



HAL
open science

Prize-collecting set multicovering with submodular pricing

Daniel Cosmin Porumbel

► **To cite this version:**

Daniel Cosmin Porumbel. Prize-collecting set multicovering with submodular pricing. *International Transactions in Operational Research*, 2018, 25 (4), pp.1221-1239. 10.1111/itor.12420 . hal-02542673

HAL Id: hal-02542673

<https://hal.science/hal-02542673v1>

Submitted on 29 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Prize-Collecting Set Multi-Covering With Submodular Pricing

Daniel Porumbel^a

^aCEDRIC CS Lab., Conservatoire National des Arts et Métiers, Paris, France

Abstract

We consider a ground set E and a submodular function $\bar{f} : 2^E \rightarrow \mathbb{R}$ acting on it. We first propose a **Set Multi-Covering** problem in which the value (price) of any $S \subseteq E$ is $\bar{f}(S)$. We show that the Linear Program (LP) of this problem can be directly solved by applying a Submodular Function Minimization (SFM) algorithm on the dual LP. However, the main results of this study concern **Prize-Collecting Multi-Covering With Submodular Pricing**, *i.e.*, a more general and more difficult **Multi-Covering** problem in which elements can be left uncovered by paying a penalty. We formulate it as a large-scale LP (with $2^{|E|}$ variables representing all subsets of E) that could be tackled by Column Generation (CG), see [18] for a CG algorithm for **Set-Covering** problems with submodular pricing. However, we do *not* solve this large-scale LP by CG, but we solve it in polynomial time by exploiting its integrality properties. More exactly, after appropriate restructuring, the dual LP can be transformed into an LP that has been thoroughly studied (as a primal) in the SFM theory. Solving this LP reduces to optimizing a strong map of $\mathcal{O}(n)$ submodular functions. For this, we use the Fleischer-Iwata framework [6] that optimizes all these $\mathcal{O}(n)$ functions within the same asymptotic running time as solving a single SFM, *i.e.*, in $\mathcal{O}(n^7\gamma + n^8)$, where $n = |E|$ and γ is the complexity of one submodular evaluation. Besides solving the problem, the proposed approach can be useful to: (1) simultaneously find the best solution of up to $\mathcal{O}(n^5)$ functions under strong map relations in $\mathcal{O}(n^8\gamma + n^9)$ time, (2) perform sensitivity analysis in very short time (even at no extra cost), (3) reveal combinatorial insight into the primal-dual optimal solutions. We present several potential applications along the paper, from production planning to combinatorial auctions.

1 Introduction

A function $f : 2^E \rightarrow \mathbb{R}$ is submodular if and only if $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$, $\forall S, T \subseteq E$. A function f is called supermodular if the inequality is reversed, *i.e.*, if $f(S) + f(T) \leq f(S \cup T) + f(S \cap T)$, $\forall S, T \subseteq E$. The optimization literature is replete with application examples in which one needs to perform Submodular Function Minimization (SFM), *i.e.*, find a subset $S \subseteq E$ of minimum submodular value $f(S)$. A vast body of work has been dedicated to SFM and important progress has been done over the last five decades (see, chronologically, papers such as [5, 11, 15, 6, 8, 12, 13, 17, 14, 9]). This paper is devoted to a classical **Set Multi-Covering** problem (cover w_e copies of each $e \in E$ with subsets $S \subseteq E$ of minimum total price) in which subsets $S \subseteq E$ are valued at a price of $\bar{f}(S)$ by a submodular function \bar{f} . The main result concern the prize-collecting formulation of this problem, *i.e.*, a long-acknowledged set-covering variant [1] in which some of the w_e copies can be left uncovered by paying a penalty x_e .

A classical approach to such problems uses an Integer Linear Program (ILP) with $2^{|E|}$ columns representing all subsets of E . After linear relaxation, this leads to an LP with prohibitively many variables that is typically optimized by Column Generation (CG) methods [2, 3]. A paradigmatic example of such CG LPs is the Gilmore-Gomory model for **Cutting-Stock**. It asks to find the minimum number of columns (subsets $S \subseteq E$ satisfying a knapsack constraint) that cover w_e copies of each $e \in E$. Our **Set-Covering** problem can always be formulated in this manner, but the column of subset $S \subseteq E$ has a price (objective function coefficient) of $\bar{f}(S)$ instead of 1. To optimize such LPs, a CG approach would first start out with a restricted set of columns; then, at each iteration, a negative reduced cost column would be added by solving a CG pricing subproblem. If the price of S is $\bar{f}(S)$, then this subproblem is a SFM problem, as it asks to minimize a submodular function minus a linear function, *i.e.*, $\bar{f}(S) - \sum_{e \in S} z_e$, where z_e is the dual value of e .

The disadvantage of a pure CG algorithm is that it might need too many iterations, *i.e.*, generate too many columns by solving a SFM problem. To overcome this, we show that the dual of the CG program can be written as an LP that has been thoroughly studied as a primal program in SFM. In fact, if the objective is not formulated in a prize-collecting manner, we obtain the submodular polytope in the dual CG LP. In a prize-collecting formulation, we obtain a well-known polytope¹ discovered by Edmonds [5] and used by most combinatorial SFM algorithms. The program associated to this polytope has integer optimum dual solutions, and so, the CG program has an integer solution and there is *no* need for branching to solve it.

Section 3 presents all proposed algorithms; they rely on performing SFM over $\mathcal{O}(n)$ subsets of E that we construct as follows. Given that each element e is demanded w_e times, we construct an embedding sequence of subsets $E_k \subsetneq E_{k-1} \cdots \subsetneq E_0 = E$ such that E_k contain the most demanded elements (highest w_e), $E_{k-1} \setminus E_k$ contain the second most demanded elements, $E_{k-2} \setminus E_{k-1}$ contain the next most demanded ones, etc. We will show in Section 3.2 that the most general problem version, the **Prize-Collecting Multi-Covering With Submodular Pricing**, can be solved by performing SFM over the above family of subsets. This can be done by optimizing a strong map sequence of submodular functions, using only one call to the Fleischer-Iwata algorithm of complexity $\mathcal{O}(n^7\gamma + n^8)$, *i.e.*, we use the push-relabel Fleischer-Iwata Framework (FIF).

While this approach is not necessarily faster than (naively) applying $\mathcal{O}(n)$ times the fastest-available SFM optimizer (*i.e.*, Orlin’s algorithm [14], to our knowledge), the FIF is more well-suited to SFM over a family of subsets or over a strong map sequence of submodular functions. In fact, it can simultaneously find in $\mathcal{O}(n^8\gamma + n^9)$ time the optimal solution for a strong map sequence of up to $\mathcal{O}(n^5)$ pricing scenarios (Section 3.3.2.2), which is faster than applying $\mathcal{O}(n^5)$ times a black-box optimizer. Such simultaneity properties can be further used to perform sensitivity analysis, especially if one can anticipate possible data changes and formalize them using strong map functions. Last but not least, by investigating how the arcs of the push-relabel FIF approach generalize on our problem (Section 3.2.1), we reveal more combinatorial insight into the problem structure.

Several applications are presented along the paper, starting with Examples 2.1.A-2.1.B and finishing with Section 4 on related work. For instance, Section 4.1 shows that the related **Submodular Cost Set-Covering** (SCSC) does not verify one of the main assumptions required in this work, *i.e.*, the price of columns is not submodular. The “submodular cost” part in SCSC does not refer to the valuation over the power set of E , but over the power set of a family of subsets of E .

The remainder of this paper is organized as follows. Section 2 introduces the LP models. Section 3 presents solution methods for these LPs, followed by a sensitivity analysis (Section 3.3) and extensions towards optimization with parametric pricing functions (Section 3.3.2). Section 4 is devoted to related ideas and applications, followed by conclusions in the last section.

2 Problem Formal Model

A **Set Multi-Covering** problem is defined on a ground set $E = \{1, 2, \dots, n\}$, a vector of demands $\mathbf{w} = [w_1 \ w_2 \ \dots \ w_n]^\top$ and a pricing (valuation) function $\bar{f} : 2^E \rightarrow \mathbb{R}$. The goal is to find a *family of subsets* of the form $S \subseteq E$ of minimum total value (cost) that cover exactly w_e copies each element $e \in E$.² We assume $w_e \in \mathbb{Z}_+, \forall e \in E$, but some fractional situations will also be discussed. We only consider here functions \bar{f} that are submodular over the power set of E and that satisfy $f(\emptyset) = 0$ (any submodular function can be normalized to this form).

It is often more convenient to use the following submodularity definition. We say $f : 2^E \rightarrow \mathbb{R}$ is submodular if and only if:

$$f(S' \cup \{e\}) - f(S') \geq f(S \cup \{e\}) - f(S), \forall S' \subseteq S \subsetneq S \cup \{e\} \subseteq E \quad (2.1)$$

¹The polytope $\{z_e : z_e \leq f(S) \forall S, z_e \leq 0 \forall e \in E\}$ for submodular function f .

