Noname manuscript No.
(will be inserted by the editor)

Maximizing submodular utility functions combined with a set-union operator over a discrete set

Stefano Coniglio · Fabio Furini · Ivana Ljubić

Received: date / Accepted: date

Abstract We study a discrete optimization problem calling for the maximization of the expected value of a submodular, concave, and differentiable function f combined with a set-union operator. The former models the decision maker's utility function, while the latter models a covering relationship between two ground sets, a set of items N and a set of metaitems $\hat{N}$. The goal of the problem is to find an optimal subset of metaitems $\hat{S} \subseteq \hat{N}$ such that the total utility of the items they cover, as determined by f, is maximized. This problem, which is a generalization of the one introduced by Ahmed and Atamtürk [2011], can be modeled as a mixed integer nonlinear program involving binary decision variables associated with the items and metaitems in N and $\hat{N}$.

In the paper, we propose a double-hypograph decomposition method which allows for projecting out the variables associated with the items in N by separately exploiting the structural properties of the function f and of the set-union operator. With this technique, the function f is linearized using an outer-approximation technique, whereas the set-union operator is linearized in two ways: (i) via a reformulation based on submodular cuts, and (ii) via a Benders decomposition. We compare the strength of the resulting inequalities from a theoretical perspective, and embed them into two corresponding branch-and-cut algorithms. We compare them experimentally to a standard reformulation based on submodular cuts as well as to a state-of-the-art global-optimization solver. The results reveal that, on our testbed, the method based on combining an outer approximation with Benders cuts significantly outperforms the other ones.

Keywords Submodular maximization · Branch-and-Cut · Benders decomposition

1. Introduction

The problem of maximizing the expected value of a concave, strictly increasing, and differentiable utility function $f : \mathbb{R} \to \mathbb{R}$ over a discrete set plays an important role in many applications. Examples can be found in, among others, investment problems with discrete choices such as infrastructure projects, venture capital, and private equity deals (Ahmed

Stefano Coniglio
Mathematical Sciences, University of Southampton,
E-mail: s.coniglio@soton.ac.uk

Fabio Furini
Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti"
Consiglio Nazionale delle Ricerche (IASI-CNR), Roma, Italy,
E-mail: fabio.furini@iasi.cnr.it

Ivana Ljubić
ESSEC Business School of Paris,
E-mail: ivana.ljubic@essec.edu

and Atamtürk [2011]), project selection (Klastorin [1990], Mehrez and Sinuany-Stern [1983], Weingartner [1966]), competitive facility location (Berman and Krass [1998], Aboolian et al. [2007], Ljubić and Moreno [2018]), and combinatorial auctions (Feige et al. [2011], Lehmann et al. [2006]). In such utility-maximization problems, the function f typically models the decision maker's risk-averse attitude.

Together with his coauthors, Shabbir Ahmed made substantial contributions in advancing the state-of-the-art of mathematical programming methods for solving this challenging set of problems (Ahmed and Atamtürk [2011], Yu and Ahmed [2016, 2017]). In this article, we study combinatorial-optimization problems in which the expected value of the function f is maximized over a discrete set when combined with a set union operator, thus generalizing the problem addressed in the latter three articles.

1.1 The problem of Ahmed and Atamtürk [2011]

Let N be a ground set of n items and M a finite set of m scenarios, each occurring with probability π_i, for $i \in M$, and let $a_{ij} \in \mathbb{Q}_+$, $i \in M, j \in N$, and $d_i \in \mathbb{Q}_+$, $i \in M$, be nonnegative numbers. Letting the variable $x_j \in \{0,1\}^n$ be equal to 1 if and only if item $j \in N$ is chosen, the problem tackled in Ahmed and Atamtürk [2011] can be cast as the following Mixed Integer Nonlinear Program (MINLP):

$$\max_{x \in \{0,1\}^n} \left\{ \sum_{i \in M} \pi_i \, f\Big(\sum_{j \in N} a_{ij} \, x_j + d_i \Big) : x \in X \right\}, \tag{1}$$

where X is a polyhedron encapsulating the constraints of the problem and $X \cap \{0,1\}^n$ is the discrete set of feasible solutions. From a stochastic-optimization perspective, Problem (1) calls for maximizing

$$\mathbb{E}_\xi \left[f\Big(\sum_{j \in N} a_j(\xi) \, x_j + d(\xi) \Big) \right]$$

subject to $x \in X$, where ξ is a random variable and the discrete realizations of $a_j(\xi)$, $j \in N$, and $d(\xi)$ are, respectively, $\{a_{1j}, \dots, a_{mj}\}$ and $\{d_1, \dots, d_m\}$.

The objective function of Problem (1) can be equivalently interpreted as a linear combination of m set functions $h_i : 2^N \to \mathbb{R}^+$ with weights π_i, for $i \in M$. For each $i \in M$ and $S \subseteq N$, h_i is defined as the composition of f to an affine set function $q_i : 2^N \to \mathbb{R}$ which maps subsets of N into the reals. Formally:

$$h_i(S) := f\Big(\underbrace{\sum_{j \in S} a_{ij} + d_i}_{q_i(S)} \Big), \quad i \in M, S \subseteq N.$$

Assuming, as done in [Ahmed and Atamtürk, 2011, Yu and Ahmed, 2017], that f be *strictly* increasing, it is not difficult to see that the function h_i is submodular for every $i \in M$, cf. [Ahmed and Atamtürk, 2011, Yu and Ahmed, 2017], and so is the whole objective function of the problem. Thanks to this, Problem (1) can be reformulated as a Mixed Integer Linear Program (MILP) by exploiting a reformulation, originally due to Nemhauser et al. [1978], which features an exponential number of linear inequalities, one per subset S of the ground set N. Ahmed and Atamtürk [2011] showed how to tighten such inequalities by a sequence-independent sequential lifting procedure. In a follow-up article, Yu and Ahmed [2017] further improved the strength of such inequalities for the case where the set of discrete choices X is subject to a single knapsack constraint. Further results on the nature of the exact lifting function adopted in Ahmed and Atamtürk [2011] are reported in Shi et al. [2020].

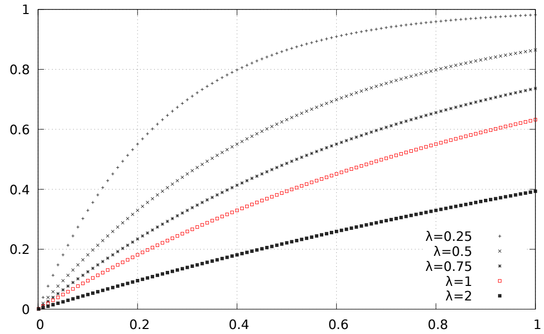

Fig. 1: The utility function $f(z) = 1 - e^{-\frac{z}{\lambda}}$.

The function f adopted in the computation tests of [Ahmed and Atamtürk, 2011] and [Yu and Ahmed, 2017] is the negative exponential function

$$f(z) = 1 - \exp\left(-\frac{z}{\lambda}\right), \tag{f}$$

which is frequently used to model the behavior of risk-averse decision makers. Here, $\lambda > 0$ represents the risk-tolerance parameter and, the larger the value of λ (and, thus, the more f is closer to being linear), the less risk-averse the decision maker is. An illustration of function (f) for different values of λ is provided in Figure 1.

1.2 The generalized problem

In this work, we study a generalization of Problem (1) in which, besides the set of items N, we are given an additional ground set $\hat{N}$ of $\hat{n}$ *metaitems*. We assume that the two ground sets N and $\hat{N}$ are linked by a *covering relationship* modeled by a bipartite graph $G = (\hat{N} \cup N, E)$ where, for each $j \in N$ and $\ell \in \hat{N}$, $\{j, \ell\} \in E$ if and only if item j is covered by metaitem ℓ. We observe that $|E|$, i.e., the size of the edge set of G, corresponds to the number of covering relationships between items and metaitems. For each item $j \in N$, we define by $\hat{N}(j) \subseteq \hat{N}$ the set of metaitems of $\hat{N}$ by which j is covered. Similarly, we define by $N(\ell) \subseteq N$ the set of items that are covered by the metaitem ℓ, for each $\ell \in \hat{N}$. An illustration of an instance of the problem is reported in Figure 2.

The key feature of this generalized problem is that, in order to take an item j of N and benefit from its profit, the decision maker has to select at least one among the metaitems $\ell \in \hat{N}$ which cover it. A list of applications of such a problem is reported at the end of the section.

Upon introducing a binary variable $y_\ell \in \{0, 1\}$ for each $\ell \in \hat{N}$, equal to 1 if and only if metaitem ℓ is chosen, the problem we study can be cast as the following MINLP:

$$\max_{\substack{x \in \{0,1\}^n \\ y \in Y \cap \{0,1\}^{\hat{n}}}} \quad \sum_{i \in M} \pi_i \, f\Big(\sum_{j \in N} a_{ij} \, x_j + d_i\Big) \tag{2a}$$

$$\text{s.t.} \quad x_j \leq \sum_{\ell \in \hat{N}(j)} y_\ell \qquad j \in N \tag{2b}$$

$$y_\ell \leq x_j \qquad \ell \in \hat{N}, j \in N(\ell), \tag{2c}$$

where Y is a polyhedron encapsulating the constraints of the problem and $Y \cap \{0, 1\}^{\hat{n}}$ is the discrete set of feasible choices of metaitems. The parameters π, a, and d have the same meaning as in Problem (1). For each $j \in N$, the constraints linking the x and y

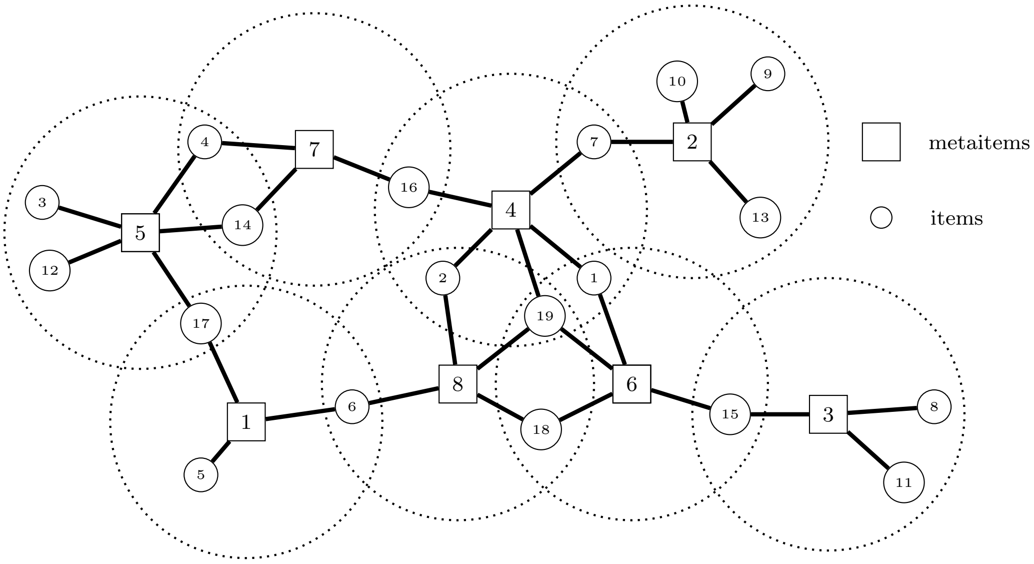

Fig. 2: An instance with $n = 19$ items (the members of N are denoted by circles) and $\hat{n} = 8$ metaitems (the members of $\hat{N}$ are denoted by squares). The relationship of coverage between a metaitem $\ell \in \hat{N}$ and an item $j \in N$ is depicted by the edge $\{\ell, j\}$. In this instance, the covering relationship between items and metaitems is induced by a "covering radius" depicted in the figure by the dotted circumferences. In other words, the sets $N(\ell)$ are determined by the discs centered on the metaitems. E.g., for $\ell = 1$, we have $N(1) = \{5, 6, 17\}$.

variables imply that an item j belonging to the ground set N is taken if and only if at least a metaitem ℓ in $\hat{N}$ has been chosen from the set of metaitems $\hat{N}(j) \subseteq \hat{N}$ that cover it. Note that, as the objective function of the problem is nondecreasing in x, the constraints $y_\ell \leq x_j, \ell \in \hat{N}, j \in N(\ell)$, can be dropped due to being automatically satisfied in any optimal solution. Besides being binary, we do not assume further constraints on x.

Problem (2) is a proper generalization of Problem (1) in which the constraints $x \in X$ imposed on the items in Problem (1) are now imposed on the metaitems ($y \in Y$). Indeed, Problem (1) is obtained by setting $\hat{N} := N$ and defining the edge set E as a matching, i.e., by introducing an edge for each pair of item $j \in N$ and metaitem $\ell \in \hat{N}$ having the same index.

Let

$$N(\hat{S}) := \bigcup_{\ell \in \hat{S}} N(\ell), \qquad \hat{S} \subseteq \hat{N} \tag{3}$$

be the *set-union operator* expressing the covering relationship between $\hat{N}$ and N. Thanks to this operator, we can, similarly to Problem (1), interpret the objective function of Problem (2) as a linear combination of m set functions $\hat{h}_i : 2^{\hat{N}} \to \mathbb{R}^+$ with weights π, $i \in M$, each defined as the composition of f with an affine function $\hat{q}_i : 2^{\hat{N}} \to \mathbb{R}$:

$$\hat{h}_i(\hat{S}) := f\Big(\underbrace{\sum_{j \in N(\hat{S})} a_{ij} + d_i}_{\hat{q}_i(\hat{S})}\Big).$$

Differently from the function q_i occurring in Problem (1), which maps the chosen set of items S into its value, $\hat{q}_i$ in Problem (2) maps the choice of metaitems $\hat{S}$ into the value of the items that $\hat{S}$ covers. Indeed, by relying on the set-union operator, $\hat{q}_i$ can be equivalently defined as:

$$\hat{q}_i(\hat{S}) := \sum_{j \in N(\hat{S})} a_{ij} + d_i \qquad i \in M, \hat{S} \subseteq \hat{N}. \tag{4}$$

The function $\hat{q}_i$ occurs frequently in combinatorial optimization problems which feature a *set covering* component, and it has been heavily studied in the combinatorial optimization literature—see, e.g., Schrijver [2003], pag. 768.

Crucially, due to d_i and a_{ij} being nonnegative for all $i \in M, j \in N$, $\hat{q}_i$ is submodular in its input $\hat{S} \subseteq \hat{N}$ (differently from q_i, which is just affine in its input $S \subseteq N$). As f is concave and increasing and $\hat{q}_i$ is submodular, it is not difficult to see that their composition $\hat{h}_i$ is submodular as well.

Problem (2) is $\mathcal{NP}$-hard even in the, arguably, simplest setting where Y only contains a single cardinality constraint, and even if one of the two key aspects of the problem is dropped by trivializing f and defining it as the identity function $f(z) := z$, as in such a case, the decision version of the set covering problem can be reduced to Problem (2). This is because, for $m = 1$, $a_{1j} = 1$ for all $j \in N$, $d_1 = 0$, and $Y = \{\hat{S} \subseteq \hat{N} : |\hat{S}| \leq k\}$, Problem (2) admits a solution of value n if and only if there is a feasible solution to the problem of covering N with at most k subsets from $\{N(\ell)\}_{\ell \in \hat{N}}$.

1.3 Applications

Problem (2) arises in different applications. An important one is found in the analysis of social networks, when identifying "key players" or "influencers" who can help to quickly distribute a piece of information through the network (Güney et al. [2021], Wu and Küçükyavuz [2018], Kempe et al. [2003], Karimi et al. [2017]). The influence maximization problem assumes a propagation mechanism based on the independent cascade model or the linear threshold model, as proposed in Kempe et al. [2003]. In the former model, the influence propagates probabilistically on the graph according to a specified edge-propagation probability over discrete time steps. In the latter, each node of the graph is influenced only if the (weighed) fraction of the node's neighbors that become active exceeds a node-specific stochastic threshold. Given a graph $\tilde{G} = (\tilde{V}, \tilde{E})$, m so-called *live-edge graphs*, say $\tilde{G}_i = (\tilde{V}, \tilde{E}_i)$ for $i \in M$, are obtained by a sampling process, with π_i being the probability of scenario i. A bipartite graph $G_i = (N \cup \hat{N}, E_i)$ is then constructed for each i. In it, the set of items N is a copy of $\tilde{V}$ (which corresponds to all the users in the social network) and the set of metaitems $\hat{N}$ is a copy of a subset of vertices from $\tilde{V}$ that can act as potential influencers. For each $i \in M$ and each $\ell \in \hat{N}$, $N_i(\ell) \subseteq N$ is the subset of users that can be reached by influencer ℓ in the live-edge graph $\tilde{G}_i$ due to belonging to the same connected component as ℓ (see also Wu and Küçükyavuz [2018] for further details). The goal is then to choose a subset of influencers so as to maximize the expected number of users that can be reached. This requires the following slight modification of our set-union operator and of function $\hat{q}_i$, defined in (3)–(4), thanks to which the former becomes scenario-dependent:

$$N_i(\hat{S}) := \bigcup_{\ell \in \hat{S}} N_i(\ell), \quad i \in M, \hat{S} \subseteq \hat{N}, \qquad \hat{q}_i(\hat{S}) := \sum_{j \in N_i(\hat{S})} a_{ij} + d_i \quad i \in M, \hat{S} \subseteq \hat{N},$$

and a_{ij} represents the value (typically equal to 1) of reaching user $j \in N$ in scenario $i \in M$. Without loss of generality, all the results presented in this paper can be applied to such a scenario-dependent set-union operator as well. However, to simplify the presentation, we will restrict ourselves to the original definition given in (3)–(4). In the standard applications of influence maximization referenced above, the function f is the identity function. Instead, by allowing f to be concave, with method we develop in this paper one can take the risk-tolerances of the decision maker into account when measuring the expected influence in the network.

A second application arises in marketing problems, where the set M models various products that a marketer is interested in advertising to a set N of potential customers, and $\hat{N}$ is a set of marketing campaigns, where each campaign $\ell \in \hat{N}$ allows for reaching a subset $N(\ell)$ of customers. In such a case, the value of a_{ij}, for each $i \in M, j \in N$, corresponds to the demands of product i that is due to customer j. Letting, for each $i \in M$, π_i be a weight measuring the relevance of a product i, the utility function f is used to express the decreasing marginal utility of serving an additional unit of customer demand.

