgements

We are grateful to Satoru Iwata for several useful discussions.

References

[1] D. Bienstock. Potential Function Methods for Approximately Solving Linear Programming Problems: Theory and Practice. CORE Lecture Series, U. Catholique de Louvain, Belgium, 2001.

[2] D. Bienstock and G. Iyengar: Solving fractional packing problems in $O^*(\frac{1}{\varepsilon})$ iterations. In *STOC '04*, pp. 146–155.

[3] F. A. Chudak and V. Eleutério: Improved approximation schemes for linear programming relaxations of combinatorial optimization problems. In *IPCO '05*, pp. 81–96.

[4] M. Charikar, S. Khuller, D. Mount and G. Narasimhan: Algorithms for facility location problems with outliers. In *SODA '01*.

[5] W. H. Cunningham: On submodular function minimiztion. *Combinatorica*, **5** (1985), pp. 185–192.

[6] S. Fujishige: Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, **5** (1980), pp. 186–196.

[7] S. Fujishige: *Submodular Functions and Optimization (Second Edition)*. Elsevier, Amsterdam, 2005.

[8] M. Groetschel, L. Lovász and A. Schrijver: Geometric Algorithms and Combinatorial Optimization, Springer, Berlin, 1988

[9] A. Hayrapetyan, C. Swamy and É. Tardos: Network design for information networks. In *SODA '05*.

[10] D. S. Hochbaum: Approximation algorithms for the set covering and vertex cover problems, *SIAM Journal on Computing*, **11** (1982), pp. 555–556.

[11] D. S. Hochbaum and S.-P. Hong: About strongly polynomial time algorithm for quadratic optimization over submodular constraints. *Mathematical Programming*, **69** (1995), pp. 269–309.

[12] S. Iwata, L. Fleischer, and S. Fujishige: A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, **48** (2001), pp. 761–777.

[13] S. Iwata and N. Zuiki: A network flow approach to cost allocation for rooted trees. *Networks*, **44** (2004), pp. 297–301.

[14] L. Lovász: Submodular functions and convexity. *Mathematical Programming — The State of the Art* (A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, 1983), pp. 235–257.

[15] Yu. Nesterov: Smooth minimization of non-smooth functions. *Mathematical Programming*, **103** (2005), pp. 127–152.

[16] Yu. Nesterov: Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, **16** (2005), pp. 235–249.

[17] Yu. Nesterov: Dual extrapolation and its applications for solving variational inequalities and related problems. CORE Discussion Paper #2003/68, CORE 2003.

[18] Yu. Nesterov: Primal-dual subgradient methods for convex problems. Technical report, CORE, 2005.

[19] Yu. Nesterov: Minimizing functions with bounded variation of subgradients. Technical report, CORE, 2005.

[20] Yu. Nesterov: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers, 2003.

[21] A. Schrijver: A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory (B)*, **80** (2000), pp. 346–355.

[22] D. Shmoys, C. Swamy and R. Levi: Facility location with service installation costs. In *SODA '04*.

[23] Z. Svitkina and É. Tardos: Facility location with hierarchical facility costs. In *SODA '06*.

[24] J. Vygen: Approximation algorithms for facility location problems (lecture notes). Report No. 05950-OR, Research Institute for Discrete Mathematics, University of Bonn, 2005.

[25] N. Young: Sequential and parallel algorithms for mixed packing and covering. In *FOCS '01*.