²The constraint of *exactly* covering the given set can be seen as a partitioning constraint. However, the “covering logic” will be more obvious in the prize-collecting versions studied from Section 2.2 on.

2.1 A Basic CG Model and Two Motivating Applications

In light of above (2.1), the notion of submodular pricing captures the idea of *diminishing marginal pricing*: the value added by a new element e to a larger set S is smaller than the value added by e to a smaller set $S' \subseteq S$. For the sake of clarity, we formulate an introductory, very simple, application of **Set-Covering with Submodular Pricing**.

Example 2.1.A. Consider a customer that needs to purchase (exactly) w_e copies of each product $e \in E$. A seller applies a submodular function \bar{f} to set prices over product bundles $S \subseteq E$. What selection of product bundles minimizes the total price for the customer?

When above \bar{f} is non-decreasing (a reasonable condition unless selling at unfairly low prices as in dumping), \bar{f} is a matroid rank function. The following application better reveals the more global notion of submodularity (e.g., \bar{f} is either positive or negative, and usually not monotone).

Example 2.1.B. We consider a set E of facilities such that each facility $e \in E$ has a (production) demand to work exactly w_e time slots (days). There is a fixed cost $c_e > 0$ per each slot (day) of work at e . We also consider a benefit $b_{e,e'} > 0$ when facilities e and e' work at the same time—e.g., a cost reduction due to resource sharing or resource complementarity. Formally, we define the submodular function $\bar{f}(S) = \sum_{e \in S} c_e - \sum_{e,e' \in S} b_{e,e'}$. The goal is to find a work schedule for all facilities that minimizes the total cost.

We formulate below the primal-dual linear relaxation of **Set-Covering with Submodular Pricing**. Program (2.2a) has 2^n variables π_S (one for each $S \subseteq E$) and could be optimized by a classical CG routine that would iteratively solve the pricing sub-problem: $\min_{S \subseteq E} \bar{f}(S) - \sum_{e \in S} z_e$. However, our solution method is presented in Section 3.1, where we will use the fact that the optimal solution of (2.2a) is always integer.

$$z_e : \begin{array}{l} \min \sum_{S \subseteq E} \bar{f}(S) \pi_S \\ \sum_{S \ni e} \pi_S = w_e \quad \forall e \in E \\ \pi_S \geq 0 \quad \forall S \subseteq E \end{array} \quad (2.2a) \quad \left. \begin{array}{l} \max \mathbf{w}^\top \mathbf{z} \\ \sum_{e \in S} z_e \leq \bar{f}(S) \quad \forall S \subseteq E \\ z_e \in \mathbb{R} \quad \forall e \in E \end{array} \right\} P(\bar{f}) \quad (2.2b)$$

2.2 Prize Collecting and Edmond's Submodular Polytope

Prize-collecting is a property of covering problems that has been long acknowledged in the optimization literature [1]. The constraint “cover exactly w_e copies of $e \in E$ ” simply becomes “cover at most w_e copies but pay a penalty x_e for any uncovered copy”. The examples 2.1.A-2.1.B evolve as follows.

Example 2.2.A. Use the same interpretation from Example 2.1.A, but the customer's exact request of purchasing w_e copies of $e \in E$ evolves: there is still a need for w_e copies, but the customer is not willing to pay more than x_e for a copy of e (to add e to some bundle). We interpret x_e as a threshold price for product e , i.e., the customer has the possibility of purchasing e from an external source at a price of x_e .

Example 2.2.B. Continuing with Example 2.1.B, we consider that there is still a workload of w_e days on each $e \in E$. However, it is possible to work fewer days in e by paying a penalty of x_e for each non-working day. One can consider that x_e bears the cost of executing the workload of one day by external means.

The penalties \mathbf{x} are usually positive, but negative values can also be coherent—e.g., if the costs c_e are much smaller than the benefits $b_{e,e'}$ in Example 2.1.B. A first primal-dual relaxation of **Prize-Collecting Multi-Covering with Submodular Pricing** is a generalization of (2.2a)-(2.2b):

$$z_e : \begin{array}{l} \min \sum_e x_e \sigma_e + \sum_{S \subseteq E} \bar{f}(S) \pi_S \\ \sigma_e + \sum_{S \ni e} \pi_S = w_e \quad \forall e \in E \\ \sigma_e \geq 0 \quad \forall e \in E \\ \pi_S \geq 0 \quad \forall S \subseteq E \end{array} \quad (2.3a) \quad \left. \begin{array}{l} \max \mathbf{w}^\top \mathbf{z} \\ \sigma_e : z_e \leq x_e \\ \pi_S : \sum_{e \in S} z_e \leq \bar{f}(S) \quad \forall S \subseteq E \\ z_e \in \mathbb{R} \quad \forall e \in E \end{array} \right\} (2.3b)$$

We observe that similar (reversed) primal-dual programs arise in the SFM literature. To bring us closer to a SFM model, we will reformulate this model so as to replace the “ $\leq x_e$ ” constraint of (2.3b) with a “ ≤ 0 ”

constraint, as it is common in SFM. The reformulation will also reveal a natural justification for calling such problems “prize-collecting”.

Variable σ_e represents the number of copies of e left uncovered. If all elements would be left uncovered, a primal solution would be $\sigma_e = w_e, \forall e \in E$ and $\pi_S = 0, \forall S \subseteq E$; the (penalty) price to pay would be $\sum_{e \in E} w_e x_e$. By replacing σ_e with $w_e - \sum_{S \ni e} \pi_S$ in the objective function of (2.3a), the objective $\min \sum_e x_e \sigma_e + \sum_{S \subseteq E} \bar{f}(S) \pi_S$ becomes $\min \sum_e x_e w_e - \sum_e \sum_{S \ni e} x_e \pi_S + \sum_{S \subseteq E} \bar{f}(S) \pi_S = \sum_{e \in E} x_e w_e + \min \sum_{S \subseteq E} (\bar{f}(S) - \sum_{e \in S} x_e) \pi_S$, which can be written $\mathbf{x}^\top \mathbf{w} + \min \sum_{S \subseteq E} f(S) \pi_S$, where $f(S)$ is submodular function:

$$f(S) = \bar{f}(S) - \sum_{e \in S} x_e \quad (2.4)$$

Taking $\sum_{e \in E} w_e x_e$ as a *reference (penalty) price*, our objective is interpreted as follows: start from the reference price and minimize it by selecting subsets $S \subseteq E$; each such selection brings an objective function increase $\bar{f}(S)$ and a reduction $\sum_{e \in S} x_e$. This later term actually removes the penalty for letting e uncovered, *i.e.*, it is a “prize” for covering e . To keep a limit w_e on the number of cost-reduction prizes for $e \in E$, one imposes $\sum_{S \ni e} \pi_S \leq w_e$; implicitly, σ_e is $w_e - \sum_{S \ni e} \pi_S$ and it can never be negative. The primal-dual programs (2.3)-(2.3b) can be restated without primal variables σ :

$$\begin{aligned} \mathbf{w}^\top \mathbf{x} + \min \sum_{S \subseteq E} f(S) \pi_S & \quad \mathbf{w}^\top \mathbf{x} + \max \mathbf{w}^\top \mathbf{z} \\ z_e : \sum_{S \ni e} \pi_S \leq w_e \quad \forall e \in E & \quad \pi_S : \sum_{e \in S} z_e \leq f(S) \quad \forall S \subseteq E \\ \pi_S \geq 0 \quad \forall S \subseteq E & \quad z_e \leq 0 \quad \forall e \in E \end{aligned} \quad (2.5a) \quad (2.5b)$$

One could have obtained Program (2.5b) by performing substitution $\mathbf{z} \leftarrow \mathbf{z} - \mathbf{x}$ in (2.3b). We will often work on the submodular polytope $P(f) = \{z \in \mathbb{R}^n : \sum_{e \in S} z_e \leq f(S) \forall S \subseteq E\}$ already mentioned in (2.2b). This polytope is used by most SFM algorithms.

Finally, notice that similar primal-dual programs are used in the **Prize-Collecting Forest** (PCF) model from [17, § 3]. However, our solution method cannot be applied on this PCF model, as there is an essential difference: they use a (vertex) ground set V , but the PCF submodular function π acts on *sets of subsets* of V (*i.e.*, it is defined on 2^{2^V} instead of 2^V). Comprehensive comparisons with this form of submodularity are further provided in Section 4.1, as it often arises in **Submodular Cost Set Covering**.

3 The Proposed Solution Method

We first discuss solution methods for the simpler **Multi-Covering with Submodular Pricing** in Section 3.1. However, the main contributions of this section concern the more general **Prize-Collecting Multi-Covering** in Section 3.2. In Section 3.3, we perform a very fast sensitivity analysis based on the methods from Section 3.2.