A third application arises in a class of stochastic competitive facility location problems with uncertain customer demands. In the deterministic competitive facility location problem (see, e.g., [Ljubić and Moreno, 2018, Berman and Krass, 1998, Aboolian et al., 2007] and further references therein), a subset of facilities has to be open subject to a budget constraint so that the captured customer demand is maximized (each customer can also choose to be served by the competitor instead). The set $\hat{N}$ corresponds to a set of potential facility locations, whereas the customers are represented by the set N. Using the multinomial logit model, for example, the market share function satisfies the properties of our function f. In the stochastic setting, the expected market share is then calculated over the set M of possible scenario realizations (each with probability π_i, $i \in M$), and a_{ij} corresponds to the demand of customer $j \in N$ under scenario $i \in M$.

1.4 Contribution and outline of the paper

Throughout the paper, we focus on the case with $\hat{n} \ll n$, where n is a few orders of magnitude larger than $\hat{n}$. This is the case of many applications, including those we introduced before. The main contribution of the paper is an exact method for solving Problem (2) based on a double-hypograph decomposition which allows for projecting out the variables associated with the items in N. The method exploits the structural properties of the function f and of the set-union operator. In it, the function f is linearized using an outer-approximation method, whereas the set-union operator is linearized in two ways: (i) via a reformulation based on submodular cuts, and (ii) via a Benders decomposition. In particular, we show that, assuming f can be computed in constant time, the inequalities arising from the outer approximation combined with Benders decomposition can be separated in linear time even for fractional points. After comparing the strength of the two resulting mixed integer linear programming reformulations from a theoretical perspective, we embed them in two branch-and-cut algorithms. According to our computational experiments, the most efficient of them allows for solving to optimality instances of the problem with up to $n = 20,000$, $\hat{n} = 60$, and $m = 100$ in a short amount of computing time.

The paper is organized as follows. After recalling some background notions on submodularity in Section 2, we introduce our decomposition approach and our two problem reformulations in Section 3. The strength of the inequalities introduced in these two reformulations is compared in Section 4. Section 5 presents an extension of our method allowing for a concave and differentiable function f which is not necessarily increasing. In Section 6, we investigate the construction of worst-case instances for the classical greedy algorithm proposed in Nemhauser et al. [1978] for maximizing a nondecreasing submodular function subject to a k-cardinality constraint when applied to Problem (2). In Section 7, we demonstrate via computational experiments the advantages offered by our method based on outer approximation, variable projection, and submodular or Benders cuts. Concluding remarks are reported in Section 8.

2. Preliminaries

Before introducing the decomposition approach that we propose for solving Problem (2), we recall, in this section, the basic terminology and notions used in optimization problems involving submodular functions. We also summarize a key result due to Nemhauser et al. [1978], which leads to a direct MINLP reformulation of Problem (2).

Let $h : 2^N \to \mathbb{R}$ be a generic set function defined for some ground set N. The associated set function $\varrho_j^h(S) := h(S \cup \{j\}) - h(S)$, $j \in N, S \subseteq N$, is called *marginal contribution* of j with respect to S. The function h is said *nondecreasing* if $\varrho_j^h(S) \geq 0$ for all $j \in N, S \subseteq N$, and *submodular* if $\varrho_j^h(S) \geq \varrho_j^h(T)$ for all $S, T \subseteq N$ with $S \subseteq T$ and $j \in N$. Throughout the paper, to indicate that a marginal contribution is referred to a specific function, we report the name of such a function as a superscript, as done in ϱ_j^h.

Every submodular set function h enjoys the following properties:

Proposition 1 (Nemhauser et al. [1978]) *If h is submodular, then:*

$$h(T) \leq h(S) - \sum_{j \in S \setminus T} \varrho_j^h(N \setminus \{j\}) + \sum_{j \in T \setminus S} \varrho_j^h(S), \qquad \text{for } S, T \subseteq N$$

$$h(T) \leq h(S) - \sum_{j \in S \setminus T} \varrho_j^h(S \setminus \{j\}) + \sum_{j \in T \setminus S} \varrho_j^h(\emptyset), \qquad \text{for } S, T \subseteq N.$$

By applying Proposition 1 to the function $\hat{h}_i$, $i \in M$, of Problem (2), we straightforwardly obtain the following MILP reformulation, which we refer to as *Reformulation based on Submodular Cuts*, or Ref(SC):

$$\max_{\substack{w \in \mathbb{R}^m \\ y \in Y \cap \{0,1\}^{\hat{n}}}} \quad \sum_{i \in M} \pi_i w_i$$

$$\text{s.t.} \quad w_i \leq \hat{h}_i(\hat{S}) + \sum_{\ell \in \hat{N} \setminus \hat{S}} \varrho_\ell^{\hat{h}_i}(\hat{S}) y_\ell - \sum_{\ell \in \hat{S}} \varrho_\ell^{\hat{h}_i}(\hat{N} \setminus \{\ell\})(1 - y_\ell) \qquad \hat{S} \subseteq \hat{N}, i \in M \qquad (\text{SC1}^{\hat{h}_i})$$

$$w_i \leq \hat{h}_i(\hat{S}) + \sum_{\ell \in \hat{N} \setminus \hat{S}} \varrho_\ell^{\hat{h}_i}(\emptyset) y_\ell - \sum_{\ell \in \hat{S}} \varrho_\ell^{\hat{h}_i}(\hat{S} \setminus \{\ell\})(1 - y_\ell) \qquad \hat{S} \subseteq \hat{N}, i \in M. \qquad (\text{SC2}^{\hat{h}_i})$$

The formulation is clearly correct as, for $\hat{S} = \{\ell \in \hat{N} : y_\ell = 1\}$, Constraints (SC1$^{\hat{h}_i}$)–(SC2$^{\hat{h}_i}$) boil down to $w_i \leq \hat{h}_i(\hat{S})$ for each $i \in M$, thereby guaranteeing $\sum_{i \in M} \pi_i w_i = \sum_{i \in M} \pi_i \hat{h}_i(\hat{S})$, which is equal to the objective function value of Problem (2).

We remark that the reformulation would still be correct if we were to replace Constraints (SC1$^{\hat{h}_i}$) by the following constraints:

$$w_i \leq \hat{h}_i(\hat{S}) + \sum_{\ell \in \hat{N} \setminus \hat{S}} \varrho_\ell^{\hat{h}_i}(\hat{S}) y_\ell \qquad \hat{S} \subseteq \hat{N}, i \in M. \qquad (\text{SC3}^{\hat{h}_i})$$

We observe that such constraints are as strong as (SC1$^{\hat{h}_i}$) unless there is an item $j \in N$ which is covered by a single metaitem $\ell \in \hat{N}$, i.e., such that $|\hat{N}(j)| = 1$ and $a_{ij} > 0$. In such a case, Constraints (SC1$^{\hat{h}_i}$) dominate Constraints (SC3$^{\hat{h}_i}$). This is because, since the function $\hat{h}_i$ is nonnegative and nondecreasing, $\varrho_\ell^{\hat{h}_i}(\hat{N} \setminus \{\ell\})$ is always equal to 0 unless there is an item $j \in N$ such that $\hat{N}(j) = \{\ell\}$ and $a_{ij} > 0$.

While extensively used in many works involving submodular functions, formulations similar to this one are known to be rather weak (Ahmed and Atamtürk [2011], Ljubić and Moreno [2018], Nemhauser et al. [1988]). Computational experiments obtained for it are reported in Section 7.

3. Decomposition approaches and MILP reformulations for Problem (2)

In spite of its weakness, the Ref(SC) works in the y-space only, which is advantageous since, as we mentioned, in this work we focus on cases where $n \gg \hat{n}$. In this section, we propose two techniques which, similarly to the Ref(SC), allow for solving Problem (2) in the y space by projecting out the x variables. As we will show in Section 7 with computational experiments, the reformulations arising from such techniques substantially outperform the Ref(SC) on the testbed we consider.

The techniques we propose are based on a decomposition approach which allows for exploiting the structural properties of the function f as well as those of the set-union operator underlying the function $\hat{q}_i$, $i \in M$. In particular, it allows for combining an outer approximation of the function f—see, e.g., Duran and Grossmann [1986], Fletcher and Leyffer [1994] for general references on outer approximations, with linear approximations introduced for each function $\hat{q}_i$ into a single MILP formulation to be solved by a branch-and-cut method by, at the same time, projecting out all the x variables.

3.1 Double-hypograph decomposition

The key idea of our decomposition approach is reformulating Problem (2) in such a way that the function $\hat{h}_i$, $i \in M$, which, we recall, is defined as $f \circ \hat{q}_i$, is decomposed into its two constituent parts, f and $\hat{q}_i$.

Letting y be the characteristic vector of $\hat{S} \subseteq \hat{N}$, the set function $\hat{q}_i(\hat{S})$, $i \in M$, can be rewritten as the following function $\hat{\mathcal{Q}}_i(y) : \{0,1\}^{\hat{n}} \to \mathbb{Z}_+$:

$$\hat{\mathcal{Q}}_i(y) := \max_{x \in \{0,1\}^n} \left\{ \sum_{j \in N} a_{ij} x_j + d_i : \quad x_j \leq \sum_{\ell \in \hat{N}(j)} y_\ell, \;\; j \in N \right\} \qquad i \in M. \tag{VF}$$

$\hat{\mathcal{Q}}_i(y)$ can be interpreted as the value function of the subproblem of computing $\hat{q}_i(\hat{S})$ for a subset of metaitems $\hat{S} \subseteq \hat{N}$ with incidence vector $y \in Y \cap \{0,1\}^{\hat{n}}$. Such a function is studied by Cordeau et al. [2019] in the context of covering facility location problems. Notice that the problem of computing $\hat{\mathcal{Q}}_i(y)$ can be solved in linear time $O(|E|)$ by letting $x^*_j := 1$ for each $j \in N$ such that $\sum_{\ell \in \hat{N}(j)} y_\ell \geq 1$, and letting $x^*_j := 0$ otherwise. Moreover, as the problem only features upper bounds on the x variables, equal to $\min\{1, \sum_{\ell \in \hat{N}(j)} y_\ell\}$ for each $j \in N$, its Linear Programming (LP) relaxation is integer. This property will be useful later.

After introducing the auxiliary variables η_i, w_i, $i \in M$, we consider the following *double hypograph* reformulation, which employs two hypograph reformulations applied in sequence:

$$\max_{\substack{w,\eta \in \mathbb{R}^m \\ y \in Y \cap \{0,1\}^{\hat{n}}}} \sum_{i \in M} \pi_i w_i$$

$$w_i \leq f(\eta_i) \qquad i \in M \tag{HYPO1}$$

$$\eta_i \leq \hat{\mathcal{Q}}_i(y) \qquad i \in M. \tag{HYPO2}$$

This reformulation is correct as, due to $\pi_i \geq 0$ for all $i \in M$, $w_i = f(\eta_i)$ holds in any optimal solution. In turn, due to f being increasing, this implies $\eta_i = \hat{\mathcal{Q}}_i(y)$ in any optimal solution.

Starting from this decomposition, in the following we propose two MILP reformulations of Problem (2) belonging to the (w, y) space and featuring $O(\hat{n} + m)$ variables. Both reformulations are obtained by projecting out the n decision variables x associated with items $j \in N$, as well as the m auxiliary variables η.

3.2 Projecting out the η variables

The auxiliary variable η_i, $i \in M$, can be projected out as follows using Fourier-Mötzkin elimination:

Proposition 2 *Constraints (HYPO1)–(HYPO2) can be replaced by the following constraints:*

$$w_i \le f(p) - f'(p)p + f'(p)\hat{\mathcal{Q}}_i(y) \qquad i \in M, p \in [0, \sum_{j\in N} a_{ij} + d_i]. \tag{OA}$$

Proof Since f is concave, for each $i \in M$ the set of all pairs $(w_i, \eta_i) \in \mathbb{R}^2$ which satisfy Constraint (HYPO1) forms a convex set. By means of an outer-approximation technique, we can restate Constraints (HYPO1) as:

$$w_i \le (f(p) - f'(p)p) + f'(p)\eta_i \qquad i \in M, p \in [0, \sum_{j\in N} a_{ij} + d_i], \tag{7}$$

where f' is the first derivative of f and $[0, \sum_{j\in N} a_{ij} + d_i]$ is a superset of the set of values that η_i (and, therefore, $\hat{\mathcal{Q}}_i$ and $\hat{q}_i$) can take. As $f' > 0$ due to f being increasing, using Fourier-Mötzkin elimination we can combine Constraints (7) and (HYPO2) to project out the η_i variables, which results in Constraints (OA). ■

In the following two subsections, we discuss two ways of obtaining MILP reformulations of Problem (2) starting from Constraints (OA). The first oneexploits the submodularity of $\hat{q}_i$, $i \in M$. The second onerelies on the integrality property of the LP relaxation of (VF), and it exploits LP duality in a Benders-cuts fashion. The idea is to combine Constraints (OA) with a finite collection of affine functions yielding an over-estimation of $\hat{\mathcal{Q}}_i$.

3.3 Reformulation based on Outer Approximation plus Submodular Cuts: Ref(OA+SC)

As $\hat{q}_i$ is submodular, the following constraints are valid due to Proposition 1:

$$\hat{\mathcal{Q}}_i(y) \le \hat{q}_i(\hat{S}) + \sum_{\ell \in \hat{N}\setminus\hat{S}} \varrho_\ell^{\hat{q}_i}(\hat{S})y_\ell - \sum_{\ell\in\hat{S}} \varrho_\ell^{\hat{q}_i}(\hat{N}\setminus\{\ell\})(1-y_\ell) \qquad \hat{S} \subseteq \hat{N} \tag{SC1$^{\hat{q}_i}$}$$

$$\hat{\mathcal{Q}}_i(y) \le \hat{q}_i(\hat{S}) + \sum_{\ell \in \hat{N}\setminus\hat{S}} \varrho_\ell^{\hat{q}_i}(\emptyset)y_\ell - \sum_{\ell\in\hat{S}} \varrho_\ell^{\hat{q}_i}(\hat{S}\setminus\{\ell\})(1-y_\ell) \qquad \hat{S} \subseteq \hat{N}. \tag{SC2$^{\hat{q}_i}$}$$

Combining these constraints with Constraints (OA) of Proposition 2, we obtain the following constraints for each choice of $\hat{S} \subseteq \hat{N}$, $i \in M$, and $p \in [0, \sum_{j\in N} a_{ij} + d_i]$:

$$w_i \le f(p) - f'(p)p + f'(p)\hat{q}_i(\hat{S}) + \sum_{\ell \in \hat{N}\setminus\hat{S}} f'(p)\varrho_\ell^{\hat{q}_i}(\hat{S})y_\ell - \sum_{\ell\in\hat{S}} f'(p)\varrho_\ell^{\hat{q}_i}(\hat{N}\setminus\{\ell\})(1-y_\ell)$$

$$w_i \le f(p) - f'(p)p + f'(p)\hat{q}_i(\hat{S}) + \sum_{\ell \in \hat{N}\setminus\hat{S}} f'(p)\varrho_\ell^{\hat{q}_i}(\emptyset)y_\ell - \sum_{\ell\in\hat{S}} f'(p)\varrho_\ell^{\hat{q}_i}(\hat{S}\setminus\{\ell\})(1-y_\ell).$$

By restricting ourselves to $p = \hat{q}_i(\hat{S})$, we obtain the following MILP reformulation of Problem (2), which we refer to as *Reformulation based on Outer Approximation plus Submodular Cuts*, or Ref(OA+SC):

$$\max_{\substack{w\in\mathbb{R}^m \\ y\in Y\cap\{0,1\}^{\hat{n}}}} \sum_{i\in M} \pi_i w_i$$

$$\text{s.t.} \quad w_i \le f(\hat{q}_i(\hat{S})) + \sum_{\ell \in \hat{N}\setminus\hat{S}} f'(\hat{q}_i(\hat{S}))\varrho_\ell^{\hat{q}_i}(\hat{S})y_\ell - \sum_{\ell\in\hat{S}} f'(\hat{q}_i(\hat{S}))\varrho_\ell^{\hat{q}_i}(\hat{N}\setminus\{\ell\})(1-y_\ell)$$
$$\hat{S} \subseteq \hat{N}, i \in M \tag{OA+SC1}$$

$$w_i \le f(\hat{q}_i(\hat{S})) + \sum_{\ell \in \hat{N}\setminus\hat{S}} f'(\hat{q}_i(\hat{S}))\varrho_\ell^{\hat{q}_i}(\emptyset)y_\ell - \sum_{\ell\in\hat{S}} f'(\hat{q}_i(\hat{S}))\varrho_\ell^{\hat{q}_i}(\hat{S}\setminus\{\ell\})(1-y_\ell)$$
$$\hat{S} \subseteq \hat{N}, i \in M. \tag{OA+SC2}$$

Proposition 3 *Ref(OA+SC) is valid.*

Proof Let $y' \in Y \cap \{0,1\}^{\hat{n}}$ be a feasible solution to Problem (2). Letting $\hat{S}' := \{\ell \in \hat{N} : y'_j = 1\}$ for each $i \in M$, the pair of Constraints (OA+SC1) and (OA+SC2) corresponding to $\hat{S} = \hat{S}'$ boil down to $w_i \leq f(\hat{q}_i(\hat{S}'))$, thereby guaranteeing $\sum_{i \in M} \pi_i w_i = \sum_{i \in M} \pi_i f(\hat{q}_i(\hat{S}'))$, which coincides with the objective function of Problem (2). ■

When solving the Ref(OA+SC) with a branch-and-cut method, as we will do for the computational results reported in Section 7, to obtain a convergent method it suffices to look for the existence of violated constraints among (OA+SC1) and (OA+SC2) just for integer solutions y^* which arise during the execution of the algorithm. The following result shows that such binary vectors can be separated very efficiently.