3.1 Basic Multi-Covering with Submodular Pricing

3.1.1 A primal approach

The primal LP (2.2a) can be solved relatively straightforwardly using a decomposition on the covering demand values. We denote the minimum demand by $\bar{w}_0 = \min\{w_i : i \in [1..n]\}$. The fact that we need w_e copies of each element e allows us to separate the first \bar{w}_0 copies of each $e \in E$ from the rest $w_e - \bar{w}_0$. This way, we can incrementally construct the solution: after covering \bar{w}_0 copies of each $e \in E$, one can remove these demands and solve a residual problem (and all elements with $w_e = \bar{w}_0$ vanish). With regards to the these first \bar{w}_0 copies, it is better to cover them by selecting \bar{w}_0 times the full set E instead of selecting \bar{w}_0 layers of some partition of E (because $\bar{f}(E) \leq \sum_{i \in [1..m]} \bar{f}(E_i)$ for any partition $\{E_1, E_2, \dots, E_m\}$ of E). More formally, the primal problem could be solved by constructing an inclusion sequence of subsets $E_k \subsetneq E_{k-1} \subsetneq \dots \subsetneq E_1 \subsetneq E_0 = E$. We start with $E_0 = E$ and inductively construct $E_{\ell+1} \subsetneq E_\ell$ by removing the lowest-demand elements from E_ℓ , *i.e.*, $E_{\ell+1} = E_\ell - \{e \in E_\ell : w_e = \bar{w}_\ell\}$, where $\bar{w}_\ell = \min_{e \in E_\ell} w_e$.

By developing this construction, the optimal primal solution becomes $\pi_S^* = \bar{w}_\ell - \bar{w}_{\ell-1}$ if $S = E_\ell$ for some $\ell \in [0..k]$ (consider $\bar{w}_{-1} = 0$) and 0 otherwise. Since this optimal solution π^* of (2.2a) is integer, we actually solved both the integer and the fractional **Set-Covering with Submodular Pricing**. The optimal would still be integer if one allowed the subsets to be selected a fractional number of times.

The advantage of the above method is that it can simply compute the primal CG optimal solution π^* in $\mathcal{O}(n^2)$ time. Indeed, as long as the effective optimal objective value does *not* need to be calculated, the sets E_0, E_1, \dots, E_k can be determined without any *submodular function evaluation*. The very low complexity $\mathcal{O}(n^2)$ of this algorithm makes it more effective (both in practice and in theory) than any CG method that would need to call a SFM algorithm at each iteration.

3.1.2 A dual method

The above pure primal approach has certain limitations: without calculating dual values, it cannot be used for sensitivity analysis (as in Section 3.3.1), it cannot be generalized to other **covering** versions and it does not reveal in-depth combinatorial properties. We will see that a dual standpoint does merit serious consideration and we hereafter use it to address all remaining questions.

To determine the dual solution in programs (2.2a)-(2.2b), we need to investigate submodular polytopes. Recall that (2.2b) actually maximizes a linear objective $\mathbf{w}^\top \mathbf{z}$ over the submodular polyhedron $P(f)$. It is not difficult to check that if $\mathbf{w} \geq \mathbf{0}$, then the optimum solution belongs to the base polyhedron, *i.e.*, to the face $\sum_{e \in E} z_e = \bar{f}(E)$. This follows from the fact that if the “ $\leq \bar{f}(E)$ ” constraint of $P(\bar{f})$ is not tight, one can always increase some z_e —this can be verified using (2.1). However, an optimal solution to (2.2b) is generated by the well-known **Greedy** algorithm using a linear order \prec such that $e_1 \prec e_2 \prec \dots \prec e_n$, where $w_{e_1} \geq w_{e_2} \geq \dots \geq w_{e_n}$. Furthermore, **Greedy** also determines the optimal solution π^* in (2.2a): $\pi_S^* = w_{e_{i-1}} - w_{e_i}$ if $S = \{e_1, e_2, \dots, e_{i-1}\}$, $\pi_S^* = 0$ otherwise $\forall i = n, \dots, 2$, and $\pi_E^* = w_{e_n}$. It is not difficult to check this is the same solution as the one obtained by the primal approach above.

3.2 Prize-Collecting Multi-Covering

The objective of (2.5b) is $\mathbf{w}^\top \mathbf{x} + \max \mathbf{w}^\top \mathbf{z}$. By ignoring the reference (penalty) price $\mathbf{w}^\top \mathbf{x}$, we obtain the program below. One should keep in mind that this reference penalty will still be later used in the sensitivity analysis when we will investigate the effect of changing the demands \mathbf{w} (Section 3.3.1) or the penalties \mathbf{x} (Section 3.3.2.3).

$$\begin{aligned} & \max \mathbf{w}^\top \mathbf{z} \\ \pi_S : & \sum_{e \in S} z_e \leq f(S) \quad \forall S \subseteq E \\ & z_e \leq 0 \quad \forall e \in E \end{aligned} \tag{3.1}$$

Theorem 1. *The optimum of (3.1) can be determined only using a restricted (basic) family of submodular constraints $\sum_{e \in S_\ell} z_e \leq f(S_\ell)$, where S_ℓ belongs to an inclusion sequence $S_k \subsetneq S_{k-1} \dots \subsetneq S_0$. Each S_ℓ is a minimizer of f over all $S \subseteq E_\ell$, where E_ℓ belongs to an inclusion sequence $E_k \subsetneq E_{k-1} \subsetneq \dots \subsetneq E_0 = E$ such that $E_{\ell+1} = E_\ell - \{\bar{e}_\ell \in E_\ell : w_{\bar{e}_\ell} = \min_{e \in E_\ell} w_e\}, \forall \ell \in [0..k]$.*

Proof. Let us start with any optimal solution z^* of (3.1) and consider its family $\mathcal{T} \subseteq 2^E$ of z^* -tight sets, *i.e.*, $\mathcal{T} = \{S \subseteq E : \sum_{e \in S} z_e^* = f(S)\}$. Let us take $S_0 = \bigcup_{S \in \mathcal{T}} S$; since the union and intersection of z^* -tight sets is also z^* -tight (see [12, Lemma 2.3] for a proof), S_0 is also z^* -tight. Using $z_e^* \leq 0, \forall e \in S_0$, it is clear that S_0 is a minimiser of f over \mathcal{T} . We will further prove that S_0 is also a minimizer of f over 2^E . We now show that $z_e^* = 0, \forall e \in E_0 - S_0$. Assume $z_e^* < 0$ for some $e \in E_0 - S_0$: since such e cannot belong to any z^* -tight set, one could feasible increase z_e^* and thus increase the objective function value—impossible given the optimality of z^* . As such, S_0 is a maximal minimizer of f on ground set $E_0 = E$.

We generate by induction all sets E_0, E_1, \dots, E_k and S_0, S_1, \dots, S_k . Given constructed sets from $S_0, S_1, \dots, S_\ell, \ell \geq 0$ and E_0, E_1, \dots, E_ℓ , we inductively construct $S_{\ell+1}$. In fact, the construction of $E_{\ell+1}$ from E_ℓ is already specified in the hypothesis: remove from E_ℓ all elements \bar{e}_ℓ of minimum demand, *i.e.*, elements $\bar{e}_\ell \in E_\ell$ for which $w_{\bar{e}_\ell} = \bar{w}_\ell = \min_{e \in E_\ell} w_e$, equivalent to $E_{\ell+1} = \{e \in E_\ell : w_e > \bar{w}_\ell\}$.

Take $S_{\ell+1}$ as the union of all z^* -tight sets containing only elements of $E_{\ell+1}$.

$$S_{\ell+1} = \bigcup_{S \in \mathcal{T}, S \subseteq E_{\ell+1}} S \quad (*)$$

Let us prove that $z_e^* = 0, \forall e \in E_{\ell+1} - S_{\ell+1}$. Assume there is $e \in E_{\ell+1} - S_{\ell+1}$ with $z_e^* < 0$ and consider all z^* -tight sets that contain e . The intersection $S_e = \bigcap_{e \in S \in \mathcal{T}} S$ needs to be z^* -tight (see [12, Lemma 2.3]). This intersection S_e cannot contain only elements of $E_{\ell+1}$, because this would put e into $S_{\ell+1}$, using (*) above. Take any $e' \in S_e - E_{\ell+1}$: one can feasibly decrease $z_{e'}^*$ and increase z_e^* by the same sufficiently small $\epsilon > 0$ (since $e, e' \in S_e$, they always appear together in z^* -tight sets that contain e). Such change would increase the objective function value by $\epsilon(w_e - w_{e'})$, which is positive using: (1) $w_e > \bar{w}_\ell = \min_{e \in E_\ell} w_{e_\ell}$ from the construction of $E_{\ell+1}$ and (2) $\bar{w}_\ell \geq w_{e'}$ from the fact that $\bar{w}_\ell < w_{e'}$ would put e' into $E_{\ell+1}$.

This objective increase contradicts the optimality of z^* : the assumption $z_e^* < 0$ is false. In a flow push-relabel interpretation, one would say $e \rightarrow e'$ is a (demand-indexed) arc of strictly positive capacity over which one can perform a push that decreases net flow $z_{e'}^*$ in e' and increases z_e^* .