Proposition 4 *Assume that the function f and its derivative can be computed in constant time. Given a vector $y^* \in \{0,1\}^{\hat{n}}$, the separation problem calling for a constraint among* (OA+SC1)–(OA+SC2) *that is strictly violated by y^* can be solved in linear time $O(|E|)$.*

Proof As observed in the proof of Proposition 3, Constraints (OA+SC1)–(OA+SC2) impose $w_i \leq f(\hat{q}_i(\hat{S}'))$ for every set $\hat{S}' \subseteq \hat{N}$. Letting $\hat{S}^* := \{\ell \in \hat{N} : y^*_\ell = 1\}$, it follows that, if y^* is infeasible, the two constraints among (OA+SC1)–(OA+SC2) with $\hat{S} = \hat{S}^*$ are violated for some $i \in M$. For any given $\hat{S}^*$, the coefficients of the two summations in the right-hand side of Constraints (OA+SC1)–(OA+SC2) are proportional to the marginal contributions of $\hat{q}_i$, which, due to assuming that f can be computed in constant time, can be computed in linear time $O(|E|)$. In particular, the coefficients $\varrho_\ell^{\hat{q}_i}(\hat{N} \setminus \{\ell\})$, $\varrho_\ell^{\hat{q}_i}(\emptyset)$ do not depend on $\hat{S}^*$ and, hence, they can be precomputed. Computing the multiplicative coefficient $f'(\hat{q}_i(\hat{S}))$ as well as $f(\hat{q}_i(\hat{S}))$ is equivalently easy as, besides simple arithmetic operations, it only requires the evaluation of f and its derivative f' at $\hat{q}_i(\hat{S}^*)$, both of which, by assumption, require linear time. ■

3.4 Reformulation based on Outer Approximation plus Benders Cuts: Ref(OA+BC)

Let us focus on the subproblem underlying the value function $\hat{\mathcal{Q}}_i$, $i \in M$, defined in (VF). Assuming $y \in Y \cap \{0,1\}^{\hat{n}}$, every optimal solution to the LP relaxation of the subproblem of computing $\hat{\mathcal{Q}}_i$ (in which $x \in \{0,1\}^n$ is replaced by $x \in [0,1]^n$) is binary. As the subproblem is bounded, we can consider its LP dual without loss of generality. Let δ and σ be the dual variables associated with, respectively, the constraints linking the x and the y variables and the unit upper-bound constraints on x. Let

$$\mathcal{P}_i := \left\{(\delta, \sigma) \in \mathbb{R}_+^{n+n} : \delta_j + \sigma_j \geq a_{ij}, j \in N\right\}, \quad i \in M,$$

be the polyhedron of the dual feasible solutions (for simplicity, we drop the index i when referring to (δ, σ)). By relying on LP duality, we obtain:

$$\hat{\mathcal{Q}}_i(y) = d_i + \min\left\{\sum_{j \in N} \sigma_j + \sum_{j \in N} \delta_j \left(\sum_{\ell \in \hat{N}(j)} y_\ell\right) : (\delta, \sigma) \in \mathcal{P}_i\right\}. \tag{9}$$

Letting $\mathcal{P}_i^e$, $i \in M$, denote the set of extreme points of $\mathcal{P}_i$, we derive the following *Benders cuts*:

$$\hat{\mathcal{Q}}_i(y) \leq d_i + \sum_{j \in N} \tilde{\sigma}_j + \sum_{j \in N} \tilde{\delta}_j \left(\sum_{\ell \in \hat{N}(j)} y_\ell\right), \quad i \in M, (\tilde{\delta}, \tilde{\sigma}) \in \mathcal{P}_i^e. \tag{BC}$$

Thanks to Proposition 2, we obtain the following alternative MILP reformulation of Problem (2), which we refer to as *Reformulation based on Outer Approximation plus Benders Cuts*, or Ref(OA+BC):

$$\max_{\substack{w \in \mathbb{R}^m \\ y \in Y \cap \{0,1\}^{\hat{n}}}} \sum_{i \in M} \pi_i w_i$$

$$\text{s.t.} \quad w_i \le f(p) - f'(p)p + f'(p)d_i + \sum_{j \in N} f'(p)\tilde{\sigma}_j + \sum_{j \in N} f'(p)\tilde{\delta}_j \Bigg(\sum_{\ell \in \hat{N}(j)} y_\ell \Bigg),$$

$$i \in M, (\tilde{\delta}, \tilde{\sigma}) \in \mathcal{P}_i^e, p \in [0, \sum_{j \in N} a_{ij} + d_i]. \tag{OA+BC}$$

For a given (not necessarily integer) $y^* \in [0,1]^{\hat{n}}$, optimal solutions to (9), which correspond to the *Benders subproblem* in the Benders decomposition literature, can be computed in closed form according to the following lemma:

Lemma 1 *A pair* $(\tilde{\delta}, \tilde{\sigma}) \in \mathbb{R}^{n \times n}$ *is an optimal solution to* (9) *if and only if it satisfies the following:*

$$\begin{array}{lll} \tilde{\delta}_j = a_{ij} & \tilde{\sigma}_j = 0 & \text{for } j \in N : \sum_{\ell \in \hat{N}(j)} y_\ell < 1 \\ \tilde{\delta}_j = 0 & \tilde{\sigma}_j = a_{ij} & \text{for } j \in N : \sum_{\ell \in \hat{N}(j)} y_\ell > 1 \\ \tilde{\delta}_j = \gamma_j a_{ij} & \tilde{\sigma}_j = (1-\gamma_j) a_{ij} \text{ for some } \gamma_j \in [0,1] & \text{for } j \in N : \sum_{\ell \in \hat{N}(j)} y_\ell = 1. \end{array}$$

Proof Problem (9) decomposes into n subproblems, one per item $j \in N$. Each subproblem asks for a value for δ_j and σ_j—we refer to it as $(\tilde{\delta}, \tilde{\sigma})$—satisfying $\delta_j + \sigma_j \ge a_{ij}$ at minimum cost $(\sum_{\ell \in \hat{N}(j)} y_\ell)\delta_j + \sigma_j$. If $\sum_{\ell \in \hat{N}(j)} y_\ell < 1$, the coefficient of δ_j is smaller than the coefficient of σ_j and, thus, we have $\tilde{\delta}_j = a_{ij}$ and $\tilde{\sigma}_j = 0$. If $\sum_{\ell \in \hat{N}(j)} y_\ell > 1$, the coefficient of δ_j is strictly larger than the coefficient of σ_j and, thus, we have $\tilde{\delta}_j = 0$ and $\tilde{\sigma}_j = a_{ij}$. If $\sum_{\ell \in \hat{N}(j)} y_\ell = 1$, the coefficients of δ_j and σ_j are identical and $\tilde{\delta}_j = \gamma_j a_{ij}, \tilde{\sigma}_j = (1-\gamma_j)a_{ij}$ is optimal for every $\gamma_j \in [0,1]$. ■

We remark that, in the lemma, the latter case corresponds to the case where the Benders subproblem is (dual) degenerate, thus admitting multiple optimal solutions. Thanks to this, Lemma 1 generalizes the results given in Propositions 1, 2, and 4 in Cordeau et al. [2019]. We note that, since any dual pair $(\tilde{\delta}, \tilde{\sigma})$ with $\gamma_j \in (0,1)$ for some $j \in N$ can be obtained as a convex combination of two other dual pairs $(\tilde{\delta}^1, \tilde{\sigma}^1)$, $(\tilde{\delta}^2, \tilde{\sigma}^2)$ (one corresponding to $\gamma_j = 1$ and the other one corresponding to $\gamma_j = 0$), the Benders cut corresponding to $(\tilde{\delta}, \tilde{\sigma})$ is implied by the Benders cuts derived from $(\tilde{\delta}^1, \tilde{\sigma}^1)$ and $(\tilde{\delta}^2, \tilde{\sigma}^2)$. Thus, without loss of generality, in Lemma 1 we can restrict ourselves to $\gamma_j \in \{0,1\}$, $j \in N$.

Thanks to the lemma, the following holds:

Proposition 5 *For a given* $w_i^* \in \mathbb{R}^+$ *and a (not necessarily integer)* $y^* \in Y \cap [0,1]^{\hat{n}}$*, the separation problem for Constraints* (OA+BC) *can be solved in linear time* $O(|E|)$ *for each* $i \in M$*.*

Proof Constraint (OA+BC) is violated by w_i^* and $y^* \in Y$ if and only if $w_i^* > f(\hat{\mathcal{Q}}_i(y^*))$, which implies that w_i^* is strictly larger than the outer approximation of f at $p^* := \hat{\mathcal{Q}}_i(y^*)$. Thanks to Lemma 1, the value of $\hat{\mathcal{Q}}_i(y^*)$, as well as that of the dual multipliers (δ, σ), can be computed in linear time $O(|E|)$. ■

The main advantage of the Benders-based reformulation Ref(OA+BC) over the submodular-cuts-based reformulation Ref(OA+SC) is that, as we show in the following section, Constraints (OA+BC) subsume Constraints (OA+SC1)–(OA+SC2). Moreover, assuming that f

and f' can be computed in constant time, the separation problem for Constraints (OA+BC) can be solved in linear time even when y^* is fractional, a property which is not enjoyed by Constraints (OA+SC1)–(OA+SC2). From a branch-and-cut perspective, this allows for the efficient separation of Constraints (OA+BC) at each node of the branch-and-bound tree after solving each LP relaxation, thus allowing for producing tighter bounds throughout the execution of the algorithm, including the root node.

4. On the strength of the new inequalities

In this section, we provide a theoretical comparison of the relative strength of the inequalities introduced in this paper featured in the three reformulations Ref(SC), Ref(OA+SC), and Ref(OA+BC).

We introduce the set $N_1 := \{j \in N : |\hat{N}(j)| = 1\}$, corresponding to the set of items in N which can be covered by a single metaitem $\ell \in \hat{N}$, and we define $\ell(j)$ as the metaitem in $\hat{N}$ covering it. Similarly, for every subset of metaitems $\hat{S} \subseteq \hat{N}$, we let $N_1(\hat{S}) := \{j \in N : |\hat{S} \cap \hat{N}(j)| = 1\}$ be the set of items in N which can be covered by a single metaitem $\ell \in \hat{S}$.

4.1 Comparison between the inequalities in Ref(SC) and Ref(OA+SC)

Let us compare Constraints (SC1$^{\hat{h}_i}$) and Constraints (OA+SC1). The following holds:

Lemma 2 *For each $i \in M$, $\hat{S} \subseteq \hat{N}$, and $\ell \in \hat{N} \setminus \hat{S}$, the coefficient of y_ℓ in Constraint* (SC1$^{\hat{h}_i}$) *is at least as tight as the coefficient of y_ℓ in Constraint* (OA+SC1). *If $\hat{q}_i(\hat{S} \cup \{\ell\}) = \hat{q}_i(\hat{S})$, both coefficients are equal to zero.*

Proof As f is concave and differentiable and $\hat{q}_i$ is nondecreasing, we have:

$$\varrho_\ell^{\hat{h}_i}(\hat{S}) = f(\hat{q}_i(\hat{S} \cup \{\ell\})) - f(\hat{q}_i(\hat{S})) \leq f'(\hat{q}_i(\hat{S})) \left(\hat{q}_i(\hat{S} \cup \{\ell\}) - \hat{q}_i(\hat{S})\right) = f'(\hat{q}_i(\hat{S}))\, \varrho_\ell^{\hat{q}_i}(\hat{S}),$$

thus proving that the coefficient of y_ℓ in (SC1$^{\hat{h}_i}$) is at least as tight as the one in (OA+SC1). If $\hat{q}_i(\hat{S} \cup \{\ell\}) = \hat{q}_i(\hat{S})$, then $\varrho_\ell^{\hat{q}_i}(\hat{S}) = \varrho_\ell^{\hat{h}_i}(\hat{S}) = 0$. ■

Lemma 3 *For each $i \in M$, $\hat{S} \subseteq \hat{N}$, and $\ell \in \hat{S}$ with $\hat{q}_i(\hat{N} \setminus \{\ell\}) > \hat{q}_i(\hat{S})$, the coefficient of $(1-y_\ell)$ in Constraint* (OA+SC1) *is tighter than the coefficient of $(1-y_\ell)$ in Constraint* (SC1$^{\hat{h}_i}$).

Proof As f is concave and differentiable and $\hat{q}_i$ is nondecreasing, we have:

$$\begin{aligned}\varrho_\ell^{\hat{h}_i}(\hat{N} \setminus \{\ell\}) =& f(\hat{q}_i(\hat{N})) - f(\hat{q}_i(\hat{N} \setminus \{\ell\})) \leq f'(\hat{q}_i(\hat{N} \setminus \{\ell\})) \left(\hat{q}_i(\hat{N}) - \hat{q}_i(\hat{N} \setminus \{\ell\})\right) = \\ =& f'(\hat{q}_i(\hat{N} \setminus \{\ell\}))\varrho_\ell^{\hat{q}_i}(\hat{N} \setminus \{\ell\}).\end{aligned}$$

As f is increasing, for all $\ell \in \hat{S} \subseteq \hat{N}$ satisfying $\hat{q}_i(\hat{N} \setminus \{\ell\}) > \hat{q}_i(\hat{S})$, $f'(\hat{q}_i(\hat{N} \setminus \{\ell\})) < f'(\hat{q}_i(\hat{S}))$ holds, implying $\varrho_\ell^{\hat{h}_i}(\hat{N} \setminus \{\ell\}) < f'(\hat{q}_i(\hat{S}))\varrho_\ell^{\hat{q}_i}(\hat{N} \setminus \{\ell\})$. ■

The following proposition provides a sufficient condition under which there is no mutual domination between Constraints (SC1$^{\hat{h}_i}$) and (OA+SC1). It also gives a sufficient condition under which Constraints (SC1$^{\hat{h}_i}$) dominate Constraints (OA+SC1).

Proposition 6 *For each $i \in M$, $\hat{S} \subseteq \hat{N}$, we have:*

1. *If $\hat{q}_i(\hat{N} \setminus \{\ell\}) > \hat{q}_i(\hat{S})$ for some $\ell \in \hat{S}$, Constraints* (SC1$^{\hat{h}_i}$) *and* (OA+SC1) *do not dominate each other.*
2. *If $\hat{q}_i(\hat{S}) = \hat{q}_i(\hat{N})$ then the Constraint* (SC1$^{\hat{h}_i}$) *dominates the Constraint* (OA+SC1).

Proof First, observe that $\hat{h}_i(\hat{S}) = f(\hat{q}_i(\hat{S}))$, so the constant terms of both inequalities are identical.

1. Assume that $\hat{q}_i(\hat{N} \setminus \{\ell\}) > \hat{q}_i(\hat{S})$ for some $\ell \in \hat{S}$. Then, the result follows directly from Lemma 2 and Lemma 3.
2. Assume now that $\hat{q}_i(\hat{S}) = \hat{q}_i(\hat{N})$. This condition is equivalent to $\hat{q}_i(\hat{S}) = \hat{q}_i(\hat{S} \cup \{\ell\})$, for all $\ell \in \hat{N} \setminus \hat{S}$. Hence, by Lemma 2, the coefficients next to y_ℓ, $\ell \in \hat{N} \setminus \hat{S}$, are equal to zero in both constraints. Moreover, as f is concave, increasing, and differentiable and $\hat{q}_i$ is nondecreasing, we have:

$$\begin{aligned}\varrho_\ell^{\hat{h}_i}(\hat{N} \setminus \{\ell\}) =& f(\hat{q}_i(\hat{N})) - f(\hat{q}_i(\hat{N} \setminus \{\ell\})) \geq f'(\hat{q}_i(\hat{N})) \left(\hat{q}_i(\hat{N}) - \hat{q}_i(\hat{N} \setminus \{\ell\})\right) = \\ =& f'(\hat{q}_i(\hat{S}))\varrho_\ell^{\hat{q}_i}(\hat{N} \setminus \{\ell\}).\end{aligned}$$

   and thus the Constraint (SC1$^{\hat{h}_i}$) dominates the Constraint (OA+SC1).

We note that it is more likely that (OA+SC1) and (SC1$^{\hat{h}_i}$) do not to dominate each other when the coverage $N(\hat{S})$ of $\hat{S}$ is small as, in such a case, the condition $\exists \ell \in \hat{S} : \hat{q}_i(\hat{N} \setminus \{\ell\}) > \hat{q}_i(\hat{S})$ is more likely to be satisfied. On the contrary, the closer the coverage of $\hat{S}$ gets to N, the more (SC1$^{\hat{h}_i}$) is likely to dominate (OA+SC1). Moreover, if f is linear or close to being linear in the interval $[\hat{q}_i(\hat{S}), \hat{q}_i(\hat{S} \cup \{\ell\})]$, then

$$\varrho_\ell^{\hat{h}_i}(\hat{S}) = f(\hat{q}_i(\hat{S} \cup \{\ell\})) - f(\hat{q}_i(\hat{S})) \simeq f'(\hat{q}_i(\hat{S})) \left(\hat{q}_i(\hat{S} \cup \{\ell\}) - \hat{q}_i(\hat{S})\right) = f'(\hat{q}_i(\hat{S}))\varrho_\ell^{\hat{q}_i}(\hat{S})$$

and, thus, the difference in tightness of the coefficients of y_ℓ for each $\ell \in \hat{N} \setminus \hat{S}$ (in which (SC1$^{\hat{h}_i}$) is tighter) almost vanishes. A numerical illustration of this observation as well as of Proposition 6 is provided by Example 1 in the Appendix.

Let us now compare Constraints (SC2$^{\hat{h}_i}$) and Constraints (OA+SC2).

Lemma 4 *For each $i \in M$, $\hat{S} \subseteq \hat{N}$, and $\ell \in \hat{N} \setminus \hat{S}$ with $\hat{q}_i(\hat{S}) > \hat{q}_i(\{\ell\})$, the coefficient of y_ℓ in Constraint* (OA+SC2) *is tighter than the coefficient of y_ℓ in Constraint* (SC2$^{\hat{h}_i}$).

Proof As f is concave, differentiable, and increasing and $\hat{q}_i$ is nondecreasing, we have:

$$\varrho_\ell^{\hat{h}_i}(\emptyset) = f(\hat{q}_i(\{\ell\})) - f(\hat{q}_i(\emptyset)) \geq f'(\hat{q}_i(\{\ell\})) \left(\hat{q}_i(\{\ell\}) - \hat{q}_i(\emptyset)\right) = f'(\hat{q}_i(\{\ell\}))\, \varrho_\ell^{\hat{q}_i}(\emptyset).$$

For all $\ell \in \hat{N} \setminus \hat{S}$ such that $\hat{q}_i(\{\ell\}) < \hat{q}_i(\hat{S})$, we have $f'(\hat{q}_i(\{\ell\})) > f'(\hat{q}_i(\hat{S}))$ and, hence, $\varrho_\ell^{\hat{h}_i}(\emptyset) > f'(\hat{q}_i(\hat{S}))\varrho_\ell^{\hat{q}_i}(\emptyset)$. ■

Lemma 5 *For each $i \in M$, $\hat{S} \subseteq \hat{N}$, and $\ell \in \hat{S}$, the coefficient of $(1 - y_\ell)$ in Constraint* (SC2$^{\hat{h}_i}$) *is at least as tight as the coefficient of $(1 - y_\ell)$ of Constraint* (OA+SC2).