The above paragraph showed that $z_e^* = 0, \forall e \in E_{\ell+1} - S_{\ell+1}$; we can also use notation $z^*(E_{\ell+1} - S_{\ell+1}) = \sum_{e \in E_{\ell+1} - S_{\ell+1}} z_e^* = 0$. Since $S_{\ell+1}$ is the largest z^* -tight subset of $E_{\ell+1}$, $S_{\ell+1}$ is a (maximal) minimizer of f over all subsets of $E_{\ell+1}$. Additionally, $S_{\ell+1} \subseteq S_\ell$, because $S_{\ell+1}$ is constructed in (*) from z^* -tight sets $S \subseteq E^{\ell+1}$ that also belong to $E^\ell \supseteq E^{\ell+1}$; all these z^* -tight sets do arise in the S_ℓ construction using the same (*). The induction step is proved; one stops at $k = \ell$ when $E^{\ell+1} = \emptyset$.

We still have to determine the optimum objective function value $\text{OPT} = \mathbf{w}^\top \mathbf{z}^*$ based only on $E_k \subsetneq E_{k-1} \subsetneq \dots \subsetneq E_0 = E$ and $S_k \subseteq S_{k-1} \dots \subseteq S_0$ —see also (3.5) for an intuitive graphical representation of these inclusion sequences (Section 3.2.1). We write $\mathbf{w}^\top \mathbf{z}^* = \sum_{\ell=0}^{k-1} \sum_{e \in E_\ell - E_{\ell+1}} w_e z_e^*$, and let us focus on determining $\sum_{e \in E_\ell - E_{\ell+1}} w_e z_e^*$ for a fixed $\ell \in [0..k]$. We argued above that $z_e^* = 0, \forall e \in E_\ell - S_\ell$. However, it is not clear how exactly z^* is defined over $S_\ell - E_{\ell+1}$, but we will see this is not really necessary. Indeed, we only need to determine $\sum_{e \in S_\ell - E_{\ell+1}} w_e z_e^*$; since all $e \in S_\ell - E_{\ell+1}$ have the same demand value \bar{w}_ℓ , this sum can be written $\bar{w}_\ell z^*(S_\ell - E_{\ell+1})$. Using $z^*(E_{\ell+1}) = z^*(S_{\ell+1}) = f(S_{\ell+1})$, it is clear that $z^*(S_\ell - E_{\ell+1}) = z^*(S_\ell) - z^*(E_{\ell+1}) = f(S_\ell) - f(S_{\ell+1})$. The optimum objective function value becomes:

$$\text{OPT} = \mathbf{w}^\top \mathbf{z}^* = \sum_{\ell=0}^{\ell=k} \bar{w}_\ell (f(S_\ell) - f(S_{\ell+1})) = \bar{w}_0 f(S_0) + \sum_{\ell=1}^k (\bar{w}_\ell - \bar{w}_{\ell-1}) f(S_\ell) \quad (3.2)$$

□

We mention that an analogous result of this theorem can be found in [16, §44.2], where $\mathbf{w}^\top \mathbf{z}$ is maximized over $\{z \in \mathbb{R}^E : \sum_{e \in S} z_e \leq f(S)\}$. While this polytope is not the same as the one from (3.1), one can relate the two polytopes using the techniques from [16, 44.1]. Please let us present the consequences of this theorem.

The primal optimal solution π^* in (2.5a) follows from above (3.2), *i.e.*, using $\sum_{S \subseteq E} f(S) \pi_S^* = \text{OPT} = \bar{w}_0 f(S_0) + \sum_{\ell=1}^k (\bar{w}_\ell - \bar{w}_{\ell-1}) f(S_\ell)$, the optimal primal solution is: $\pi_{S_0}^* = \bar{w}_0, \pi_{S_\ell}^* = \bar{w}_\ell - \bar{w}_{\ell-1}, \forall \ell \in [1..k]$ and $\pi_S^* = 0, \forall S \in 2^E - \{S_0, S_1, \dots, S_k\}$.

A Strong Map Sequence of functions can be constructed from the inclusion sequence $E_k \subsetneq E_{k-1} \dots \subsetneq E_0$. We define the following family of E_ℓ -kernel functions ($\forall \ell \in [0..k]$):

$$f^\ell(S) = f(S \cap E_\ell) + \sum_{e \in S - E_\ell} (|f(e)| + 1), \quad (3.3)$$

As a sum between a submodular function and a unimodular one, f^ℓ is submodular. One can check that if $f(S_\ell) = \min_{S \subseteq E_\ell} f(S)$, then $f(S_\ell) = f^\ell(S_\ell) = \min_{S \subseteq E} f^\ell(S)$. The strict positivity of $|f(e)| + 1$ only ensure that all minimisers of f^ℓ belong to E_ℓ . Furthermore, we show that the following is a strong map sequence of submodular functions :

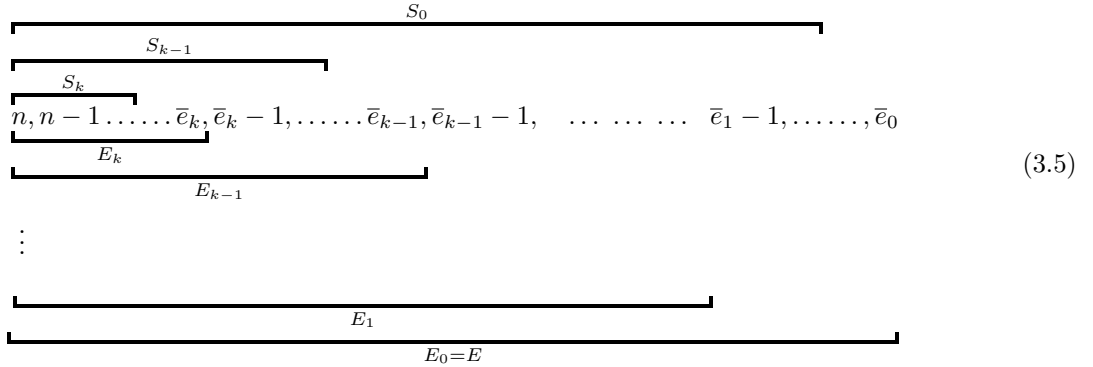
$$f^k \rightarrow f^{k-1} \rightarrow \dots \rightarrow f^1 \rightarrow f^0 = f \quad (3.4)$$

First, recall that two functions f and \hat{f} are in a strong-map relation $f \rightarrow \hat{f}$ if and only if $f(S \cup T) - f(S) \geq \hat{f}(S \cup T) - \hat{f}(S), \forall S, T \subseteq E$. The proof goes on by induction on T , considering any fixed $S \subseteq E$. Starting with $T = \emptyset$, the strong map inequality $f^\ell(S \cup T) - f^\ell(S) \geq f^{\ell-1}(S \cup T) - f^{\ell-1}(S)$ is clearly verified. We show it is still verified after substituting $T \leftarrow T \cup \{e\}, \forall e \in E - (S \cup T)$. Since e can belong to any of $E_\ell \subsetneq E_{\ell-1} \subseteq E$, there are three cases depending on its position: (i) if $e \in E - E_{\ell-1}$, both sides increase by $|f(e)| + 1$; (ii) if $e \in E_{\ell-1} - E_\ell$, the left side increases by $|f(e)| + 1$ and the right side by $f((E_{\ell-1} \cap (S \cup T)) \cup \{e\}) - f(E_{\ell-1} \cap (S \cup T))$, and (iii) if $e \in E_\ell$, the left side increases by $f((E_\ell \cap (S \cup T)) \cup \{e\}) - f(E_\ell \cap (S \cup T))$ and right side by $f((E_{\ell-1} \cap (S \cup T)) \cup \{e\}) - f(E_{\ell-1} \cap (S \cup T))$. Both cases (ii) and (iii) produces left increases larger than or equal to the right one, by virtue of (2.1).

The Push-Relabel Fleischer-Iwata framework [6] is tailored especially for strong map sequences of submodular functions. In fact, we showed that the proposed problem can be solved by minimizing the strong map sequence (3.4). The Fleischer-Iwata framework performs this task in $\mathcal{O}((n^7 + kn^2)\gamma + n^8)$ asymptotic running time. Since constructing all E_ℓ and f_ℓ requires less than $\mathcal{O}(n^8)$ and k is asymptotically smaller than $\mathcal{O}(n^5)$, the total asymptotic running time is $\mathcal{O}(n^7\gamma + n^8)$.

3.2.1 Combinatorial Structure of Optimal z^*

A graphical representation of the inclusion sequences can further clarify the situation. Without restricting generality, we consider that $E = \{1, 2, \dots, n\}$ is indexed such that $w_n \geq w_{n-1} \dots \geq w_1$. This series actually contains $k+1$ distinct demand values $\bar{w}_k > \bar{w}_{k-1} \dots \geq \bar{w}_0$. The last element with demand \bar{w}_ℓ is here denoted $\bar{e}_\ell, \forall \ell \in [0..k]$. All elements from $\bar{e}_\ell - 1$ to $\bar{e}_{\ell-1}$ have demand $\bar{w}_{\ell-1}$ and they form set $E_{\ell-1} - E_\ell$, see (3.5) below.