Proof Since f is concave and differentiable and $\hat{q}_i$ is nondecreasing, we have:

$$\begin{aligned}\varrho_\ell^{\hat{h}_i}(\hat{S} \setminus \{\ell\}) =& f(\hat{q}_i(\hat{S})) - f(\hat{q}_i(\hat{S} \setminus \{\ell\})) \geq f'(\hat{q}_i(\hat{S})) \left(\hat{q}_i(\hat{S}) - \hat{q}_i(\hat{S} \setminus \{\ell\})\right) = \\ =& f'(\hat{q}_i(\hat{S}))\, \varrho_\ell^{\hat{q}_i}(\hat{S} \setminus \{\ell\}),\end{aligned}$$

thus proving that the coefficient of $(1 - y_\ell)$ in (SC2$^{\hat{h}_i}$) is at least as tight as the one in (OA+SC2). ■

We deduce the following:

Proposition 7 *For each $i \in M$ and $\hat{S} \subseteq \hat{N}$, we have:*

1. *If $\hat{q}_i(\hat{S}) \leq \hat{q}_i(\{\ell\})$ for some $\ell \in \hat{N} \setminus \hat{S}$, Constraints* (SC2$^{\hat{h}_i}$) *and* (OA+SC2) *do not dominate each other.*

2. *If $\hat{q}_i(\hat{S}) > \hat{q}_i(\{\ell\})$ for all $\ell \in \hat{N} \setminus \hat{S}$ and $\hat{q}_i(\hat{S}) = \hat{q}_i(\hat{S} \setminus \{\ell\})$ for all $\ell \in \hat{S}$, then* (OA+SC2) *strictly dominates* (SC2$^{\hat{h}_i}$).

Proof As $\hat{h}_i(\hat{S}) = f(\hat{q}_i(\hat{S}))$, the constant terms of the two inequalities are identical. The first result follows from Lemma 4 and Lemma 5. The second results follows from Lemma 4 and from the fact that, if $\hat{q}_i(\hat{S}) = \hat{q}_i(\hat{S} \setminus \{\ell\})$ holds for each $\ell \in \hat{S}$, then $\varrho_\ell^{\hat{q}_i}(\hat{S} \setminus \{\ell\}) = 0$, and so the coefficient next to $(1 - y_\ell)$ is equal to zero in both inequalities. ∎

A numerical illustration of Proposition 7 is provided by Example 2 in the Appendix.

We conclude the section with the following result:

Proposition 8 *In the general case, Constraints* (OA+SC1) *and* (OA+SC2) *do not dominate each other.*

Proof Consider an instance with $N := \{1, \ldots, 6\}$, $\hat{N} := \{1, \ldots, 5\}$, $G = (N \cup \hat{N}, E)$ with $N(1) = \{2\}$, $N(2) = \{2, 3\}$, $N(3) = \{3\}$, $N(4) = \{1, 4\}$, $N(5) = \{2, 3, 5, 6\}$, $m = 1$, $f(z) = 1 - e^{-\frac{1}{\lambda} z}$ with $\lambda = 10$, and $a = (0.7, 0.6, 0.5, 0.4, 0.3, 0.2)$. Let $\hat{S} = \{1, 2\}$. The (OA+SC1) and (OA+SC2) inequalities read:

$$w \le 0.105 - 0.000(1 - y_1) - 0.000(1 - y_2) + 0.000 y_3 + 0.099 y_4 + 0.045 y_5$$
$$w \le 0.105 - 0.000(1 - y_1) - \mathbf{0.044}(1 - y_2) + \underline{0.045} y_3 + 0.099 y_4 + \underline{0.144} y_5.$$

As the latter is tighter at the coefficient of $(1 - y_2)$ and weaker at the coefficients of y_3 and y_5 (underlined), the two inequalities do not dominate each other. ∎

4.2 Comparison between the inequalities in Ref(OA+SC) and Ref(OA+BC)

Let us compare the inequalities (OA+SC1)–(OA+SC2) and (OA+BC1)–(OA+BC2) featured in Ref(OA+SC) and Ref(OA+BC) for a given $i \in M$ and $\hat{S} \subseteq \hat{N}$—we recall that (OA+BC1)–(OA+BC2) are defined also for extreme vertices that, due to not corresponding to an integer $\tilde{y}$, do not correspond to a set $\hat{S} \subseteq \hat{N}$. Since these inequalities are obtained by combining (SC1$^{\hat{q}_i}$), (SC2$^{\hat{q}_i}$), and (BC) with the outer-approximation constraint (OA), comparing them requires understanding the relationship between the submodular cuts (SC1$^{\hat{q}_i}$), (SC2$^{\hat{q}_i}$) and the Benders cut (BC).

For the purpose, we first consider an additional submodular cut:

$$\hat{\mathcal{Q}}_i(y) \le \hat{q}_i(\hat{S}) + \sum_{\ell \in \hat{N} \setminus \hat{S}} \varrho_\ell^{\hat{q}_i}(\hat{S}) y_\ell \qquad i \in M, \hat{S} \subseteq \hat{N}. \tag{SC3$^{\hat{q}_i}$}$$

Similarly to what we observed in Section 2 when comparing (SC3$^{\hat{h}_i}$) to (SC1$^{\hat{h}_i}$), (SC3$^{\hat{q}_i}$) and (SC1$^{\hat{q}_i}$) are equally strong for a given $i \in M$ and $\hat{S} \subseteq \hat{N}$ unless $N_1 \neq \emptyset$ and there is an item $j \in N$ which is covered by a single metaitem $\ell \in \hat{N}$, i.e., such that $|\hat{N}(j)| = 1$, and $a_{ij} > 0$. We can establish the following:

Proposition 9 *For each pair of $i \in M$ and $\hat{S} \subseteq \hat{N}$, the submodular cut* (SC3$^{\hat{q}_i}$) *coincides with the Benders cut* (BC) *obtained with dual multipliers defined as follows:*

$$\tilde{\delta}_j = \begin{cases} a_{ij} & \textit{if } j \notin N(\hat{S}) \\ 0 & \textit{otherwise} \end{cases} \qquad \tilde{\sigma}_j = \begin{cases} a_{ij} & \textit{if } j \in N(\hat{S}) \\ 0 & \textit{otherwise} \end{cases} \qquad j \in N. \tag{10}$$

Proof Observe that, due to Lemma 1, $(\tilde{\delta}, \tilde{\sigma})$ is an extreme point belonging to $\mathcal{P}_i^e$, obtained by setting $\gamma_j := 0$ for all $j \in N$. By the choice of multipliers in (10), we have $\sum_{j \in N} \tilde{\sigma}_j = \sum_{j \in N(\hat{S})} a_{ij}$. Thus, we have $d_i + \sum_{j \in N} \tilde{\sigma}_j = \hat{q}_i(\hat{S})$. Furthermore, for each $\ell \in \hat{N}$, the coefficient of y_ℓ (i.e., $\sum_{j \in N(\ell)} \tilde{\delta}_j$) is equal to $\sum_{j \in N(\ell) \setminus N(\hat{S})} a_{ij}$, which corresponds to $\varrho_\ell^{\hat{q}_i}(\hat{S})$, i.e., to the contribution of all the items in N not covered by $\hat{S}$ but which would be covered by adding ℓ to $\hat{S}$. The claim follows. ■

In the general case where N_1 is not necessarily empty, Constraints (SC3$^{\hat{q}_i}$) are dominated by (SC1$^{\hat{q}_i}$). The following result shows that Constraints (SC1$^{\hat{q}_i}$) are subsumed by the constraints contained in Ref(OA+BC):

Proposition 10 *For each pair of $i \in M$ and $\hat{S} \subseteq \hat{N}$, the submodular cut* (SC1$^{\hat{q}_i}$) *is equal to the following special case of the Benders cut* (BC)*:*

$$\hat{\mathcal{Q}}_i(y) \le \hat{q}_i(\hat{S}) + \sum_{\ell \notin \hat{S}} \left(\sum_{j \in N(\ell) \setminus N(\hat{S})} a_{ij} \right) y_\ell - \sum_{\ell \in \hat{S}} \left(\sum_{j \in N(\ell) \cap N_1} a_{ij} \right) (1 - y_\ell), \tag{BC1}$$

obtained from Lemma 1 by letting $\gamma_j = 1$ for all $j \in N_1$, i.e., with the following choice of multipliers:

$$\tilde{\delta}_j = \begin{cases} a_{ij} & \text{if } j \notin N(\hat{S}) \text{ or } j \in N_1 \\ 0 & \text{otherwise} \end{cases} \qquad \tilde{\sigma}_j = \begin{cases} a_{ij} & \text{if } j \in N(\hat{S}) \setminus N_1 \\ 0 & \text{otherwise} \end{cases} \qquad j \in N. \tag{11}$$

Proof Observe that, as a consequence of Lemma 1, the pair $(\tilde{\delta}, \tilde{\sigma})$ in Equation (11) corresponds to the extreme solution to the dual of the Benders subproblem defined in (9) that is obtained by setting $\gamma_j := 1$ if $j \in N_1$ and $\gamma_j := 0$ otherwise. Due to (11), we have:

$$d_i + \sum_{j \in N} \tilde{\sigma}_j = d_i + \sum_{j \in N(\hat{S}) \setminus N_1} a_{ij} = \hat{q}_i(\hat{S}) - \sum_{j \in N(\hat{S}) \cap N_1} a_{ij} \tag{12}$$

$$\sum_{j \in N(\ell)} \tilde{\delta}_j = \begin{cases} \sum_{j \in N(\ell) \setminus N(\hat{S})} a_{ij} & \ell \in \hat{N} \setminus \hat{S} \\ \sum_{j \in N(\ell) \cap N_1} a_{ij} & \ell \in \hat{S}. \end{cases} \tag{13}$$

Thanks to (12) and (13), (BC1) can be rewritten as:

$$\hat{\mathcal{Q}}_i(y) \le \hat{q}_i(\hat{S}) - \sum_{j \in N(\hat{S}) \cap N_1} a_{ij} + \sum_{\ell \notin \hat{S}} \left(\sum_{j \in N(\ell) \setminus N(\hat{S})} a_{ij} \right) y_\ell + \sum_{\ell \in \hat{S}} \left(\sum_{j \in N(\ell) \cap N_1} a_{ij} \right) y_\ell. \tag{14}$$

Similarly to the proof of Proposition 9, the coefficient of y_ℓ in the third term is the marginal contribution of $\ell \in \hat{N}$ with respect to $\hat{S}$. As the summation in the second term only considers items $j \in N(\hat{S})$ which belong to N_1, we have:

$$\sum_{j \in N(\hat{S}) \cap N_1} a_{ij} = \sum_{\ell \in \hat{S}} \sum_{j \in N(\ell) \cap N_1} a_{ij}.$$

Thus, we can reorder the terms of the last two summations in (14), obtaining (BC1). By noting that

$$\varrho_\ell^{\hat{q}_i}(\hat{N} \setminus \{\ell\}) = \sum_{j \in N(\ell) \cap N_1} a_{ij},$$

the proof is concluded. ■

Proposition 10 implies that, for each $i \in M$ and $\hat{S} \subseteq \hat{N}$, Constraint (OA+SC1) coincides with the following special case of the (OA+BC) constraint:

$$w_i \le f(\hat{q}_i(\hat{S})) + \sum_{\ell \in \hat{N} \setminus \hat{S}} f'(\hat{q}_i(\hat{S})) \Bigg(\sum_{j \in N(\ell) \setminus N(\hat{S})} a_{ij} \Bigg) y_\ell - \sum_{\ell \in \hat{S}} f'(\hat{q}_i(\hat{S})) \Bigg(\sum_{j \in N(\ell) \cap N_1} a_{ij} \Bigg) (1 - y_\ell)$$
$$\hat{S} \subseteq \hat{N}, i \in M. \qquad \text{(OA+BC1)}$$

The following proposition shows that the submodular constraint in (SC2$^{\hat{q}_i}$) corresponding to each pair of $i \in M$ and $\hat{S} \subseteq \hat{N}$ is strictly dominated by the Benders cuts in (BC) obtained according to Lemma 1 for y equal to the characteristic vector of $\hat{S}$ and $\gamma_j := 1$ for all $j \in N$.

Proposition 11 *The submodular cut* (SC2$^{\hat{q}_i}$) *corresponding to each pair of* $i \in M$ *and* $\hat{S} \subseteq \hat{N}$ *is dominated by the following special case of the Benders cut* (BC)*:*

$$\hat{\mathcal{Q}}_i(y) \le d_i + \sum_{j \in N(\hat{S})} a_{ij} + \sum_{\ell \in \hat{N} \setminus \hat{S}} \Bigg(\sum_{j \in N(\ell) \setminus N(\hat{S})} a_{ij} + \sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij} \Bigg) y_\ell$$
$$- \sum_{\ell \in \hat{S}} \Bigg(\sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij} \Bigg) (1 - y_\ell), \qquad \text{(BC2)}$$

which is strictly stronger than (SC2$^{\hat{q}_i}$) *if* $\exists \ell \in \hat{N} \setminus \hat{S} : \varrho_\ell^{\hat{q}_i}(\emptyset) > \varrho_\ell^{\hat{q}_i}(\hat{S}) + \sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij}$*. The cut is obtained for the following choice of dual multipliers, obtained from Lemma 1 by setting* $\gamma_j = 1$ *for all* $j \in N$*:*

$$\tilde{\delta}_j = \begin{cases} a_{ij} & \text{if } j \notin N(\hat{S}) \text{ or } j \in N_1(\hat{S}) \\ 0 & \text{otherwise} \end{cases} \qquad \tilde{\sigma}_j = \begin{cases} a_{ij} & \text{if } j \in N(\hat{S}) \setminus N_1(\hat{S}) \\ 0 & \text{otherwise} \end{cases} \qquad j \in N. \quad (15)$$

Proof With the choice of γ in (15), we rewrite (BC) as:

$$\hat{\mathcal{Q}}_i(y) \le \quad d_i + \sum_{j \in N(\hat{S}) \setminus N_1(\hat{S})} a_{ij} + \sum_{\ell \in \hat{N} \setminus \hat{S}} \Bigg(\sum_{j \in N(\ell) \setminus N(\hat{S})} a_{ij} + \sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij} \Bigg) y_\ell + \sum_{\ell \in \hat{S}} \Bigg(\sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij} \Bigg) y_\ell. \tag{16}$$

Observe that $\varrho_\ell^{\hat{q}_i}(\emptyset) = \sum_{j \in N(\ell)} a_{ij}$ and $\varrho_\ell^{\hat{q}_i}(\hat{S} \setminus \{\ell\}) = \sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij}$. Thus, the right-hand side of (SC2$^{\hat{q}_i}$) can be rewritten as:

$$d_i + \sum_{j \in N(\hat{S})} a_{ij} + \sum_{\ell \in \hat{N} \setminus \hat{S}} \Bigg(\sum_{j \in N(\ell)} a_{ij} \Bigg) y_\ell - \sum_{\ell \in \hat{S}} \Bigg(\sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij} \Bigg) (1 - y_\ell),$$

which, noting that $\sum_{j \in N(\hat{S})} a_{ij} - \sum_{\ell \in \hat{S}} \sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij} = \sum_{j \in N(\hat{S}) \setminus N_1(\hat{S})} a_{ij}$, can be restated as:

$$d_i + \sum_{j \in N(\hat{S}) \setminus N_1(\hat{S})} a_{ij} + \sum_{\ell \in \hat{N} \setminus \hat{S}} \Bigg(\sum_{j \in N(\ell)} a_{ij} \Bigg) y_\ell + \sum_{\ell \in \hat{S}} \Bigg(\sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij} \Bigg) y_\ell,$$

which is larger than the right-hand side of (16). By rewriting $\sum_{j \in N(\hat{S}) \setminus N_1(\hat{S})} a_{ij}$ as $\sum_{j \in N(\hat{S})} a_{ij} - \sum_{\ell \in \hat{S}} \sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij}$ and collecting y_ℓ for all $\ell \in \hat{S}$, (16) is rewritten as (BC2), concluding the proof. ∎

Proposition 11 shows that, for each $i \in M$ and $\hat{S} \subseteq \hat{N}$, Constraint (OA+SC2) is dominated by the following (OA+BC) constraint:

$$w_i \leq f(\hat{q}_i(\hat{S})) + \sum_{\ell \in \hat{N} \setminus \hat{S}} f'(\hat{q}_i(\hat{S})) \Bigg(\sum_{j \in N(\ell) \setminus N(\hat{S})} a_{ij} + \sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij} \Bigg) y_\ell$$
$$- \sum_{\ell \in \hat{S}} f'(\hat{q}_i(\hat{S})) \Bigg(\sum_{j \in N(\ell) \cap N_1(\hat{S})} a_{ij} \Bigg) (1 - y_\ell) \qquad \hat{S} \subseteq \hat{N}, i \in M \tag{OA+BC2}$$

We conclude the section by showing the following:

Proposition 12 *In the general case, Constraints* (OA+BC1) *and* (OA+BC2) *do not dominate each other.*

Proof Consider the instance used in the proof of Proposition 8 and let $\hat{S} = \{1, 2\}$. The (OA+BC1) and (OA+BC2) inequalities read:

$$w \leq 0.105 - 0.000(1 - y_1) - 0.000(1 - y_2) + 0.000 y_3 + 0.099 y_4 + 0.045 y_5$$
$$w \leq 0.105 - 0.000(1 - y_1) - \mathbf{0.044}(1 - y_2) + \underline{0.045} y_3 + 0.099 y_4 + \underline{0.089} y_5.$$

As the latter is tighter at the coefficient of $(1 - y_2)$ and weaker at the coefficients of y_3 and y_5 (underlined), the two inequalities do not dominate each other. ∎

4.3 Comparison of the inequalities in Ref(SC) and Ref(OA+BC)

Since, due to Proposition 10, Constraint (OA+SC1) and (OA+BC1) coincide, we have the following:

Corollary 1 *For each $i \in M$ and $\hat{S} \subseteq \hat{N}$, Constraints* (SC1$^{\hat{h}_i}$) *and* (OA+BC1) *enjoy the same relationship as Constraints* (SC1$^{\hat{h}_i}$) *and* (OA+SC1)*, as reported in Proposition 6.*

Since, due to Proposition 11, Constraint (OA+BC2) dominate Constraints (OA+SC2), we deduce the following:

Corollary 2 *For each $i \in M$ and $\hat{S} \subseteq \hat{N}$, Constraints* (SC2$^{\hat{h}_i}$) *and* (OA+BC2) *enjoy the same relationship as Constraints* (SC2$^{\hat{h}_i}$) *and* (OA+SC2)*, as reported in Proposition 7 and, in the general case, Constraints* (SC2$^{\hat{h}_i}$) *and* (OA+BC2) *do not dominate each other.*

Proof Consider the instance used in the proof of Proposition 8 and let $\hat{S} = \{1, 2, 4\}$, Constraints (SC2$^{\hat{h}_i}$) and Constraints (OA+BC2) read:

$$w \leq 0.198 - 0(1 - y_1) - 0.041(1 - y_2) + 0.049 y_3 - 0.093(1 - y_4) + 0.148 y_5$$
$$w \leq 0.198 - 0(1 - y_1) - \underline{0.040}(1 - y_2) + \mathbf{0.041} y_3 - \underline{0.088}(1 - y_4) + \mathbf{0.081} y_5.$$

The letter inequality is tighter at the coefficients of y_3 and y_5 (in boldface) but weaker at those of $(1 - y_2)$ and $(1 - y_4)$ (underlined) and, thus, the two inequalities do not dominate each other. ∎

4.4 Summary of the theoretical results on the relative strength of the inequalities

Before providing a summary of the theoretical results derived in this section, we consider a relevant special case in which the function f is assumed to be affine.