The value of optimal z^* over the whole E can be constructed by a **Greedy** approach on a restricted set of variables. The proof of Theorem 1 only shows that $z^*_e = 0, \forall e \in E_\ell - S_\ell, \forall \ell \in [0..k]$ and let us fix these values. To determine z^*_e on all other $e \in \bigcup_{\ell \in [1..k]} S_\ell - E_{\ell+1}$, we can apply a **Greedy** approach based on a linear order $e_1 \prec e_2 \prec e_3 \dots e_n$ that is consistent with:

$$S_k \prec S_{k-1} - E_k \prec \dots \prec S_1 - E_2 \prec S_0 - E_1$$

A solution z^\prec generated by **Pure Greedy** can have strictly positive values $z^\prec_{e_i}$, as calculated by the **Pure Greedy** formula $z^\prec_{e_i} = f(\{e_1, e_2, \dots, e_i\}) - f(\{e_1, e_2, \dots, e_{i-1}\})$. To overcome this, we can simply change the **Greedy** formula to $z^*_{e_i} \leftarrow z^-(z^\prec_{e_i}) = \min(z^\prec_{e_i}, 0)$. However, this can only be used to determine *one* optimal solution. All optimal solutions can be generated using the classical base polyhedron SFM approach: remove negativity constraints in (3.1), impose $\sum_{e \in E} z_e = f(E)$ and replace the objective with $\sum_{e \in E} w_e z^-(z_e)$, so as to construct an equivalent base polyhedron model.³ Any solution of (3.1) can be seen as a projection (via z^-) of a solution in the base polyhedron, using [12, Lemma 2.4]. As such, any optimal solution can be generated by projecting a linear combination of solutions constructed by **Pure Greedy** on the base polyhedron.

A Flow Network enriched with Demand-Indexed Arcs can be associated to the base polyhedron formulation, generalizing classical SFM approaches. For instance, push-relabel algorithms [15, 6] define arcs $e \rightarrow e'$

³The base polyhedron approach is a classical technique in SFM, see Section 4.2 for references and (4.1) for the exact mathematical model.

for any $e, e' \in E$ with $z_e < 0$ and $z_{e'} > 0$ (we use the arc directions from [12], *i.e.*, $e \rightarrow e'$ could stand for a path from k_1 to k_4 in [12, Fig. 1]) and they are still valid. Indeed, the **Greedy** solution of an order \prec such that $e' \prec e$ would not get worse by reversing e and e' : this amounts to a non-strict increase of $z_e < 0$ and a non-strict decrease of $z_{e'} > 0$ (*e.g.*, by a sufficiently small value), which can only increase $z^-(z_e) + z^-(z_{e'})$. This way, one tries to find orders \prec^* satisfying $e \prec^* e'$ for e and e' as above, so as to move strictly positive components to the right. Our model generalizes this by introducing **demand-indexed** arcs $e \rightarrow e'$ even for $z_e, z_{e'} < 0$: such an arc can be constructed when $w_e > w_{e'}$, *e.g.*, when $e \in S_{l+1} - E_{l+2}$ and $e' \in S_l - E_{l+1}$ (in this case, $w_e \geq \bar{w}_{l+1} > \bar{w}_l \geq w_{e'}$). As above, a **Greedy** solution of an order \prec such that $e' \prec e$ would not get worse by reversing e and e' : this can only increase z_e and decrease $z_{e'}$ at least by the same sufficiently small value ϵ , yielding an increase of the objective by $\epsilon(w_e - w_{e'})$, as in one argument in the proof of Theorem 1.

3.3 Fast Sensitivity Analysis using Optimal z^* and the Push-Relabel Framework

Since the input parameters (*e.g.*, demand w_e , penalty x_e ($\forall e \in E$), price function \bar{f} , etc) might represent *managerial decisions*, the choice of the best parameter values might be the main issue to be studied. For such purposes, we propose two sensitivity analysis methods: (i) Section 3.3.1 below, in which we make use of some primal-dual combinatorial properties studied in Section 3.2.1, and (ii) Section 3.3.2 that exploits the Fleischer-Iwata Framework (FIF) [6, 15].

While the FIF is not necessarily the fastest solution method for our problem (applying Orlin Algorithm [14] $\mathcal{O}(n)$ times requires less asymptotic running time), it is very fast and well-suited for sensitivity analysis, because it can simultaneously minimize up to $\mathcal{O}(n^5)$ functions in strong map relations. If some parameter changes can be anticipated and formalized using strong map relations (Section 3.3.2), the FIF can determine in advance updated optimal solutions for any of up to $\mathcal{O}(n^5)$ possible parameter changes—in some cases (Section 3.3.2.1), this comes at no extra computational complexity cost.

3.3.1 Sensitivity Analysis Using the $\pi^* - z^*$ Duality

We here focus on the effects of demand changes (modifications of \mathbf{w}). Some natural questions might be: how can one rapidly update the optimal solution if product i is no longer needed by the customer (in Example 2.1.A) ? What happens if certain locations in Example 2.1.B are required to perform some extra work? Instead of re-optimizing after any such demand change, the optimum value can be more rapidly updated by exploiting the value of an optimum (CG dual) z^* .

In classical LP sensitivity analysis [7, Ch. 6], one interprets z_e^* as a shadow price describing the instantaneous change, per unit of constraint $\sum_{S \ni e} \pi_S \leq w_e$, in the optimum objective value obtained by relaxing this constraint in (2.5a). Given the high degeneracy of (2.5a), we will actually define and use $z_{\min}^*(e)$ and $z_{\max}^*(e)$, the minimum and respectively maximum value of z_e^* over all optimal z^* , that can be found using the method from Section 3.2.1. If $e \in E_\ell - S_\ell$ for some $l \in [1..k]$, then $z_{\max}^*(e) = z_{\min}^*(e) = 0$. Otherwise, $e \in S_\ell - E_{\ell+1}$ for some $\ell \in [0..k]$ (using $E^{k+1} = \emptyset$). Using (2.1) on (3.5), the minimum (respectively maximum) z_e^* is realized if order \prec puts e last (respectively, first) in $S_\ell - E_{\ell+1}$. One can check in (3.5) that: (i) the first case puts e at the last position of S_ℓ and, (ii) the second case puts e right after all elements of $E_{\ell+1}$. Case (i) leads to $z_{\min}^*(e) = f(S_\ell) - f(S_\ell - \{e\})$ and case (ii) to $z_{\max}^*(e) = f(E_{\ell+1} \cup \{e\}) - f(E_{\ell+1})$.

The demand changes do not modify the feasibility of an optimal z^* in (3.1) or (2.5b). If one performs substitution $w_e \leftarrow w_e + \alpha$ (with $\alpha > 0$), the reference penalty $\mathbf{w}^\top \mathbf{x}$ is increased by αx_e (α penalties more) and $\mathbf{w}^\top \mathbf{z}^*$ is increased by αz_e^* . As such, substitution $w_e \leftarrow w_e + \alpha$ leads to an objective increase of at least $\alpha(x_e + z_{\max}^*(e))$. This simply results from calculating the objective function change induced on a solution z^* such that $z_e^* = z_{\max}^*(e)$. Conversely, a demand decrease $w_e \leftarrow w_e - \alpha$ leads to a total objective increase of $\alpha(-x_e - z_{\min}^*(e))$ or even more according to the structural changes triggered on the sets E_1, E_2, \dots, E_k . However, if any of the above minimum objectives increases is greater than 0, the demand changes surely increases the total objective value. Perhaps rather surprisingly at a first glance, by increasing the demand, the total price can sometimes decrease: if $f(|S|) = 0, \forall S \subsetneq E$ and $f(E) = -M$, the substitution $w_e \rightarrow \max_{e' \in E} w_{e'}, \forall e \in E$ reduces the cost to very small values—considering M very large.

3.3.2 Sensitivity Analysis using the Fleischer-Iwata Framework

If certain parameter changes could be formulated by extending the strong map sequence (3.4), it is possible to anticipate them at very limited complexity costs. The Fleischer-Iwata Framework can determine in advance certain updated optimal solutions. The total complexity is either multiplied by one n factor (*i.e.*, $\mathcal{O}(n^8\gamma + n^9)$) or even unchanged in some cases (§ 3.3.2.1), and this, even for extended sequences of length up to $\mathcal{O}(n^5)$.