Proposition 13 *If f is an affine function, for each $i \in M$ and $\hat{S} \subseteq \hat{N}$ Constraint* (OA+BC1) *coincides with Constraint* (SC1$^{\hat{h}_i}$)*, and Constraint* (OA+BC2) *dominates Constraint* (SC2$^{\hat{h}_i}$)*.*

Proof Let $f(z) := \alpha z + \beta$ for some $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}$. Then, $f'(\hat{q}_i(\hat{A})) = \alpha$ for all $\hat{A} \subseteq \hat{N}$. Thus, $\varrho_\ell^{\hat{h}_i}(\hat{B}) = \alpha \varrho_\ell^{\hat{q}_i}(\hat{B}) = f'(\hat{q}_i(\hat{A}))\varrho_\ell^{\hat{q}_i}(\hat{B})$ for all $\ell \in \hat{N}$ and $\hat{A}, \hat{B} \subseteq \hat{N}$. It follows that (OA+SC1) coincides with (SC1$^{\hat{h}_i}$), and that (OA+SC2) coincides with (SC2$^{\hat{h}_i}$). Thus, the claim follows due to the fact that (OA+BC1) coincides with (OA+SC1) due to Proposition 10, and due to the fact that (OA+BC2) dominates with (OA+SC2) due to Proposition 11. ■

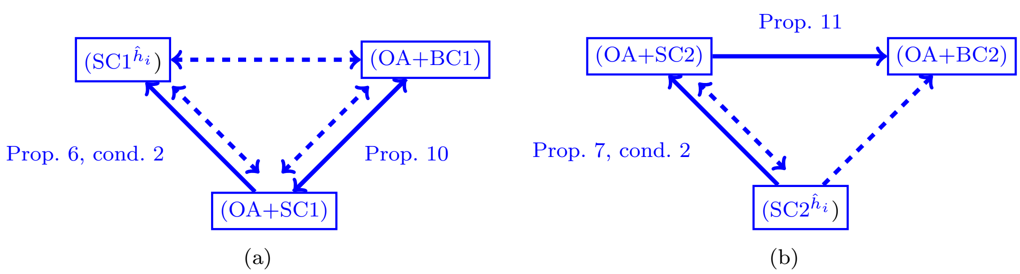

Fig. 3: Relative strength of the (new) inequalities, where an arrow pointing from A to B indicates that A is strictly dominated by B (subject to some conditions) and an arrow in both directions indicates that the two inequalities are identical. Dashed lines refer to the case when f is affine (see Proposition 13). Solid lines refer to the general case, and the proposition corresponding to each relationship is reported next to its arrow.

A visual summary of the relative strength of the families of inequalities studied in this paper is provided in Figure 3.

4.5 The special case of Problem (1)

We conclude this section by considering the application of our inequalities to Problem (1) of [Ahmed and Atamtürk, 2011, Yu and Ahmed, 2017]. To solve Problem (1) by means of our reformulations, we must, first, cast it as an instance of Problem (2). We set $\hat{N} = N$ and define the edge set E of the graph G as a matching. In such a case where G is the matching graph, the sets N and $\hat{N}$ enjoy a bijection relationship thanks to which each $\ell \in \hat{N}$ is mapped into a single $j = j(\ell) \in N$ and each $j \in N$ is mapped into a single $\ell = \ell(j) \in \hat{N}$. Thus, we have $x_j = y_{\ell(j)}$ and $x_{j(\ell)} = y_\ell$ for each $j \in N$ and $\ell \in \hat{N}$. This implies $\hat{\mathcal{Q}}_i(y) = \sum_{j \in N} a_{ij} x_j + d_i$, which, for $S := \{j \in N : x_j = 1\}$, coincides with function $\hat{q}_i$ of Problem (2), as well as with function q_i of Problem (1). Since, as a consequence, $\hat{\mathcal{Q}}_i, \hat{q}_i$, and q_i are linear functions, the application of either the submodular cuts (SC1$^{\hat{q}_i}$)–(SC2$^{\hat{q}_i}$) to $\hat{q}_i$ or the Benders cut (BC) to $\hat{\mathcal{Q}}(y)$ would only result in rewriting the hypograph constraint (HYPO2) as the trivial inequality $\eta_i \le \sum_{j \in N} a_{ij} x_j + d_i$. It follows that the two reformulations Ref(OA+SC) and Ref(OA+BC) coincide with the following outer-approximation reformulation, to which we refer as Ref(OA):

$$\max_{\substack{w\in\mathbb{R}^m\\ x\in X\cap\{0,1\}^n}} \quad \sum_{i\in M}\pi_i w_i \tag{17a}$$

$$\text{s.t.} \quad w_i \le f(p) - f'(p)p + f'(p)\left(d_i + \sum_{j\in N} a_{ij}x_j\right) \quad i\in M, p\in\left[0, \sum_{j\in N} a_{ij} + d_i\right]. \tag{17b}$$

We notice that, given a (fractional) solution x^*, the separation problem for Constraints (17b) is solved by simply setting $p := \sum_{j\in N} a_{ij}x^*_j + d_i$.

In [Yu and Ahmed, 2017], the authors propose a version, stronger for the case where the problem is subject to a single knapsack constraint, of the inequalities obtained by subadditive, sequence-independent lifting introduced in Ahmed and Atamtürk [2011] by exploiting the submodularity of the objective function $h = f\circ q_i$, $i\in M$—see Section 1.

In particular, Yu and Ahmed [2017] propose a branch-and-cut algorithm based on the separation of the uplifted version of the submodular constraint (SC2$^{\hat{h}_i}$)—inequality (9) reported on page 153 of Yu and Ahmed [2017], as well as on the separation of the standard submodular inequality (SC1$^{\hat{h}_i}$).

For comparing the inequalities considered in [Yu and Ahmed, 2017] subject to a knapsack constraint ($\sum_{j\in N}\beta_j x_j \le B$) and employing the submodular function (f) to those featured in Ref(OA), we illustrate the following two results.

Proposition 14 *In the general case, for a given $i\in M$ and $S\subseteq N$, Constraints* (SC1$^{\hat{h}_i}$) *and Constraints* (17b) *do not dominate each other.*

Proof Consider the following instance, taken from [Yu and Ahmed, 2017]. Let G be the matching graph with $n = \hat{n} = 6$. Consider a single scenario. Assume $f = -e^{-ax}$. Consider a knapsack constraint with budget $B = 1$ and weights and item values defined as:

$$\beta = (0.3023, 0.1892, 0.3884, 0.1047, 0.5938, 0.6699)$$
$$a = (0.3008, 0.3621, 0.4233, 0.6395, 0.1164, 0.0448).$$

For $S = \{1,2\}$, the (SC1$^{\hat{h}_i}$) constraint considered in [Yu and Ahmed, 2017] and the (17b) inequality in Ref(OA) corresponding to the same set read:

$$w \le -0.515 - 0.053(1-x_1) - 0.066(1-x_2) + 0.178x_3 + 0.244x_4 + 0.057x_5 + 0.023x_6$$
$$w \le -0.515 - \mathbf{0.155}(1-x_1) - \mathbf{0.186}(1-x_2) + \underline{0.218}x_3 + \underline{0.330}x_4 + \underline{0.060}x_5 + \underline{0.024}x_6.$$

Since the latter inequality is tighter at the coefficients of $(1-x_1)$ and $(1-x_2)$ (highlighted in boldface) and weaker at the coefficients of x_3, x_4, x_5, x_6 (underlined), the two inequalities do not dominate each other. ■

Proposition 15 *In the general case, for a given $i\in M$ and $S\subseteq N$, the lifted version of Constraint* (SC2$^{\hat{h}_i}$) *proposed in Yu and Ahmed [2017] and Constraints* (17b) *do not dominated each other.*

Proof Consider the same example as in the proof of Proposition 14. For $S = \{1,2,3,4\}$, the inequality obtained with the lifting procedure in [Yu and Ahmed, 2017] (inequality (9) in the paper) and the Constraint (17b) in Ref(OA) read:

$$w \le -0.178 - 0.062(1-x_1) - 0.077(1-x_2) - 0.093(1-x_3) - 0.159(1-x_4) + 0.037x_5 + 0.017x_6$$
$$w \le -0.178 - \underline{0.053}(1-x_1) - \underline{0.064}(1-x_2) - \underline{0.075}(1-x_3) - \underline{0.113}(1-x_4) + \mathbf{0.021}x_5 + \mathbf{0.008}x_6.$$

Since the latter inequality is tighter than the previous one at the coefficients of $(1-x_1), (1-x_2), (1-x_3), (1-x_4)$ (underlined) and weaker at the coefficients of x_5 and x_6 (highlighted in boldface), we conclude that the two inequalities do not dominate each other. ■

Computational experiments comparing the impact of these inequalities on solving instances of Problem (1) are reported in Section 7.

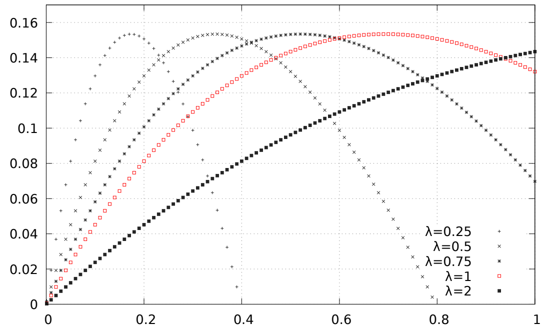

Fig. 4: The not nondecreasing utility function $f(z) = 1 - e^{-\frac{z}{\lambda}} - \alpha\,\frac{z}{\lambda}$, with $\alpha = 0.5$.

5. Extension to the case where f is not nondecreasing

In this section, we discuss an extension of our methods to the case where f is concave and differentiable, but not necessarily increasing. Such a case arises in applications where, e.g., f is composed of two terms: a revenue term which increases with the amount of customers, with a positive decreasing derivative due to economies of scale, and a negative component which accounts for the total amount of money spent for running the service, assuming a unit cost per customer which decreases with the number of customers. In such cases, once the system becomes overloaded and the optimal running capacity is overreached, f changes from being increasing to being decreasing.

As an example, consider the following strictly concave but not nondecreasing function:

$$f(z) = \left(1 - \exp\left(-\frac{z}{\lambda}\right)\right) - \alpha\,\frac{z}{\lambda}, \tag{f-n-nd}$$

where λ is the risk-tolerance parameter and $\alpha > 0$ is an additional parameter establishing a trade-off between the increasing and the decreasing terms. An illustration of this function for different values of λ is provided in Figure 4.

When combining a function f such as the one defined (f-n-nd) with the set function $\hat{q}_i$, the obtained function $\hat{h}_i := f \circ \hat{q}_i$ ($\hat{h}_i(\hat{S}) := 1 - e^{-\frac{\hat{q}_i(\hat{S})}{\lambda}} - \alpha\,\frac{\hat{q}_i(\hat{S})}{\lambda}$ for the case of (f-n-nd)) is the difference of two increasing submodular functions, which, in the general case, is not submodular. As a consequence, the Ref(SC) and the submodular constraints in Proposition 1 are not valid for Problem (2). This, in particular, implies that the inequalities introduced in [Ahmed and Atamtürk, 2011] and [Yu and Ahmed, 2017] are not valid for it either.

To account for this case, we need to modify the double-hypograph decomposition we proposed in Section 3.1 as follows:

$$\max_{\substack{w,\eta \in \mathbb{R}^m \\ y \in Y \cap \{0,1\}^{\hat{n}}}} \quad \sum_{i \in M} \pi_i w_i$$

$$w_i \leq f(\eta_i) \qquad i \in M \tag{HYPO1'}$$

$$\eta_i = \hat{\mathcal{Q}}_i(y) \qquad i \in M, \tag{HYPO2'}$$

where the inequality sign "$\leq$" in Constraints (HYPO2) has been replaced by the equality sign "$=$" in Constraints (HYPO2'). Such an inequality is necessary as, due to $f(\eta_i)$ being decreasing for a sufficiently large η_i, the constraint $\eta_i \leq \hat{\mathcal{Q}}_i(y)$ is not guaranteed to be satisfied as an equation in any optimal solution.

We can reformulate Constraints (HYPO2') linearly thanks to the following result (in which LBSC stands for Lower Bounding Submodular Cut):

Proposition 16 *Constraints* (HYPO2') *can be replaced by* (SC1$^{\hat{q}_i}$)–(SC2$^{\hat{q}_i}$) *together with the following constraints:*

$$\eta_i \geq \hat{q}_i(\hat{S}) + \sum_{\ell \notin \hat{S}} \varrho_\ell^{\hat{q}_i}(\hat{N} \setminus \{\ell\}) y_\ell - \sum_{\ell \in \hat{S}} \varrho_\ell^{\hat{q}_i}(\emptyset)(1 - y_\ell) \qquad i \in M, \hat{S} \subseteq \hat{N}. \qquad \text{(LBSC)}$$

Proof To deal with the equations in Constraints (HYPO2'), we need to introduce not just over- but also under-estimator for the (submodular) function $\hat{\mathcal{Q}}_i(y)$. Two over-estimators are given by Constraints (SC1$^{\hat{q}_i}$) and (SC2$^{\hat{q}_i}$). To build an under-estimator, we rely on Proposition 2.1 of Nemhauser et al. [1978] which states that, for a submodular function $\hat{q}_i$, the following inequality holds:

$$\hat{q}_i(\hat{S}) \leq \hat{q}_i(\hat{T}) + \sum_{\ell \in \hat{S} \setminus \hat{T}} \varrho_\ell^{\hat{q}_i}(\hat{S} \cap \hat{T}) - \sum_{\ell \in \hat{T} \setminus \hat{S}} \varrho_\ell^{\hat{q}_i}(\hat{T} \setminus \ell) \qquad \hat{S}, \hat{T} \subseteq \hat{N}.$$

Constraint (LBSC) is obtained by substituting the marginals with the following upper and lower bounds

$$\varrho_\ell^{\hat{q}_i}(\hat{S} \cap \hat{T}) \leq \varrho_\ell^{\hat{q}_i}(\emptyset); \qquad \varrho_\ell^{\hat{q}_i}(\hat{T} \setminus \ell) \geq \varrho_\ell^{\hat{q}_i}(\hat{N} \setminus \ell),$$

and by encoding the set $\hat{T}$ via the binary vector y. Since such constraints are tight whenever y corresponds to the characteristic vector of $\hat{S}$, the claim follows. ■

We can linearize Constraints (HYPO1') following an outer approximation method similar to the one proposed in Section 3.2. Following the derivations in the proof of Proposition 2, and after applying Fourier-Mötzkin elimination to project out the η_i variables, we notice that Constraints (OA+SC1)–(OA+SC2) are valid only for $\hat{S} \subseteq \hat{N} : f'(\hat{q}_i(\hat{S})) \geq 0$—notice that, if $f'(\hat{q}_i(\hat{S})) = 0$, both constraints boil down to $w_i \leq f(\hat{q}_i(\hat{S}))$. For $\hat{S} \subseteq \hat{N} : f'(\hat{q}_i(\hat{S})) < 0$, we introduce the following constraints derived from Constraints (LBSC) of Proposition 16:

$$w_i \geq f(\hat{q}_i(\hat{S})) + \sum_{\ell \notin \hat{S}} f'(\hat{q}_i(\hat{S})) \varrho_\ell^{\hat{q}_i}(\hat{N} \setminus \{\ell\}) y_\ell - \sum_{\ell \in \hat{S}} f'(\hat{q}_i(\hat{S})) \varrho_\ell^{\hat{q}_i}(\emptyset)(1 - y_\ell)$$
$$i \in M, \hat{S} \subseteq \hat{N} : f'(\hat{q}_i(\hat{S})) < 0. \qquad \text{(OA+LBSC)}$$

The final reformulation that we obtain, which we refer to by Ref(OA+SC+LBSC), features Constraints (OA+SC1), OA+SC2, and (OA+LBSC).

Finally, we point out that we can also substitute Constraints (OA+BC1) and (OA+BC2) for (OA+SC1) and (OA+SC2), obtaining a reformulation featuring (OA+BC1), OA+BC2, and (OA+LBSC), to which we refer by Ref(OA+BC+LBSC). Experiments carried out with Ref(OA+BC+LBSC) assuming the (not nondecreasing) utility function f defined in (f-n-nd) are reported in Section 7.

6. Greedy algorithm and worst-case instances

A simple and effective way to obtain feasible solutions when maximizing a submodular function subject to a k-cardinality constraint is represented by the *greedy algorithm* proposed in Nemhauser et al. [1978]. This algorithm computes a solution whose value is at least $1 - 1/e$ times the optimal solution value of the instance, where e is the base of the natural logarithm. For the basic variant of Problem (2) with a single k-cardinality constraint, the greedy algorithm starts with an empty set of metaitems ($\hat{S} = \emptyset$), and, at each iteration, the set $\hat{S}$ is enlarged with the metaitem

$$\ell \in \arg\max_{\ell \notin \hat{S}} \sum_{i \in M} \pi_i \, \varrho_\ell^{\hat{h}_i}(\hat{S}),$$

until k metaitems are selected. In other words, at each iteration the algorithm selects the metaitem with the largest marginal contribution with respect to the objective function. In

Section (7.1.1), we discuss the extensions of the greedy algorithm employed for the different constraints tested for Problem (2).

Greedy algorithms for maximizing a monotone increasing submodular function subject to different combinatorial constraints have been actively studied since the seminal work of Nemhauser et al. [1978]. In Conforti and Cornuéjols [1984] the notion of (total) *curvature* of a submodular function has been introduced. In the same paper, it is shown that, when the function has total curvature c, the greedy algorithm achieves an approximation factor of $(1 - \frac{1}{e^c})\frac{1}{c}$. In Sviridenko et al. [2017], a combination of the greedy algorithm with a local-search technique has been proposed which allows for approximating the more general problem of maximizing a nondecreasing submodular function subject to a single matroid constraint achieving an approximation factor of $(1 - \frac{c}{e})$. Considering a single knapsack constraint, a $1 - \frac{1}{e} - \epsilon$ approximation algorithm is proposed in Ene and Nguyen [2019] (for $\epsilon > 0$).

The quality of the greedy solutions is typically quite good in practice but there are situations in which the greedy algorithm delivers solutions of poor quality. In the remainder of the section, we show how to construct *worst-case instances* for Problem (2), i.e., instances on which the greedy algorithm asymptotically achieves the worst-case approximation factor of $1 - 1/e$. We consider the case where Problem (2) is subject to a k-cardinality constraint and the function f is defined as (f).

Letting $m = 1$ and $f(z) := z$, Problem (2) with a k-cardinality constraint corresponds to the Maximum Coverage Problem (MCP), see, e.g., [Cordeau et al., 2019]. A procedure for creating worst-case instances for the MCP can be found in [Hochbaum and Pathria, 1998]. Here, we show how to adapt such a procedure to create worst-case instances for Problem (2). Let us introduce the matrix $U = [u_{st}]$ with $s = 0, 1, \ldots, k$ and $t = 1, 2, \ldots, k$, for some $k \in \mathbb{N}$. This matrix has $k + 1$ rows and k columns with entries:

$$\text{Row } 0 \to u_{0t} := \begin{cases} k-1, & t = 1, \\ k-2, & \text{otherwise} \end{cases} \qquad \text{Row } 1 \to u_{1t} := \begin{cases} 0, & t = 1, \\ 1, & \text{otherwise} \end{cases}$$

$$\text{Row s} \to u_{st} := \left(\frac{k}{k-1}\right)^{s-2}, \quad 2 \le s \le k, \quad 1 \le t \le k.$$

Each entry (s, t) of this matrix is associated with an item j (there are $k\,(k+1)$ items in N in total) whose objective function coefficients are set to:

$$a_{1j} := \frac{u_{st}}{\Delta} \qquad \text{where} \quad \Delta := k\,(k-1)\left(\frac{k}{k-1}\right)^{k-1}.$$

The constant Δ is equal to the sum of all the entries of the matrix U. In addition, we introduce $2k$ metaitems defined as follows. The first k metaitems, which we denote by $\mathcal{R} = \{r_1, r_2, \ldots, r_k\}$, correspond to the rows $1, 2, \ldots, k$ of the matrix U; each metaitem r_s, $s = 1, \ldots, k$, covers all the items of row s. The second k metaitems, which we denote by $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$, correspond to the columns $1, 2, \ldots, k$ of the matrix U; each metaitem c_t, $t = 1, \ldots, k$, covers all the items of column t. Overall, the set of items is $N = \{(s, t) : s = 0, \ldots, k, t = 1, \ldots, k\}$ and the set of metaitems is $\hat{N} = \{\mathcal{R}\} \cup \{\mathcal{C}\}$.

With f defined as $f(z) := 1 - e^{-\frac{z}{\lambda}}$, the optimal solution value is $Z = 1 - e^{-\frac{1}{\lambda}}$. While an optimal solution of this value is obtained by selecting all k metaitems in $\mathcal{C}$, the greedy algorithm constructs one by picking (in this order) the metaitems $r_k, r_{k-1}, \ldots, r_1$, which results in a greedy solution of value equal to $Z^G = 1 - e^{-\frac{\Delta - (k-1)^2}{\Delta\,\lambda}}$. Hence, we have:

$$\lim_{\lambda \to \infty} \frac{Z^G}{Z} = \lim_{\lambda \to \infty} \frac{1 - e^{-\frac{\Delta - (k-1)^2}{\Delta\,\lambda}}}{1 - e^{-\frac{1}{\lambda}}} = \lim_{\lambda \to \infty} \frac{\frac{\Delta - (k-1)^2}{\Delta \lambda}}{\frac{1}{\lambda}} = \frac{\Delta - (k-1)^2}{\Delta} = 1 - \left(1 - \frac{1}{k}\right)^k,$$

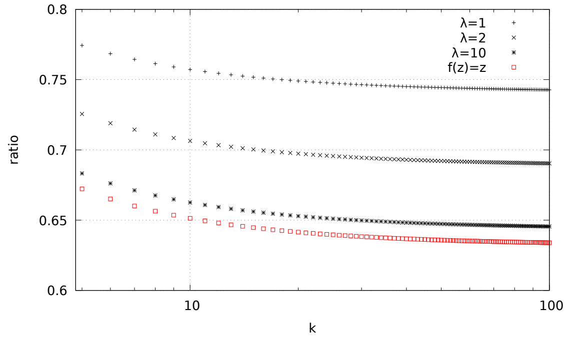

Fig. 5: Ratio between the greedy heuristic solution value Z^G and the optimal solution value Z for the worst-case instances with the utility function $f(z) = 1 - e^{-\frac{z}{\lambda}}$ for different values of $\lambda \in \{1, 2, 10\}$, and also considering the identity utility function $f(z) = z$. The horizontal axis reports in logarithmic scale the value of $k \in \{5, 6, \ldots, 100\}$.

and:

$$\lim_{k\to\infty} \lim_{\lambda\to\infty} \frac{Z^G}{Z} = \lim_{k\to\infty} 1 - \left(1 - \frac{1}{k}\right)^k = 1 - \frac{1}{e}.$$

Figure 5 illustrates the value of the approximation factor as a function of λ and k, also considering the case where f is the identity function. In the latter case, the optimal solution value is equal to 1 and the ratio Z^G/Z is $1 - \left(1 - \frac{1}{k}\right)^k$ (cf. Figure 5). We observe that, with $f(z) = 1 - e^{-\frac{z}{\lambda}}$, the ratio Z^G/Z already drops below 0.65 for $\lambda = 10$ and $k > 30$.

7. Computational results

In this section, we assess the effectiveness of the MILP reformulations proposed in the paper from a computational perspective. We carry out three set of experiments, considering i) instances of Problem (2) featuring a strictly increasing function f (Section 7.1), ii) instances of Problem (2) featuring a not nondecreasing function f (Section 7.2), and iii) instances of Problem (1), studied in [Ahmed and Atamtürk, 2011] (Section 7.3).

Our experiments are performed on a single node of the IRIDIS 5 cluster, equipped with 40 dual 2.0 GHz Intel Skylake processors and 192 GB of DDR4 memory. Our code is written in C/C++ and compiled with `gcc 9.2.1`, using the `-O3` compiler optimization flag. Our branch-and-cut algorithms are implemented in `CPLEX 12.9.0`, using the `CALLABLE LIBRARY` framework, and run in single-threaded mode. [1] We set a time limit of 600 seconds for each run. [2]

In all of our experiments, we adopt the greedy algorithm of Section 6 (suitably extended to consider the specific constraints at hand—see further) to provide a first feasible solution to our branch-and-cut algorithms.

[1] To avoid numerical issues, we set both `CPX_PARAM_EPRHS` (corresponding the degree to which the basic variables of a model may violate their bounds) `CPX_PARAM_EPGAP` (corresponding to the gap between the best integer objective and the objective of the best node remaining) to 10^{-9}. Moreover, we set the parameters `Preprocessing_Linear` and `MIP_Strategy_CallbackReducedLP` to 0, as recommended by the `CPLEX` user manual for the `CPXsetusercutcallbackfunc` and `CPXsetlazyconstraintcallbackfunc` functions. These functions are used for separating the constraints featured in our reformulations which are violated by fractional and integer points, respectively.

[2] The source code can be downloaded from `https://github.com/fabiofurini/Maximizing-submodular-utility-functions`.

7.1 Experiments with Problem (2) with f increasing

We experiment with the four branch-and-cut (B&C) algorithms: `SC`, `OA+SC`, `OA+BC`, and `OA+BC+f`. `SC` is based on the standard reformulation Ref(SC) of Section 2, which relies on Constraints ($\text{SC1}^{\hat{h}_i}$) and ($\text{SC2}^{\hat{h}_i}$). `OA+SC` is based on the the reformulation Rev(OA+SC) of Section 3.3, which relies on Constraints (OA+SC1) and (OA+SC2). `OA+BC` is based on the reformulation Rev(OA+BC) of Section 3.4, which relies on Constraints (OA+BC1) and (OA+BC2). `OA+BC+f` is also based on the reformulation Rev(OA+BC) of Section 3.4, but,in it, Constraints (OA+BC1) and (OA+BC2) are also separated at fractional points (which can be can as efficiently as for integer points). These constraints are generated according to Lemma 1: the first one is obtained by setting $\gamma_j := 1$ for all $j \in N_1$ and $\gamma_j := 0$ for all $j \in N \setminus N_1$, and second one by setting $\gamma_j := 1$ for all $j \in N$. The separation is performed with a violation tolerance of 10^{-6}.

7.1.1 Instances

In our first set of experiments with an increasing function f, we adopt function (f) (see Section 1.1), as proposed by Ahmed and Atamtürk [2011], Yu and Ahmed [2016]. We consider five values of $\lambda \in \{0.25, 0.5, 0.75, 1, 2\}$.

We consider four different combinatorial structures associated with the constraint set Y of Problem (2): (i) a *k-cardinality constraint* $\sum_{\ell \in \hat{N}} y_\ell \leq k$—see, e.g., Nemhauser et al. [1978]; (ii) a *knapsack constraint*: $\sum_{\ell \in \hat{N}} \beta_\ell \, y_\ell \leq B$ with weights $\beta_\ell \geq 0$, $\ell \in \hat{N}$, and budget $B \geq 0$—see, e.g., Sviridenko [2004] and Ahmed and Atamtürk [2011], Yu and Ahmed [2016]; (iii) a set of *partition-matroid constraints* $\sum_{\ell \in \hat{N}_q} y_\ell \leq k_q$, with $1 \leq q \leq p$, according to which the metaitem set $\hat{N}$ is partitioned into p elements $\hat{N}_1, \ldots, \hat{N}_p$ and a cardinality constraint with budget k_p is imposed on each of them—see, e.g., Friedrich et al. [2019] and Sviridenko [2004]; (iv) a set of *conflict constraints* $y_\ell + y_{\ell'} \leq 1$, $\{\ell, \ell'\} \in \mathcal{E}$, used to model pairwise incompatibilities between metaitems; these constraints are based on a *conflict graph* $\mathcal{G} = (\hat{N}, \mathcal{E})$, where $\hat{N}$ is the set of metaitems and $\mathcal{E}$ is the set of edges representing the conflicts.

The greedy algorithm of Section 6 can be extended to find feasible solutions for all the combinatorial structures described above. For the variant of Problem (2) featuring a knapsack constraint, the greedy algorithm inserts the metaitem with the largest marginal contribution which does not violate the knapsack constraint. For the variant featuring the partition constraints, before adding each metaitem, the algorithm checks if the cardinality constraint of each element of the partition is not violated (Cornuejols et al. [1977] showed that this approach guarantees an approximation ratio of 1/2). For the case where conflict constraints are imposed, the greedy algorithm simply checks for their violation before inserting a metaitem in the solution (as the problem of maximizing a linear function subject to a set of conflicts is inapproximable within $|\hat{N}|^{1-\epsilon}$ for any $\epsilon > 0$ unless $\mathcal{ZPP} = \mathcal{NP}$ [Håstad, 1999], no constant-factor approximation factor can be obtained for this variant).

Motivated by the fact that Problem (2) can be used to model the stochastic MCP with uncertain customer demands and concave utility function, we generate our instances by extending the procedure proposed by ReVelle et al. [2008], Cordeau et al. [2019] for the deterministic MCP. To model the covering relationship between the sets of items N and metaitems $\hat{N}$, the sets N and $\hat{N}$ are embedded in $\mathbb{R}^2$ and the covering relationship is spatially induced by associating a covering radius $\hat{R}$ with each metaitem. The coordinates of each item and metaitem in $N \cup \hat{N}$ are chosen uniformly at random from $[0, 30]$. For each item $j \in N$, the set $\hat{N}(j)$ contains all the metaitems $\ell \in \hat{N}$ whose Euclidean distance from j is no larger than the covering radius $\hat{R}$. We consider $\hat{R} \in \{2, 4, 5, 6\}$, with smaller values of $\hat{R}$ which result in sparser instances. The items are sampled in such a way that each of them is covered by at least a metaitem. For each scenario $i \in M$ and item $j \in N$, we set a_{ij} to zero with probability $r \in \{25\%, 50\%\}$ and, with probability $1 - r$, we draw the value of u_{ij} uniformly at random from the set $\{1, 2, \ldots, 10\}$. Then, we introduce the coefficient $\Delta := \sum_{i\in M} \pi_i \sum_{j\in N} u_{ij}$, corresponding to the expected maximum return across all scenarios, and define the normalized value $a_{ij} := u_{ij}/\Delta$. For each choice of n, $\hat{n}$, and constraint structure, we generate 5 instances using a different seed for the random-number generator, 4 different values for the radius $\hat{R}$, and 2 different values for the scenario-item probability r.

We consider three main classes of instances. (i) The class k-`card`: 7200 instances featuring a k-cardinality constraint obtained by choosing $n \in \{5,000, 10,000, 20,000\}$, $\hat{n} \in \{40, 50, 60\}$, and $k \in \{10, 15\}$. (ii) The class `kp`: 3600 instances featuring a knapsack constraint obtained by choosing $n \in \{5,000, 10,000, 20,000\}$ and $\hat{n} \in \{40, 50, 60\}$; β_ℓ and B are defined by following the procedure described by Martello et al. [1999] for generating `strongly correlated` instances, which are among the most difficult instances for the knapsack problem [3]. (iii) The class `p-matroid`: 1200 instances featuring partition-matroid constraints obtained by choosing $n \in \{5,000, 10,000, 20,000\}$, $\hat{n} = 60$, $p = 3$, and $k_q = 5$, $q = 1, \ldots, 3$. For each of these three classes, we generate a second class identical to the former but also featuring conflict constraints induced by a conflict graph $\mathcal{G} = (\hat{N}, \mathcal{E})$ generated by following the Erdös-Rény model with an edge probability of 1%; we refer to these three classes as k-`card & confl`, `kp & confl`, and p-`matroid & confl`. Overall, the testbed contains 24,000 instances with diversified features and four different constraint structures.

7.1.2 Comparison of the four B&C algorithms

In Table 1, we compare the performance of the four B&C algorithms (`SC`, `OA+SC`, `OA+BC`, and `OA+BC+f`) for solving Problem (2) with f increasing. The table is vertically divided in five parts. The first part reports the instance sublcass and the total number of instances it contains (# total). For each of the four exact methods, we report the total number of instances solved to optimality within the time limit (denoted by # opt) and the average gap for the unsolved instances at the time limit. These gaps are calculated as $(UB - LB)/UB \cdot 100$ where UB and LB are, respectively, the best upper bound and the incumbent solution value achieved by the corresponding method by the time limit.

Let us first consider `SC` and `OA+SC`. When compared to `SC`, the number of instances solved to optimality by `OA+SC` is about 80% larger, and the average gap for the unsolved instances is reduced from 6.1% to 2.0%. For all subclasses of instances except for the `kp` class, the number of instances solved by `OA+SC` is three to four times larger than the number of instances solved by `SC`. Further improvements in the performance are obtained with `OA+BC`. This is in line with our theoretical results, where we show that Constraints (OA+BC2) strictly dominate Constraints (OA+SC2) (see Proposition 11), and that Constraints (OA+BC1) are equivalent to Constraints (OA+SC1) (see Proposition 10). Thanks to this, `OA+BC` solves about $1,700$ additional instances, and it reduces the average gap for the unsolved ones to 1.2%. By looking at the number of instances solved with `OA+BC+f`, we can see that a drastic improvement can be achieved by separating Constraints (OA+BC1)–(OA+BC2) also on fractional solutions found in the branching tree. In particular, `OA+BC+f` is able to solve 94% of the instances across all four classes. With `OA+BC+f`, the average gap for the remaining 1,553 unsolved instances is of only 0.3%. We observe that, when moving from `OA+BC` to `OA+BC+f`, the biggest improvements (in terms of the number of solved instances and the resulting gaps for the unsolved ones) are achieved on the p-`matroid` class. In the Appendix, we report a performance profile for each subclass of instances which clearly demonstrates the superiority of `OA+BC+f` for all three considered classes of instances. [4]

---

[3] The instances are obtained by setting $\beta_\ell = p_\ell + R/10$, where $p_\ell = \sum_{i\in N} \pi_i \sum_{j\in \hat{N}(\ell)} u_{ij}$, $R = 1000$, and $B := \lfloor 0.5 \sum_{\ell\in\hat{N}} \beta_\ell \rfloor$.

[4] We also experimented with solving the problem directly using the formulation reported in (2) with `BARON` 20.4 (Sahinidis [2017], Tawarmalani and Sahinidis [2005]), one of the state-of-the-art solvers for global optimization. Our extensive preliminary tests showed that `BARON` is computationally outperformed by all our

Table 1: Computational performance of the four B&C algorithms, within a time limit of 600 secs. The gaps are reported in percentage for the instances not solved to proven optimality.

		SC		OA+SC		OA+BC		OA+BC+f		greedy
Instance class	# total	# opt	gap	# opt	gap	# opt	gap	# opt	gap	gap
k-card	7,200	837	7.0	3,002	2.6	3,821	1.6	**7,058**	**0.2**	1.8
k-card & confl	7,200	890	7.1	3,097	2.8	3,923	1.7	**6,948**	**0.4**	1.9
kp	3,600	2,548	4.7	2,991	0.3	3,060	0.2	**3,160**	**0.1**	0.7
kp & confl	3,600	2,639	5.0	3,066	0.4	3,165	0.3	**3,246**	**0.2**	0.9
p-matroid	1,200	98	6.5	398	2.9	398	1.8	**1,056**	**0.3**	1.7
p-matroid & confl	1,200	102	6.6	400	3.0	398	1.9	**979**	**0.4**	2.1
Gran total	24,000	7,114	6.1	12,954	2.0	14,765	1.2	**22,447**	**0.3**	1.5

Our implementation of the greedy heuristic requires on average $\approx$ 1 sec of computing time. In Table 1, we report the gap of the greedy heuristic for the instances which cannot be solved to optimality by OA+BC+f. This gap is obtained by replacing LB by the value of the greedy solution (Z^G). By comparing the gaps obtained with OA+BC+f to those obtained with the greedy algorithm, we observe that the average gap of the instances that are not solved by OA+BC+f decreases from 1.5% to 0.3%. Overall, Table 1 shows that the quality of the incumbent solutions found by the B&C algorithm OA+BC+f is superior to the quality of the solutions found by the greedy algorithm not only on those which OA+BC+f manages to solve to optimality, but also one those where the algorithm incurs in the time limit.