3.3.2.1 Solving the problem simultaneously for several demand vectors Take linear order $n \prec n-1 \prec n-2 \prec \dots \prec 1$ that is consistent with the inclusion sequence $E_k \subsetneq E_{k-1} \dots E_0 = E$. Consider now that demand vector \mathbf{w} is changed to \mathbf{w}' ; the new vector generates its own inclusion sequence $E'_{k'} \subsetneq E'_{k'-1} \dots E'_0 = E$ that satisfies the following property: $\forall \ell \in [0..k], \ell' \in [0..k']$, either $E_\ell \subseteq E'_{\ell'}$ or $E_\ell \supseteq E'_{\ell'}$. This ensures the existence of a mixed inclusion sequence, for instance:

$$\begin{aligned} E'_{k'} \subsetneq E'_{k'-1} \cdots \subsetneq E'_{k'-j_1+1} \subseteq E_k \subseteq \\ E'_{k'-j_1} \cdots \subsetneq E'_{k'-j_2+1} \subseteq E_{k-1} \subseteq \\ \dots \\ E'_{k'-j_k} \cdots \subsetneq E'_0 \subseteq E_0 = \\ E \end{aligned}$$

One can easily generalize the strong map sequence $f^k \rightarrow f^{k-1} \rightarrow \dots f^0$ from (3.4): use (3.3) to define new $E'_{\ell'}$ -kernel functions for all new $E'_{\ell'}$ subsets above. These new kernel functions form a strong map sequence that include (3.4). As such, the Fleischer-Iwata framework requires the same asymptotic running time $\mathcal{O}(n^7\gamma + n^8)$ to provide the minimizer of f over all subsets of $E'_{\ell'}$ and E_ℓ (with $\ell' \in [0..k']$ and $\ell \in [0..k]$), and so, solve the problem simultaneously for both \mathbf{w} and \mathbf{w}' .

Let us finish with a simple example. An increment-only demand change $\mathbf{w} \rightarrow \mathbf{w}'$ (*i.e.*, $w'_e - w_e \in \{0, 1\}$, $\forall e \in E$) does generate a valid mixed inclusion sequence. For a fixed $\ell \in [1..k]$, one needs to use a linear order \prec that places all $e \in E_\ell$ with $w'_e = w_e + 1$ before all other $e \in E_\ell$. The incremented elements lead to new subsets $E'_{\ell'}$, embedded between $E_{\ell+1}$ and E_ℓ ($E_{\ell+1} \subseteq E'_{\ell'} \subseteq E_\ell$). This idea can be generalized: instead of simple increments, consider any larger variations only within $[0, \bar{w}_{\ell+1} - \bar{w}_\ell]$, or only within $[-(\bar{w}_\ell - \bar{w}_{\ell-1}), 0]$.

3.3.2.2 Pricing functions in strong map relations Possible changes in the pricing function can be anticipated and formulated as strong map sequences of the form $f_m \rightarrow f_{m-1} \rightarrow \dots f_0 = f = f_{-1} \rightarrow f_{-2} \rightarrow \dots f_{-p}$. A E_ℓ -kernel function f_i^ℓ can be directly defined by applying (3.3) from Section 3.1 on f_i , $\forall i \in \{m, m-1, \dots, -p+1, -p\}$. These E_ℓ -kernel functions also form a strong map sequence $f_m^\ell \rightarrow f_{m-1}^\ell \rightarrow \dots f_0^\ell = f^\ell = f_{-1}^\ell \rightarrow f_{-2}^\ell \rightarrow \dots f_{-p}^\ell$. For each $\ell \in [0..k]$, this sequence can be optimized by the parametric Fleischer-Iwata framework of complexity $\mathcal{O}(n^7\gamma + n^8)$. This complexity bound holds as long $p+m \in \mathcal{O}(n^5)$, because each of the $m+p+1$ functions f_i^ℓ induces an additional computational overhead of $\mathcal{O}((m+p)n^2\gamma)$ [6, § 3]. By considering all $k+1$ possible values of ℓ , the total complexity is $\mathcal{O}(k(n^7\gamma + n^9)) = \mathcal{O}(n^8\gamma + n^9)$.

In fact, the complexity at each value of ℓ can be refined to $\mathcal{O}(|E_\ell|^{7\gamma} + |E_\ell|^8)$, because only $|E_\ell|$ elements matter at level ℓ . This way, the total complexity is $\sum_{\ell=0}^k \mathcal{O}(|E_\ell|^{7\gamma} + |E_\ell|^8)$: if $|E_\ell| \leq \frac{n}{(\ell+1)^{1/6}}$, this complexity sum is dominated by $\mathcal{O}(n^7\gamma) \sum_{\ell=0}^{\infty} \frac{1}{(\ell+1)^{7/6}} + \mathcal{O}(n^8) \sum_{\ell=0}^{\infty} \frac{1}{(\ell+1)^{8/6}} = \mathcal{O}(n^7\gamma + n^8)$. We used the fact that the hyperharmonic series $\sum_{\ell=1}^{\infty} \frac{1}{n^s}$ converges to the Riemann zeta function $\zeta(s)$ if $s > 1$.

The example below shows that the proposed $\mathcal{O}(n^8\gamma + n^9)$ solution method is not only useful for sensitivity analyses, but also for parametric optimization associated to a strong map sequence of functions.

Example 3.3.2.B *Continuing with Examples 2.1.B and 2.2.B, we consider a family of subsets $\mathcal{B} = \{B_1, B_2, \dots, B_n\} \subseteq 2^E$; we note b_i the benefit resulting from simultaneous work in B_i . A submodular pricing*

function can be defined with a similar formula as in Example 2.1.B: $\bar{f}_{\mathcal{B}}(S) = \sum_{e \in S} c_e - \sum_{\substack{B_i \in \mathcal{B} \\ B_i \subseteq S}} b_i$.

We now consider a scenario in which the benefit sets \mathcal{B} cannot be easily identified (due to absence of information, or, why not, poor understanding). However, suppose one can indicate (anticipate) a linear order $B_1 \prec B_2 \prec \dots \prec B_{n'}$: if $i \prec j$, then it is easier to obtain benefits from simultaneous work in B_i than in B_j . Denoting $\prec(B_i) = \{B_1, B_2, \dots, B_i\}$, $\forall i \in [1..n']$, the following functions form a strong map sequence:

$$f_{\prec(B_1)} \rightarrow f_{\prec(B_2)} \rightarrow f_{\prec(B_3)} \rightarrow \dots \rightarrow f_{\prec(B_{n'})}.$$

As long as $n' \in \mathcal{O}(n^5)$, the Fleischer-Iwata framework can simultaneously find the optimal solution for all above functions within $\mathcal{O}(k(n^7\gamma + n^8)) = \mathcal{O}(n^8\gamma + n^9)$ asymptotic running time. To the best of our knowledge, this is not possible with other FSM algorithms. A more naive approach would require applying the fastest available SFM optimizer up to nn' times.

3.3.2.3 Changing the Non-Covering Penalties \mathbf{x} First, recall that x_e can be interpreted as a threshold pricepoint for e , *e.g.*, in Example 2.2.A, the customer is not willing to pay more than x_e to add article e to an existing bundle. Reasonable questions might be: what happens if the threshold pricepoint x_e increases or decreases for some $e \in E$? Can one anticipate such variations and construct in advance alternatives solutions with no computational overhead? What is the return on the investment necessary for buying an asset that could generate any $e \in E$ at some price $x_e^1 \leq x_e$?

A substitution $x_e \leftarrow x_e^1$, $\forall e \in E$ would first modify the reference penalty $\mathbf{w}^\top \mathbf{x}$. The submodular pricing function $f(S) = \bar{f}(S) - \sum_{e \in S} x_e$ (see (2.4)) is replaced with $f_1(S) = f(S) + \sum_{e \in S} x_e - x_e^1$. Function f_1 only differs from f by some unimodular increases $x_e - x_e^1 \geq 0$ on $e \in S$; this can be used to prove the strong map relation $f_1 \rightarrow f$. One can further consider more penalty reductions by formulating a gradual sequence $\mathbf{x} > \mathbf{x}^1 > \mathbf{x}^2 > \dots > \mathbf{x}^m$. For instance, the first elements from this series might represent penalty reductions on a few $e \in E$, followed by reductions on increasingly larger subsets of E towards the end of the series.

Such sequences of price reductions generate strong map relations $f_m \rightarrow f_{m-1} \rightarrow \dots \rightarrow f_1 \rightarrow f$, generalizing $f_1 \rightarrow f$. Using the approach described just above (§ 3.3.2.2), the minimizers of all such functions can be generated in $\mathcal{O}(n^8\gamma + n^9)$ time, as long as $m \in \mathcal{O}(n^5)$.

4 Related Problems and Ideas

4.1 Submodular Cost Set Covering

Given ground set (universe) E and a family $N \subseteq 2^E$ of subsets of E , **Submodular Cost Set Cover** [9] is a problem defined using a function $f_N : 2^N \rightarrow \mathbb{R}$ that is submodular *over the power set of N* ; the goal is to find $S_N \subseteq N$ such that $f_N(S_N)$ is minimized and $\bigcup_{S \in S_N} S = E$. We obtain **Submodular Edge Cover** by simply considering (E, N) as a graph: N contains only subsets of E with two elements. It is useful to interpret **Submodular Cost Set Cover** on a bipartite graph: consider vertices $N \cup E$ and add arcs from N to E such that $S \rightarrow e$ is an arc $\Leftrightarrow S \ni e$ (with $S \in N$ and $e \in E$). The objective is to select some $S_N \subseteq N$ that is incident to all $e \in E$ and that minimizes $f_N(S_N)$. Certain papers also introduce a polymatroid rank function $f_{\text{cover}} : 2^N \rightarrow \mathbb{Z}_+$ such that $f_{\text{cover}}(S_N)$ is the number of vertices $e \in E$ incident to S_N , see [11, Ex. 1.4]; the covering condition can be expressed $f_{\text{cover}}(S_N) = f_{\text{cover}}(N)$.

The submodularity of $f_N : 2^N \rightarrow \mathbb{R}$ does not imply submodularity over the power set of E . Let us prove this with a simple example: $E = \{1, 2, 3\}$ and $N = \{A, B, C, D\}$, where $A = \{1\}, B = \{2\}, C = \{2, 3\}, D = \{3\}$; define $f_N(S_N) = \sum_{S \in S_N} g_S, \forall S_N \subseteq N$, where $g_A = g_B = g_D = 1$ and $g_C = 3$. If one would try to extend this definition to a CG setting with submodular pricing over the power set of E , the four columns below could be obtained.

$$\begin{array}{cccc} f_N(\{A\})=1 & f_N(\{A, B\})=2 & f_N(\{A, D\})=2 & f_N(\{A, C\})=4 \\ \hline 1 & 1 & 1 & 1 \\ & 2 & & 2 \\ & & 3 & 3 \end{array}$$

The above column pricing is not submodular over the power set of $\{1, 2, 3\}$: there is no diminishing marginal pricing as in (2.1). The marginal value of $\{2\}$ is increasing from $1 = f_N(\{A, B\}) - f_N(\{A\})$ (*i.e.*, add $\{2\}$ to $\{1\}$, see second column compared to the first) to $2 = f_N(\{A, C\}) - f_N(\{A, D\})$ (add $\{2\}$ to $\{1, 3\}$, see last column compared to the third).

Submodular Cost Set Cover is also more complex in the sense that the number of variables of the associated LP can be much larger. If $|N| \in \mathcal{O}(2^n)$, the (natural) use of one variable per subset of N leads to a doubly-exponential number of variables $\mathcal{O}(2^{2^n})$ —consider, for instance [9, § 4.4] with a large N . This phenomenon is also discussed in the first paragraph of [17, § 6]; their starting program (called (PCF-IP), p. 2, § 1) is similar to (2.3a), but their submodular function π has domain 2^{2^V} , where V is a ground set.

However, **Set-Covering with Submodular Pricing** can be seen as a particular case of **Submodular Cost Set Cover** as follows: (i) set $N = \{\{e_1\}, \{e_2\}, \dots, \{e_n\}\}$ and (ii) choose an appropriate definition for f_N , *i.e.*, $f_N(S_N) = f(\bigcup_{S \in S_N} S)$, $\forall S_N \subseteq N$. Due to these particularizations, our primal CG LPs (*e.g.*, (2.2a) or (2.5a)) have SFM polytopes as dual LPs (*e.g.*, (2.2b) or (2.5b)); this allowed us to use **Greedy** to show that these LPs have integer primal optimal solutions (when the demands \mathbf{w} are integer). On the other hand, “Submodular Cost” covering problems have similar LPs with *fractional* optimal solutions; they are often tackled by rounding methods, leading to approximation results [9].

4.2 Convex Minimization on the Base Polyhedron

A classical approach in SFM consists of restating (3.1) as an optimization problem over the base polyhedron. Using the arguments from [12, § 2.4], one can propose the following equivalent formulation:

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e z^-(z_e) \\ \sum_{e \in S} z_e & \leq f(S) \quad \forall S \subsetneq E \\ \sum_{e \in E} z_e & = f(E) \quad , \end{aligned} \tag{4.1}$$

where $z^-(z_e) = \min(z_e, 0)$. The linear constraints define the base polyhedron, *i.e.*, it is the submodular polyhedron facet that verifies with equality the constraint associated to E .

We can say that the objective is to minimize $\sum_{e \in E} -g_e(z_e)$ over the base polyhedron, where $g_e(z_e) = w_e z_e$ if $z_e \leq 0$ and $g_e(z_e) = 0$ if $z_e > 0$ (for all $e \in E$). Since $-g_e$ is convex, this is a convex minimization problem over a base polyhedron. Such problems have been already studied in the literature [13]. However, to the best of our knowledge, the convex optimization approaches do not yet solve programs like (4.1) for the following reasons: (i) differentiable functions are usually used (above $-g_e$ is not differentiable in 0) and (ii) strict convexity is usually required [13]. Regarding (ii), $-g_e$ is not strictly convex because it does not verify $-g(z' + z'') > -g(z') + -g(z'')$, $\forall z', z'' \in \mathbb{R}$ (linear functions do not satisfy strict convexity).

4.3 Applications Well-Suited for the Proposed Approach

4.3.1 Combinatorial Auctions and Supermodular Function Maximization

Column Generation (CG) is a common technique for solving Combinatorial Auction Problem (CAP) models with prohibitively many columns [4, § 3.5]. In this context, program (2.5a) is interpreted as in Example 2.2.A, but with an objective representing the seller (auctioneer): E is a set of articles offered in an auction, a subset $S \subseteq E$ is a bundle that can be sold to bidders. The auctioneer has w_e copies of each article $e \in E$; e can either be sold in a bundle $S \ni e$ or be kept to be sold later at a threshold price x_e . The optimum of (2.5a) is actually the minimum profit that the auctioneer can obtain by selling all articles.

The relevant goal of an auctioneer is profit maximization. For this purpose, we transform the model: negate the objective function coefficients and the minimization objective becomes a maximization one, but bundle prices become $-f(S)$ ($\forall S \subseteq E$); the threshold prices become $-x_e, \forall e \in E$. If $-f$ is *supermodular*, f is submodular; by minimizing $\mathbf{w}^\top \mathbf{x} + \min \sum_{S \subseteq E} f(S) \pi_S$ in (2.5a), one actually maximizes $-\mathbf{w}^\top \mathbf{x} + \min \sum_{S \subseteq E} -f(S) \pi_S$. As such, one actually solves the Combinatorial Auction Problem (CAP) with *supermodular bundle pricing*: given supermodular bid function $-f : 2^E \rightarrow \mathbb{R}$, select the bids that maximize the total sales profit.

While submodularity is a more popular pricing method in well-known package deals (with diminishing marginal prices), *supermodular bundle pricing* is also commonly used in game theory or economics to model *bundle synergy* [19]. For instance, in finance, a diversified portfolio (bundle) is more valuable (less risky) than the total value of its constituent assets. In art, a large (complete) collection is worth more than the sum of its constituent articles. Generally speaking, bundle synergy captures the idea that the whole is worth more than the sum of its parts, see interesting examples in [19]. As such, CAP bid functions are often super-additive [4]: a bid on $S \cup S'$ needs to be more profitable for the auctioneer than the sum of the bids on S and S' —otherwise, the auctioneer ignores bid on $S \cup S'$, considering it less profitable than the separate bids on S and S' . Supermodularity is a stronger condition than super-additivity, but still reasonable in auctions: given article $e \in E$, it is natural for a collector to bid more on some new article e to extend a rich collection S than to extend an incomplete collection $S' \subsetneq S$. This translates into increasing marginal prices: the bid function $-f$ satisfies $-f(S \cup \{e\}) - (-f(S)) \geq -f(S' \cup \{e\}) - (-f(S'))$, $\forall S' \subseteq S \subsetneq S \cup \{e\} \subseteq E$, a sufficient supermodularity condition, using (2.1).

Let us investigate a classical formulation (CAP1 in [4, § 3.1]) in which there is a set N of bidders with bid functions $-f_j$ ($\forall j \in N$). After notational translations and some simplifications, this CAP1 model can be written as a generalization of (2.5a), using variable π_S^j to indicate that bundle S is sold to bidder $j \in N$.

$$\begin{aligned} \min \sum_{j \in N} \sum_{S \subseteq E} f_j(S) \pi_S^j &= \max \sum_{j \in N} \sum_{S \subseteq E} -f_j(S) \pi_S^j \\ \sum_{j \in N} \sum_{S \ni e} \pi_S^j &\leq w_e \quad \forall e \in E \\ \pi_S^j &\in \mathbb{Z}_+ \quad \forall S \subseteq E, j \in N \end{aligned} \tag{4.2}$$

For such models, CG is even more well-suited to iterative auctions [4, § 4.3]. In iterative auctions, the dual values z_e are used by the bidders to update their functions and propose better columns. However, the above LP is already more complicated than (2.5a), because its dual LP (generalizing (2.5b)) actually requires optimizing over the intersection of $|N|$ sub-modular polytopes. The dual of CAP1 leads in [4, § 3.4.5], under super-additivity and non-decreasing conditions, to an instance of the polymatroid intersection problem, which is polynomially solvable. In light of these observations it would be interesting to find whether our approach could be modified to optimize models like (4.2), but this is a subject for later research.