In the following, we analyze the instance features which represent the main drivers of the computing time of OA+BC+f, by focusing on 7,200 instances from the subclass k-card. To graphically assess the impact of these features on the performance, we report four box plots in Figure 6. We report the computing times of OA+BC+f after aggregating the instances by the covering radius $\hat{R}$, the value of λ, the number of items n, and the number of metaitems $\hat{n}$.[5] Figure 6 shows that, by doubling the number of items, the average computing time doubles. However, by increasing the number of metaitems by 50%, the average computing time grows by a factor of 10. Increasing the length of the covering radius (and, thus, the density of G) also results in slightly increased computing times. However, very sparse instances (obtained by $\hat{R} = 2$, which are the only ones with curvature being strictly smaller than one) are surprisingly easy for our B&C methods. Finally, for larger values of λ, the instances tend to become easier to solve (as evidenced by the smaller computing times), which is likely to be a consequence of the fact that, the larger λ, the more the function f is close to being linear. Some further analysis has shown that by increasing the number of scenarios from 50 to 100, the average computing time also doubles, whereas changing the value of k from 10 to 15 does not have a significant impact on the computing time of OA+BC+f.

Table 2 illustrates the most relevant features concerning the execution of the best-performing B&C algorithm (OA+BC+f) on the 22,447 instances it manages to solve to optimality: LP gap (computed as $(Z^{\mathrm{LP}} - Z)/Z^{\mathrm{LP}} \cdot 100$, where Z^{LP} is the optimal solution value of the LP relaxation at the root node, and Z is the optimal solution value of the problem), number of generated inequalities, number branching nodes, and computing time. For what concerns the average and the maximum LP gaps, the table shows that very small LP gaps are obtained across all classes, suggesting that the tight LP gap is one of the main drivers of the excellent computation performance of OA+BC+f. In particular, the gaps are never larger

B&C algorithms. BARON fails to solve many instances with $n = 5,000$ items, and it is unable to solve any instance with $n = 10,000$.

[5] The computing times are reported in logarithmic scale through their quantiles. The lines extending vertically from the boxes indicate the variability outside the upper and lower quantiles. The outliers are plotted as individual points.

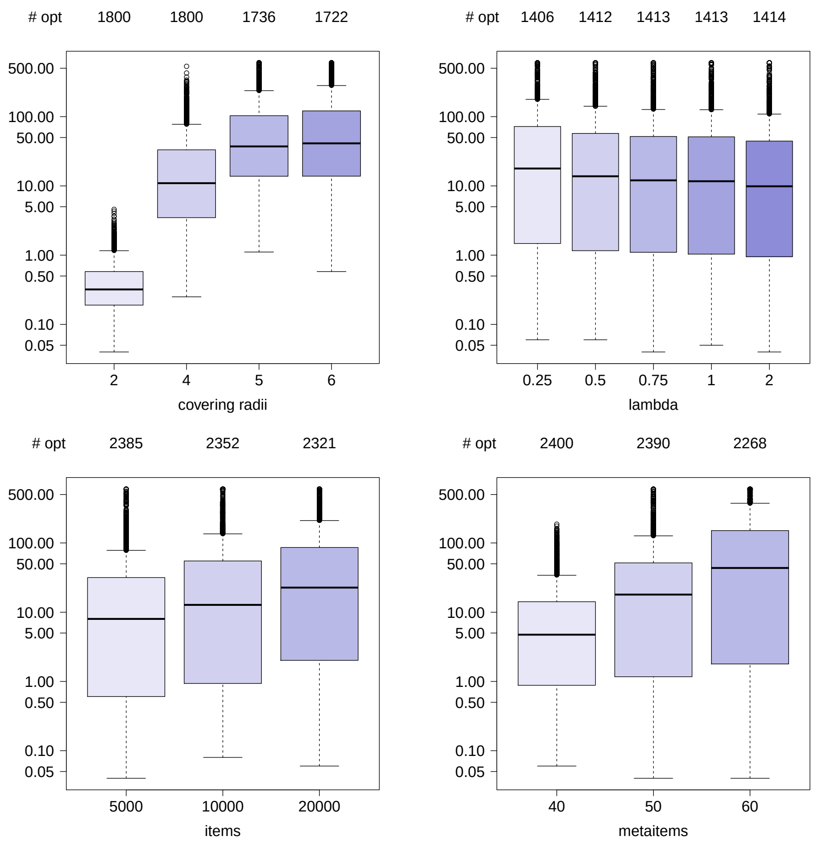

Fig. 6: Computing times (in logarithmic scale) of `OA+BC+f` for the `k-card` instances disaggregated by: (i) the covering radius $\hat{R}$, (ii) the risk-tolerance λ, (iii) the number of items n, and (iv) the number of metaitems $\hat{n}$. The numbers above each boxplot show the number of instances solved within the time limit of 600 secs.

Table 2: Detailed results obtained by using the `OA+BC+f` B&C algorithm on the instances solved to proven optimality within a time limit of 600 seconds.

	LP gap (%)			avg # of ineq.		# of nodes		time [s]
Instance class	avg	max	time [s]	frac. sol.	int. sol.	avg	max	avg
k-`card`	0.048	0.798	12.4	8,038.6	2,750.8	5.9	156	46.1
k-`card & confl`	0.066	1.244	12.4	8,505.6	2,459.1	6.6	119	52.7
`kp`	0.036	0.699	2.4	6,703.7	2,484.0	250.3	29,430	31.5
`kp & confl`	0.047	0.797	2.3	6,155.0	2,288.3	317.4	16,960	38.3
`p-matroid`	0.066	0.650	22.6	14,455.3	3,384.7	11.9	136	116.1
`p-matroid & confl`	0.081	1.191	21.1	13,529.2	2,846.7	11.1	78	129.3

than $\approx 1.25\%$ and their average values are smaller than $\approx 0.1\%$. The time taken to solve the LP relaxation is quite small and it varies as a function of the class of the instances. For what concerns the average number of Constraints (OA+BC1) and (OA+BC2) generated for separating fractional and integer points, the table show that, as one may expect, many more cuts are generated when fractional solutions are separated. The increased number of inequalities is well compensated by the tight LP gaps. The table indicates that the average number of nodes is relatively small in most of the instances, which constitutes a second indication of why the performance of `OA+BC+f` is good. The only exceptions are the instances from the `kp` class, for which several thousands of nodes are explored, albeit the time taken to solve each of them is extremely small—smaller, on average, than for the other instances. The average computing time is in line with the percentage of instances solved for the different classes, showing that the hardest instances are the ones featuring partition-matroid constraints.

We conclude this analysis by investigating the performance of `OA+BC+f` when applied to the worst-case instances we introduced in Section 6. As far as computing times are concerned, even instances with $k = 1,000$ (which correspond to $n = 1,001,000$ and $\hat{n} = 2,000$) can be solved within a fraction of a second at the root node of the branching tree by `OA+BC+f`. These worst-case instances show that the quality of the greedy solutions can be very far from the optimum, indicating that an efficient exact method can be extremely beneficial for applications (e.g., in finance) where obtaining high-quality (or optimal) solutions is crucial.

7.2 Experiments with a not nondecreasing utility function f for Problem (2)

In this section, we report the results of our experiments on instances where the utility function f is not nondecreasing, a case which, as discussed in Section 5, leads to a function $\hat{h}_i$, $i \in M$, which is not submodular. We consider the utility function (f-n-nd) introduced in Section 5 in which we set $\lambda \in \{0.25, 0.5, 0.75, 1, 2\}$ and $\alpha = 0.5$.

Three B&C algorithms are applied to this problem variant: `OA+SC+LBSC`, `OA+BC+LBSC` and `OA+BC+f+LBSC`. They are an extension of the three B&C algorithms described in Section 7.1.2 (i.e., `OA+SC`, `OA+BC`, and `OA+BC+f`). In each of them, Constraints (OA+SC1)–(OA+SC2) (in `OA+SC`), Constraints (OA+BC1)–(OA+BC2) (in `OA+BC`), and the fractional counterparts of the latter (in `OA+BC+f`) are separated at points where f is strictly increasing and, in addition, the lower-bounding constraints (OA+LBSC) are separated at integer points at which f is decreasing. In line with the previous experiments, each B&C algorithm is initialized with the greedy algorithm (see Section 6), halting it as soon as the marginal contribution becomes negative.

Table 3 summarizes our computational results obtained on a subset of 800 instances of the subclass k-`card` in which we selected $\hat{n} = 40$, $n \in \{5,000, 10,000\}$, and $k = 10$. The results are reported by aggregating the instances by the risk-tolerance parameter λ, with a total (# total) of 180 instances per value of λ. For each of the three B&C algorithms, we report the number of instances solved to optimality within the time limit of 600 seconds (# opt) and the average gaps for the remaining instances (gap). The results are in line with those obtained for the monotone case, and they indicate that outer approximation with Benders cuts (`OA+BC+LBSC`) clearly outperforms the outer approximation with submodular cuts (`OA+SC+LBSC`). This phenomenon is even more pronounced if Benders cuts are additionally separated at fractional points (setting `OA+BC+f+LBSC`). Indeed, `OA+BC+f+LBSC` solves the largest number of instances (286 out of 800) and, on average, provides the smallest gaps for the unsolved ones (2.9%). We notice that the problem becomes harder for small values of λ, where the decreasing term of f becomes dominant. Indeed, for $\lambda = 0.25$, none of the instances can be solved to optimality and the largest average gaps, ranging between 7.4% and 8.1%, are obtained. In spite of this, the tests show that our new decomposition principle based on a double-hypograph reformulation (and, in particular, algorithm `OA+BC+f+LBSC`) can be effectively applied to tackle Problem (2) also for the cases in which the utility function f is not nondecreasing, and hence the overall objective function of the problem is not submodular. We recall that the latter implies that the problem cannot be solved by classical submodular-maximization techniques based on, e.g., the submodular cuts of Nemhauser et al. [1978].

Table 3: Comparison of three B&C algorithms for the utility function (f-n-nd), $\alpha = 0.5$.

		`OA+SC+LBSC`		`OA+BC+LBSC`		`OA+BC+f+LBSC`	
	# total	# opt	gap	# opt	gap	# opt	gap
$\lambda = 0.25$	160	0	7.4	0	8.1	0	8.0
$\lambda = 0.5$	160	**7**	4.8	**7**	4.7	6	4.8
$\lambda = 0.75$	160	**40**	1.2	**40**	1.3	**40**	1.4
$\lambda = 1$	160	55	0.2	77	0.2	**80**	0.2
$\lambda = 2$	160	64	2.3	93	1.2	**160**	0.0
Total/avg	800	166	3.2	217	3.1	**286**	2.9

7.3 Experiments when solving the problem of Ahmed and Atamtürk [2011]

In this section, we present the results obtained with our methods when applied for the solution of Problem (1) with a single knapsack constraint, studied in Ahmed and Atamtürk [2011], Yu and Ahmed [2017], which Problem (2) generalizes. We adopt the reformulation Ref(OA) derived in Section 4.5, which is based on Constraints (17). In the experiments, Constraints (17b) are separated at integer points only (the computational benefits of separating fractional points is negligible for these instances). We refer to the resulting B&C method as `OA`.

The state-of-the-art exact method for Problem (1) is given by Yu and Ahmed [2017] whose results outperform the previous ones given by Ahmed and Atamtürk [2011]. In the remainder of this section, we refer to the approach of Yu and Ahmed [2017] as `BnC-LI` (where LI stands for lifted inequalities). It is important to mention that the separation problem for the new inequalities proposed in either work can be solved in polynomial time only for integer solutions and, that, for fractional solutions, a heuristic separation approach is adopted (differently from our case, where the separation problem is always solved to optimality). To test the performance of `OA` against `BnC-LI`, we generate the same set of instances tested in Yu and Ahmed [2017]. We refer the reader to [Ahmed and Atamtürk, 2011] for a detailed description on how the input parameters are generated. Along the lines of Yu and Ahmed [2017], we generate a set of instances with $n \in \{100, 150, 200\}$, $m \in \{50, 100\}$ and $\lambda \in \{0.8, 1, 2\}$. For each value of these parameters, we create, as in Yu and Ahmed [2017], 20 random instances, thus obtaining a testbed of 360 instances. [6]

Since the source code of `BnC-LI` is not available, the results of `BnC-LI` reported in Table 4 are directly taken from the table reported on page 161 in Yu and Ahmed [2017], for which the authors used a Python code based on `Gurobi` 5.6.3, running on a 2.3 GHz x86 Linux workstation with 7 GB memory. Moreover, the results in Yu and Ahmed [2017] are obtained by halting `Gurobi` at a 10^{-5} optimality gap, whereas, in our experiments, `CPLEX` is run with a smaller optimality gap of 10^{-9}.

In Table 4, we compare the performance of `OA`, `BnC-LI`, and `BARON`. The results are aggregated by the value of n, λ and m. For the two B&C methods we report the average number of cuts, the average number of branching nodes, and the average computing time. For `BARON`, we report the average computing time.

While all three methods `OA`, `BnC-LI`, and `BARON` are able to solve to proven optimality all the considered instances, they exhibit a substantial difference in terms of performance. The table clearly shows the main trends of the results, especially when comparing the order of magnitude of the reported numbers. As far as the computing time is concerned, the best approach is `OA`, which solves all the instances in an average time always smaller than 0.2 seconds, never exceeding 0.32 seconds. Notably, the average computing time of `BnC-LI` can

[6] The instances can be downloaded at https://github.com/fabiofurini/Maximizing-submodular-utility-functions/blob/main/exputil_data_set.tar.xz

Table 4: Computational performance of `OA`, `BnC-LI` of Yu and Ahmed [2017] and `BARON` when solving Problem (1) with a single knapsack constraint and employing function (f).

		BnC-LI			OA			BARON
	# total	cuts	nodes	time [s]	cuts	nodes	time [s]	time [s]
$n = 100$	120	706.3	214.2	27.0	1,308.5	285.1	**0.1**	2.7
$n = 150$	120	1,012.5	410.2	79.0	1,207.0	345.1	**0.2**	3.8
$n = 200$	120	1,689.3	882.0	135.7	1,322.4	319.3	**0.2**	4.3
$\lambda = 0.8$	120	1,702.3	784.2	115.8	1,368.5	305.9	**0.2**	3.7
$\lambda = 1$	120	1,188.5	516.8	81.8	1,350.3	303.3	**0.2**	3.6
$\lambda = 2$	120	517.3	205.3	44.0	1,119.1	340.4	**0.2**	3.5
$m = 50$	180	757.3	524.8	58.8	810.2	329.2	**0.1**	2.6
$m = 100$	180	1,514.8	479.4	102.3	1,748.4	303.8	**0.2**	4.6
Grand total	360	1,136.1	502.1	80.6	1,279.3	316.5	0.2	3.6

be as high as several hundreds of seconds, with a maximum of about 800 seconds. We also notice that `OA` scales much better than `BnC-LI` for an increasing number of items n, and that `BnC-LI` tends to struggle more for small values of λ. The table also indicates that the number of cutting planes and branching nodes for `OA` is typically smaller than that for `BnC-LI`. The difference is stronger for the harder instances with a larger number of items n and a smaller value of λ. The maximum number of cuts for `OA` never exceeds 1,600, whereas the same value can be as high as 12,864 for `BnC-LI`. Moreover, the maximum number of nodes for `OA` never exceeds 1,400, whereas the same value can be as high as 7,031 for `BnC-LI`. As to `BARON`, its computing times are up to three orders of magnitude larger than those for `OA`, and as high as 15 seconds.

Overall, our analysis shows that, in spite of the experiments being run on different architectures and with different solvers, the difference in the computing times of `OA` and `BnC-LI` is of various orders of magnitude (up to 3) in favor of `OA`, strongly suggesting that a similar performance (in which `OA` significantly beats `BnC-LI`) should be observed consistently across different computing environments.

8. Conclusions

In this article, we have studied a generalization of a discrete maximization problem with a concave utility function originally studied by Ahmed and Atamtürk [2011], obtained by combining the concave and differentiable function with a set-union operator. We have introduced a double-hypograph reformulation for the problem that allowed us to decompose it into its two main components, one related to the concave utility function, and the other one related to the set-union operator. The two components are then linearized: the concave function is replaced by its first order over-estimation, and the set-union operator is linearized either via submodular cuts or via a Benders decomposition, thus leading to two MILP reformulations featuring one or two exponential families of constraints. We have studied the relationship between the reformulations we proposed, providing theoretical results which compare the strength of the associated constraints. In our computational study, we have shown that a branch-and-cut approach based on Benders cuts combined with an outer approximation drastically outperforms the alternative MILP reformulations as well as a state-of-the-art solver for mixed-integer nonlinear optimization. This is particularly true when tackling problems in which the number of items is many orders of magnitude larger than the number of metaitems. Finally, we have also provided necessary modifications to tackle the problem setting in which function f is not nondecreasing, and assessed the performance of our methods for it. We remark that our approach can be straight-forwardly generalized to the case in which a different function f_i is considered in each scenario i, as well as to the case where f is a piecewise linear function, for which only a number of supporting functions are needed and, if the number of such functions is smaller than the range of values of its argument (p, in the paper), the overall approach is likely to be more efficient.

Many open questions still remain to be addressed for this rich optimization problem. One way to linearize the problem is to work in the (x, y)-space and proceed along the lines of Ahmed and Atamtürk [2011] to "project f out onto the x variables", which is a way of deriving outer-approximation cuts for an hypograph reformulation of the problem. Once the linearization is performed in the (x, y)-space, the x variables could be projected out in a Benders-like fashion. Finally, one could obtain an alternative MILP formulation in the y-space by generating Generalized Benders Cuts (Geoffrion [1972], Fischetti et al. [2016]) derived from the associated convex continuous subproblem. The connection between these different types of decomposition and linearizations and the strength of the respective MILP reformulations (both in the (x, y)- and in the y-space) remain an open topic for future research.

Another line of research addresses the investigation of problems where the function $\hat{q}_i$ is not induced by the set-union operator. In particular, our decomposition approach and the inequalities we derived based on the outer approximation combined with submodular cuts can be applied to any problem involving a function $h_i = f \circ \hat{q}_i$ where $\hat{q}_i$ is a monotone increasing submodular set function. A notable example is the one where $\hat{q}_i$ is the rank (or weighted rank) function of a matroid, such as, e.g., the value function of a graphic matroid whose ground set is determined by the set variable $\hat{S}$. In such a case, one could transform $\hat{q}_i$ into the value function $\hat{\mathcal{Q}}_i$ of a suitable optimization problem and construct Benders cuts analogue to those we derived by leveraging the full description of the convex hull of the problem underlying $\hat{\mathcal{Q}}_i$.

Acknowledgements The authors would like to thank three anonymous referees whose comments helped us greatly improve the quality of the paper.

References

R. Aboolian, O. Berman, and D. Krass. Competitive facility location model with concave demand. *European Journal of Operational Research*, 181(2):598–619, 2007.

S. Ahmed and A. Atamtürk. Maximizing a class of submodular utility functions. *Mathematical programming*, 128(1-2):149–169, 2011.