4.3.2 From SFM to Set-Covering Models: A Multi-Level Location-Supply Example

In general, most submodular function $\bar{f} : 2^E \rightarrow \mathbb{R}$ can generate a **Prize Collecting Multi-Covering with Submodular Pricing** application. For this, we can usually apply a *multi-level extension* that constructs program (2.3a) by assigning to each $e \in E$ a number of levels (separate covering requests) w_e : each selection of $S \ni e$ covers a separate level of e . In fact, it is this technique that generated Examples 2.1.A and 2.1.B: the levels w_e represent copies of products in the former case and working days in the latter; in this section, w_e will represent a number of available quality levels. We will also exemplify a *minimization reformulation technique* to ensure that \bar{f} is an appropriate function for minimization, *i.e.*, even if function \bar{f} below does not initially have a coherent cost minimization interpretation, a bi-level modelling can make it well-suited to the objective of (2.3a). These two techniques are presented on a generalization of a **Simple Plant Location Problem** [10, eq. (6)]; in fact, we present it using an equivalent warehouse terminology, inspired by the submodular function (noted \bar{f}) from [12, ex. 1.4]. CG models with more complex submodular pricing functions already exist in the location-inventory literature [18, Appendix B].

We consider a community composed of a set of retail stores R and a set E of potential locations for warehouses. An external warehouse operator is hired to distribute supplies from warehouses to retail stores. We propose a bi-level modelling, with two types of decisions:

- open a warehouse at $e \in E$: this yields a benefit ϕ_e for the community (more economic activity in e , less unemployment) and a cost of $\alpha\phi_e$ for the external operator. This level of decisions are taken by the community;
- serve retail store r from e : this represents a price b_{r_e} to be payed by the community and a benefit βb_{r_e} to be realized by the operator ($\beta > 0$), *i.e.*, a part of this benefit βb_{r_e} is realized by making retail store

r pay a proportional price b_{re} . We consider that the decision of serving retail store r from e rests only with the external operator. As such, the operator naturally maximizes its benefit for serving $r \in R$, *i.e.*, the benefit for serving r is $\beta \max_{e \in S} b_{re}$, where $S \subseteq E$ is the set of open warehouses (decision taken above).

If the community decides to open warehouses $S \subseteq E$, the operator has a benefit of $\bar{f}_{\alpha,\beta}(S) = -\alpha \sum_{e \in S} \phi_e + \beta \sum_{r \in R} \max_{e \in S} b_{re}$ and the community has to pay a total price of $\bar{f}(S) = -\sum_{e \in S} \phi_e + \sum_{r \in R} \max_{e \in S} b_{re}$. It is possible to have a benefit both for the community ($\bar{f}(S) < 0$) and for the operator ($\bar{f}_{\alpha,\beta}(S) > 0$). However, a natural objective for the community is to choose to open warehouses S for which $\bar{f}(S)$ is minimized; \bar{f} is actually the submodular function [12, Ex. 4], but the objective is now cost minimization instead of benefit maximization. Without a bi-level modelling, it is not coherent to minimize \bar{f} while using cost terms like $\max_{e \in S} b_{re}$.

To get closer to program (2.3a), the *multi-level extension* associates to each $e \in E$ a maximum level w_e . This could be seen as a maximum quality level of the produced articles (*e.g.*, in equivalent **Plant Location Interpretation**) or of the available storage capacities: $w_e = 1$ when only storage of non-perishable items is available in e , $w_e = 2$ for storage of (long-lasting) food, $w_e = 3$ for refrigerated storage, $w_e = 4$ for frozen storage, etc. We consider that if $e \in E$ is able to work at level w_e , e can also work at lower levels, *e.g.*, a frozen warehouse can also be used as a refrigerated warehouse. For any $e \in E$ with w_e available levels, a solution that only uses only $w_i - \sigma_e$ levels is penalized by $x_e \sigma_e$, *i.e.*, unused capacities or levels. For the community, the objective is to minimize the cost; the full program can be written exactly as (2.3a).

5 Conclusions

We presented a solution method based on SFM for **Prize-Collecting Multi-Covering with Submodular Pricing**. The practical relevance of this class of problems is underlined by revealing applications in rather unexpected domains, ranging from production planning and warehouse location to combinatorial auctions (see Examples 2.2.A-2.2.B or Section 4.3). The problem can be formulated using an LP with 2^n variables (columns); this LP can be optimized using a Column Generation (CG) model with a submodular pricing sub-problem. This type of CG algorithms was already applied for a location-inventory problem [18] and we hope that our alternative algorithm sheds useful light into the topic.

Our solution method resides in the fact that, after an appropriate reformulation (Section 2.2), the above CG LP has a dual LP that is thoroughly studied as a primal program in SFM [12, § 2.4]. This, in turn, shows that the optimal solution of the CG (fractional) LP is always integer; it can be constructed within $\mathcal{O}(n^7\gamma + n^8)$ asymptotic running time, using the Fleischer-Iwata push-relabel algorithm for strong map sequences of submodular functions (Section 3). A more naive approach would be applying $\mathcal{O}(n)$ times the fastest-available SFM optimizer as a black-box tool. However, by getting more combinatorial insight into push-relabel models (*e.g.*, see Section 3.2.1), the proposed approach provides other advantages. For instance, it reveals two faster sensitivity analysis techniques: (i) exploit the combinatorial structure of the optimal primal-dual solutions in submodular polytopes (Section 3.3.1) or (ii) formulate input parameter changes as strong map sequences of functions (Section 3.3.2). Furthermore, we showed that our solution method can actually determine the optimal solution for any of the $\mathcal{O}(n^5)$ possible pricing scenarios discussed in Section 3.3.2.2 using $\mathcal{O}(n^8\gamma + n^9)$ time, (significantly) more rapidly than naively calling $\mathcal{O}(n^6)$ times a faster SFM black-box optimizer.

References

- [1] Egon Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989.
- [2] Cynthia Barnhart, Ellis Johnson, George Nemhauser, Martin Savelsbergh, and Pamela Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.

- [3] O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical Programming*, 113(2):299–344, 2008.
- [4] Sven de Vries and Rakesh V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309, 2003.
- [5] Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In Richard Guy, Haim Hanani, Norbert Sauer, and Johanan Schonheim, editors, *Combinatorial Structures and their Applications*, pages 69–87. Gordon and Breach, 1970.
- [6] Lisa Fleischer and Satoru Iwata. A push-relabel framework for submodular function minimization and applications to parametric optimization. *Discrete Applied Mathematics*, 131(2):311 – 322, 2003.
- [7] Frederick Hillier and Gerald Lieberman. *Introductions to Operations Research (7th edition)*. McGraw-Hill, 2001.
- [8] Satoru Iwata. A faster scaling algorithm for minimizing submodular functions. *SIAM Journal of Computing*, 32(4):833–840, 2003.
- [9] Satoru Iwata and Kiyohito Nagano. Submodular function minimization under covering constraints. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 671–680. IEEE Computer Society, Atlanta, Georgia, USA, 2009.
- [10] Jakob Krarup and Peter Mark Pruzan. The simple plant location problem: survey and synthesis. *European Journal of Operational Research*, 12(1):36–81, 1983.
- [11] László Lovász. Submodular functions and convexity. In Achim Bachem, Bernhard Korte, and Martin Grtschel, editors, *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.
- [12] S. Thomas McCormick. Submodular function minimization. In G.L. Nemhauser K. Aardal and R. Weismantel, editors, *Discrete Optimization*, volume 12 of *Handbooks in Operations Research and Management Science*, pages 321 – 391. Elsevier, 2005.
- [13] Kiyohito Nagano. On convex minimization over base polytopes. In Matteo Fischetti and DavidP. Williamson, editors, *Integer Programming and Combinatorial Optimization*, volume 4513 of *Lecture Notes in Computer Science*, pages 252–266. Springer Berlin Heidelberg, 2007.
- [14] James B Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.
- [15] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.
- [16] Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume B. Springer Science & Business Media, 2002.
- [17] Yogeshwer Sharma, Chaitanya Swamy, and David Williamson. Approximation algorithms for prize collecting forest problems with submodular penalty functions. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1275–1284. SIAM, 2007.
- [18] Zuo-Jun Max Shen, Collette R. Coullard, and Mark S. Daskin. A joint location - inventory model. *Transportation Science*, 37(1):40–55, 2003.
- [19] Daniel Vainsencher, Ofer Dekel, and Shie Mannor. Bundle selling by online estimation of valuation functions. In Lise Getoor and Tobias Scheffer, editors, *Proceedings the 28th International Conference on Machine Learning (USA)*, pages 1137–1144. Omnipress, 2011.