O. Berman and D. Krass. Flow intercepting spatial interaction model: a new approach to optimal location of competitive facilities. *Location science*, 6(1-4):41–65, 1998.

M. Conforti and G. Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete applied mathematics*, 7(3):251–274, 1984.

J.-F. Cordeau, F. Furini, and I. Ljubić. Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research*, 275(3):882–896, 2019.

G. Cornuejols, M. L. Fisher, and G. L. Nemhauser. Exceptional paper—location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.

E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307–339, 1986.

A. Ene and H. L. Nguyen. A nearly-linear time algorithm for submodular maximization with a knapsack constraint. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

U. Feige, V. S. Mirrokni, and J. Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

M. Fischetti, I. Ljubić, and M. Sinnl. Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557–569, 2016.

R. Fletcher and S. Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66:327–349, 1994.

T. Friedrich, A. Göbel, F. Neumann, F. Quinzan, and R. Rothenberger. Greedy maximization of functions with bounded curvature under partition matroid constraints. In *The Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 2272–2279. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33012272.

A. Geoffrion. Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.

E. Güney, M. Leitner, M. Ruthmair, and M. Sinnl. Large-scale influence maximization via maximal covering location. *European Journal of Operational Research*, 289(1):144–164, 2021.

J. Håstad. Clique is hard to approximate withinn 1- ε. *Acta Mathematica*, 182(1):105–142, 1999.

D. S. Hochbaum and A. Pathria. Analysis of the greedy approach in problems of maximum k-coverage. *Naval Research Logistics*, 45(6):615–627, 1998.

M. R. Karimi, M. Lucic, H. Hassani, and A. Krause. Stochastic submodular maximization: The case of coverage functions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6856–6866, Red Hook, NY, USA, 2017. Curran Associates Inc.

D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In L. Getoor, T. E. Senator, P. M. Domingos, and C. Faloutsos, editors, *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pages 137–146. ACM, 2003.

T. Klastorin. On a discrete nonlinear and nonseparable knapsack problem. *Operations Research Letters*, 9(4):233–237, 1990.

B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

I. Ljubić and E. Moreno. Outer approximation and submodular cuts for maximum capture facility location problems with random utilities. *European Journal of Operational Research*, 266(1):46 – 56, 2018.

S. Martello, D. Pisinger, and P. Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, 1999.

A. Mehrez and Z. Sinuany-Stern. Resource allocation to interrelated risky projects using a multiattribute utility function. *Management Science*, 29(4):430–439, 1983.

Â. G. L. Nemhauser, L. A. Wolsey, et al. Integer programming. Technical report, Université catholique de Louvain, Center for Operations Research and ..., 1988.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

C. ReVelle, M. Scholssberg, and J. Williams. Solving the maximal covering location problem with heuristic concentration. 35(2):427–435, 2008.

N. V. Sahinidis. *BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2017.

A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.
X. Shi, O. Prokopyev, and B. Zeng. Sequence independent lifting for the set of submodular maximization problem. In *25th international Conference on Integer programming and Combinatorial Optimization (IPCO)*, pages 1–12. Springer, 2020.
M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41 – 43, 2004.
M. Sviridenko, J. Vondrák, and J. Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Mathematics of Operations Research*, 42(4):1197–1218, 2017.
M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.
H. M. Weingartner. Capital budgeting of interrelated projects: survey and synthesis. *Management Science*, 12(7):485–516, 1966.
H. Wu and S. Küçükyavuz. A two-stage stochastic programming approach for influence maximization in social networks. *Computational Optimization and Applications*, 69(3): 563–595, 2018.
J. Yu and S. Ahmed. Maximizing expected utility over a knapsack constraint. *Operations Research Letters*, 44(2):180–185, 2016.
J. Yu and S. Ahmed. Maximizing a class of submodular utility functions with constraints. *Mathematical Programming*, 162(1-2):145–164, 2017.

Appendix

Further numerical examples

The following example illustrates the results in Proposition 6:

Example 1 Consider the instance used in the proof of Proposition 8, and let $\hat{S} = \{4\}$, for which we have $1.6 = \hat{q}_i(\hat{N} \setminus \{\ell\}) > \hat{q}_i(\hat{S}) = 1.1$. The (SC1$^{\hat{h}_i}$) and (OA+SC1) constraints read:

$$w \leq 0.105 + 0.053y_1 + 0.094y_2 + 0.044y_3 - 0.088(1 - y_4) + 0.133y_5$$
$$w \leq 0.105 + \underline{0.054}y_1 + \underline{0.099}y_2 + \underline{0.045}y_3 - \mathbf{0.098}(1 - y_4) + \underline{0.144}y_5.$$

Since the latter is tighter at the coefficient of $(1 - y_4)$, but weaker at those of y_1, y_2, y_3, and y_5, the two inequalities do not dominate each other.

Increasing the value of $\lambda = 100$ to make the function f closer to linear[7], for the same set we have the following (SC1$^{\hat{h}_i}$) and (OA+SC1) inequalities:

$$w \leq 0.01095 + 0.00592y_1 + 0.01083y_2 + 0.00494y_3 - 0.01076(1 - y_4) + 0.01570y_5$$
$$w \leq 0.01095 + \underline{0.00594}y_1 + \underline{0.01089}y_2 + \underline{0.00495}y_3 - \mathbf{0.01089}(1 - y_4) + \underline{0.01583}y_5.$$

While the latter is tighter than the former at the coefficient of $(1 - y_4)$ (in boldface) and weaker at those of y_1, y_2, y_3, and y_5 (underlined), the difference in the coefficients of y_1, y_2, y_3, and y_5 is less. ■

The following example illustrates the results in Proposition 7:

Example 2 Consider the instance of Example 1. Let $\hat{S} = \{5\}$, and observe that $\hat{q}_i(\hat{S}) > \hat{q}_i(\{\ell\})$ holds for each $\ell \in \hat{N} \setminus \hat{S}$. The (SC2$^{\hat{h}_i}$) inequality reads:

$$w \leq 0.148 + 0.059y_1 + 0.105y_2 + 0.049y_3 + 0.105y_4 - 0.148(1 - y_5).$$

[7] Note that we increased the numerical precision to 5 decimal places.

For the same set, the (OA+SC2) inequality reads:

$$w \le 0.148 + \mathbf{0.052} y_1 + \mathbf{0.094} y_2 + \mathbf{0.043} y_3 + \mathbf{0.094} y_4 - \underline{0.136}(1 - y_5),$$

which is tighter at the coefficients of y_1, y_2, y_3, and y_4 (highlighted in boldface), but weaker at the coefficient of $(1 - y_5)$ (underlined). The two inequalities do not dominate each other.

Consider now the instance of Example 1 with an extra metaitem identical to metaitem 5, i.e., with $N(6) = \{5, 6\}$. Let $\hat{S} = \{5, 6\}$, for which we have $\hat{q}_i(\hat{S}) \ge \hat{q}_i(\{\ell\})$ for all $\ell \in \hat{N} \setminus \hat{S}$ and $\hat{q}_i(\hat{S}) = \hat{q}_i(\hat{S} \setminus \{\ell\})$ for all $\ell \in \hat{S}$. The (SC2$^{\hat{h}_i}$) inequality reads:

$$w \le 0.148 + 0.059 y_1 + 0.105 y_2 + 0.049 y_3 + 0.105 y_4 - 0(1 - y_5) - 0(1 - y_6).$$

For the same set, the (OA+SC2) inequality reads:

$$w \le 0.148 + \mathbf{0.052} y_1 + \mathbf{0.094} y_2 + \mathbf{0.043} y_3 + \mathbf{0.094} y_4 - 0(1 - y_5) - 0(1 - y_6),$$

which is tighter at the coefficient of y_1, y_2, y_3, and y_4 (highlighted in boldface), and hence it dominates the (SC2$^{\hat{h}_i}$) one. ■

On the relation with the curvature of submodular functions

The set N_1, corresponding to the set of items in N which can be covered by a single metaitem $\ell \in \hat{N}$, is related to the notion of *curvature* of a submodular function introduced by Conforti and Cornuéjols [1984]. For a monotone increasing set function $f : 2^N \to \mathbb{R}^+$, we call *curvature w.r.t. a set* $S \subseteq N$ the quantity:

$$c^f(S) := 1 - \min_{\ell \in S, \varrho_\ell(\emptyset) > 0} \frac{\varrho_\ell(S \setminus \{\ell\})}{\varrho_\ell(\emptyset)},$$

and *total curvature* the quantity:

$$c^f := 1 - \min_{S \subseteq N} c^f(S) = 1 - \min_{\ell \in N : \varrho_\ell(\emptyset) > 0} \frac{\varrho_\ell(N \setminus \{\ell\})}{\varrho_\ell(\emptyset)},$$

(where the equality holds due to the monotonicity of f).

The notion of total curvature c plays a central role in establishing the approximation factor of the greedy algorithm used for solving the problem of maximizing a nondecreasing submodular function subject to a cardinality constraint. In particular, Conforti and Cornuéjols [1984] show that the greedy algorithm for this problem achieves a $(1 - \frac{1}{e^c})\frac{1}{c}$ approximation factor, which ranges from 1 when $c^f = 0$ (i.e., when f is linear and the greedy algorithm is exact) to $1 - \frac{1}{e}$ when $c^f = 1$ (the tight worst-case factor already established by Nemhauser et al. [1978]).

Let, for each $i \in M$, $c^{\hat{q}_i}$ be the total curvature of $\hat{q}_i$, and let $c^{\hat{h}_i}$ be the total curvature of $\hat{h}_i$. The following holds:

Observation 1 *For each $i \in M$, we have $c^{\hat{q}_i} < 1$ (respectively $c^{\hat{h}_i} < 1$) if and only if $N_1 \neq \emptyset$ and, for all $\ell \in \hat{N}$, there is some $j \in N(\ell)$ with $a_{ij} > 0$ which is only covered by ℓ (i.e., with $\hat{N}(j) = \{\ell\}$).*

Indeed, by definition of $c^{\hat{q}_i}$ (respectively, $c^{\hat{h}_i}$), we have $c^{\hat{q}_i} < 1$ (respectively, $c^{\hat{h}_i} < 1$) if and only if $\varrho_\ell^{\hat{q}_i}(N \setminus \{\ell\}) > 0$ (respectively, $\varrho_\ell^{\hat{q}_i}(N \setminus \{\ell\}) > 0$) holds for all $\ell \in N$, which is the case if and only if, for each $\ell \in \hat{N}$, there is some $j \in N(\ell)$ with $a_{ij} > 0$ which is only covered by ℓ (i.e., with $\hat{N}(j) = \{\ell\}$). As shown in the following, some of the results we derived for the comparison of the different inequalities we considered in this paper can be related to the total curvature $c^{\hat{h}_i}$ and $c^{\hat{q}_i}$ of $\hat{h}_i$ and $\hat{q}_i$, respectively.

Observation 2 *If* $c^{\hat{h}_i} < 1$*, then Constraints* (SC3$^{\hat{h}_i}$) *are strictly dominated by Constraints* (SC1$^{\hat{h}_i}$)*. Similarly, if* $c^{\hat{q}_i} < 1$*, then Constraints* (SC3$^{\hat{q}_i}$) *are strictly dominated by Constraints* (SC1$^{\hat{q}_i}$)*.*

Indeed, in Section 2, we notice that Constraints (SC3$^{\hat{h}_i}$) are dominated by (SC1$^{\hat{h}_i}$) if there exists an item $j \in N$ which is covered by a single metaitem $\ell \in \hat{N}$, i.e., such that $|\hat{N}(j)| = 1$, and $a_{ij} > 0$. We notice that if $c^{\hat{h}_i} < 1$, this is satisfied for all $\ell \in \hat{N}$.

When comparing the strength of Constraints (SC1$^{\hat{h}_i}$) and (OA+SC1), we show in Proposition 6 that a sufficient condition for Constraint (SC1$^{\hat{h}_i}$) to dominate (OA+SC1) is given by $\hat{q}_i(\hat{S}) = \hat{q}_i(\hat{N})$. The latter condition implies that the curvature be $c^{\hat{q}_i} = 1$.

Concerning the strength of Constraints (SC2$^{\hat{h}_i}$) and (OA+SC2), we show in Proposition 7 that the sufficient condition for Constraint (OA+SC2) to dominate (SC2$^{\hat{h}_i}$) is given by $\hat{q}_i(\hat{S}) > \hat{q}_i(\{\ell\})$ for all $\ell \in \hat{N} \setminus \hat{S}$ and $\hat{q}_i(\hat{S}) = \hat{q}_i(\hat{S} \setminus \{\ell\})$. We remark that the condition $\hat{q}_i(\hat{S}) = \hat{q}_i(\hat{S} \setminus \{\ell\})$, implies a curvature $c^{\hat{q}_i}(\hat{S}) = 1$.

Performance profiles comparing `SC`, `OA+SC`, `OA+BC`, and `OA+BC+f`

A graphical representation of the relative computational performance of the four branch-and-cut algorithms (`SC`, `OA+SC`, `OA+BC`, and `OA+BC+f`) is given by the performance profile (see Dolan and Moré [2002]) reported in Figures 7, 8 and 9. Each figure reports the results for the three macro categories of the `testbed-one`: in Figure 7 there are the sets `k-card` and `k-card & confl`, in Figure 8 there are the sets `kp` and `kp & confl` and, finally, in Figure 9 there are the sets `p-matroid` and `p-matroid & confl`. For each instance and algorithm, let τ be the normalized time defined as the ratio of the computing time required to solve the instance taken by the algorithm and the minimum computing time taken by all algorithms. For each algorithm and for each value of τ in the horizontal axis, the vertical axis reports the percentage of the instances for which the corresponding algorithm spent at most τ times the computing time of the fastest algorithm. Each curve starts from the percentage of instances in which the corresponding algorithm is the fastest. At the right end of the chart, we can read the percentage of instances solved by a specific algorithm. The best performance is graphically represented by the curves occupying the upper part of the figure. The values on the horizontal axis are reported in logarithmic scale. The figure clearly shows that `OA+BC+f` is the best algorithm for this sets of instances, as it is able to solve more instances and it is the fastest algorithm in most of the cases. The second best algorithm is `OA+BC`. These figure show that separating the Benders cuts for fractional solutions is crucial for the efficiency of the method. The two branch-and-cut algorithms based on submodular cuts, i.e., `OA+SC` and `SC`, have worse performance and they are both outperformed by `OA+BC+f` (and by `OA+BC`).

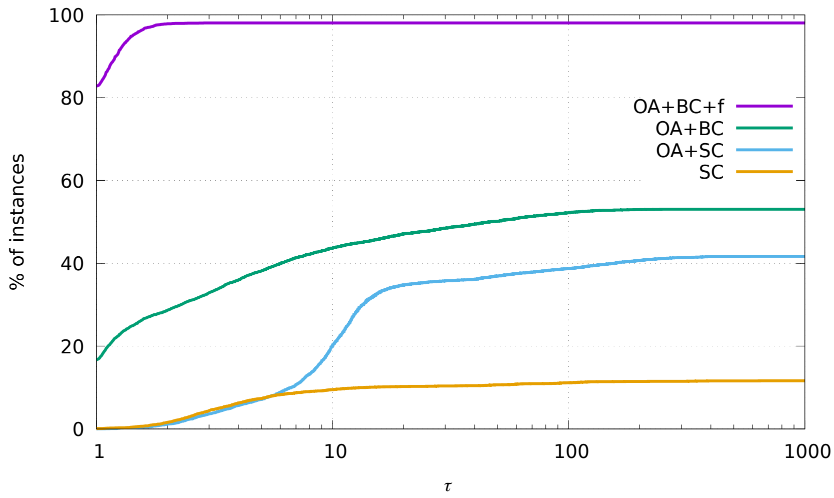

Fig. 7: Performance profile comparing the four branch-and-cut algorithms for the `k-card` and `k-card & confl` sets of instances.

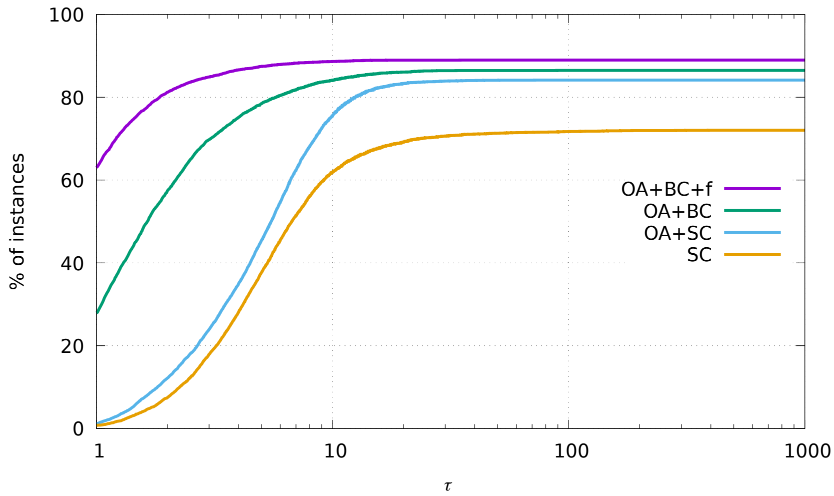

Fig. 8: Performance profile comparing the four branch-and-cut algorithms for the `kp` and `kp & confl` sets of instances.

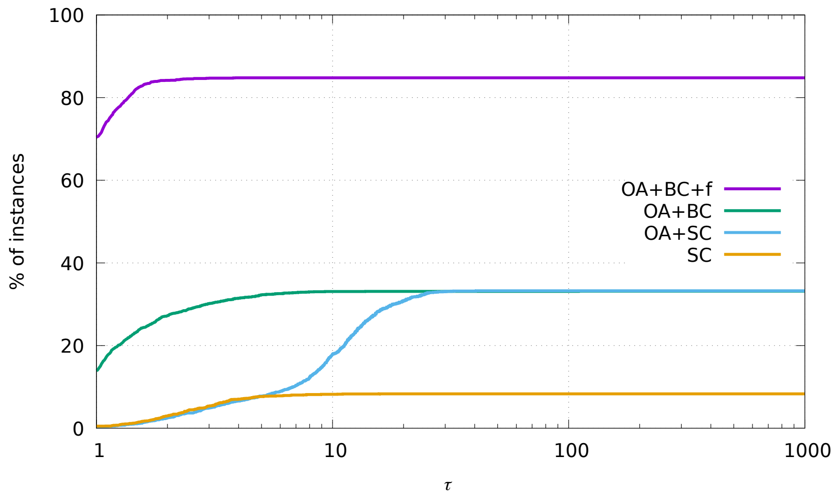

Fig. 9: Performance profile comparing the four branch-and-cut algorithms for the `p-matroid` and `p-matroid & confl` sets of instances.