Discrete Optimization 42 (2021) 100670

Contents lists available at ScienceDirect

Discrete Optimization

www.elsevier.com/locate/disopt

An exact cutting plane method for k-submodular function maximization

Qimeng Yu, Simge Küçükyavuz [*]

Department of Industrial Engineering and Management Sciences Northwestern University, Evanston, IL, USA

A R T I C L E   I N F O

Article history:
Received 29 March 2021
Received in revised form 25 June 2021
Accepted 12 August 2021
Available online 1 September 2021

Keywords:
k-submodular maximization
Cutting plane
Multi-type sensor placement

A B S T R A C T

A natural and important generalization of submodularity – k-submodularity – applies to set functions with k arguments and appears in a broad range of applications, such as infrastructure design, machine learning, and healthcare. In this paper, we study maximization problems with k-submodular objective functions. We propose valid linear inequalities, namely the k-submodular inequalities, for the hypograph of any k-submodular function. This class of inequalities serves as a novel generalization of the well-known submodular inequalities. We show that maximizing a k-submodular function is equivalent to solving a mixed-integer linear program with exponentially many k-submodular inequalities. Using this representation in a delayed constraint generation framework, we design the first exact algorithm, that is not a complete enumeration method, to solve general k-submodular maximization problems. Our computational experiments on the multi-type sensor placement problems demonstrate the efficiency of our algorithm in constrained nonlinear k-submodular maximization problems for which no alternative compact mixed-integer linear formulations are available. The computational experiments show that our algorithm significantly outperforms the only available exact solution method—exhaustive search. Problems that would require over 13 years to solve by exhaustive search can be solved within ten minutes using our method.

1. Introduction

Submodularity is an important concept in integer and combinatorial optimization. Several functions of great theoretical interest in combinatorial optimization are submodular, such as the set covering function and the graph cut function. Submodularity also arises in numerous practical applications, including the influence maximization problem [1], the sensor placement problem [2], and the hub location problem [3]. Submodularity is shown to be the discrete analogue of convexity [4], and the unconstrained submodular *minimization* problem is polynomially solvable [5–8]. However, submodular minimization with simple

[*] Corresponding author.
E-mail addresses: kim.yu@u.northwestern.edu (Q. Yu), simge@northwestern.edu (S. Küçükyavuz).

https://doi.org/10.1016/j.disopt.2021.100670
1572-5286/© 2021 Elsevier B.V. All rights reserved.

constraints, such as cardinality constraints, is generally NP-hard [9]. Submodular *maximization* is known to be NP-hard even in the unconstrained case. Nemhauser et al. [10] prove that the greedy method for maximizing a monotone submodular function subject to a cardinality constraint is a $(1-1/e)$-approximation algorithm.

In addition to the approximation algorithms, polyhedral approaches have been popular in submodular optimization research. Edmonds [11] gives linear inequalities which fully describe the convex hull of the epigraph for a submodular function (see also [12]). These inequalities are referred to as extended polymatroid inequalities, as they are closely related to a structure called extended polymatroid. Atamtürk and Narayanan [13] establish a polarity result analogous to the relationship between extended polymatroids and extended polymatroid inequalities for non-submodular functions. With this observation, Atamtürk and Narayanan [14] present an alternative proof for the convex hull description of the epigraphs for submodular functions. Nemhauser and Wolsey [15] give an exact method for submodular maximization from a polyhedral perspective. They show that maximizing any submodular function is equivalent to solving a mixed-integer linear program with exponentially many linear inequalities, referred to as the submodular inequalities. Atamtürk and Narayanan [14] provide valid inequalities for general set functions by exploiting their submodular–supermodular decomposition. Moreover, the polyhedral approach has received renewed interest, both in terms of strengthening extended polymatroid inequalities [16,17] and submodular inequalities [18–20], as well as extending their use to stochastic settings [21–25]. In addition, submodular properties in mixed-binary convex quadratic and conic optimization problems are discovered and exploited in [13,26–29].

Submodularity can be generalized to functions with k set arguments for any positive integer k, resulting in the concept called *k-submodularity*. This term is introduced by Huber and Kolmogorov [30], and it encompasses submodularity and *bisubmodularity* [31,32] as special cases when $k = 1$ and $k = 2$, respectively. Researchers have studied *k-submodular minimization*, where $k \geq 2$, using various approaches. Qi [32] proves an analogue of Lovász extension for bisubmodular functions, implying that this class of functions can be minimized in polynomial time using the ellipsoid method. Subsequently, weakly and strongly polynomial-time algorithms are proposed for unconstrained bisubmodular minimization [33,34]. Yu and Küçükyavuz [35] take a polyhedral approach and present a complete linear convex hull description for the epigraph of any bisubmodular function. Based on this polyhedral characterization, the authors propose an effective cutting plane algorithm to solve constrained bisubmodular minimization. Huber and Kolmogorov [30] generalize the Min-Max Theorem for submodular and bisubmodular minimization to the k-submodular case with $k \geq 3$. Whether k-submodular functions can be minimized in polynomial-time when $k \geq 3$ is still an open problem.

The *k-submodular maximization* problem – a generalization of the NP-hard submodular maximization problem – is also NP-hard. Extensive research has been devoted to developing approximation algorithms and proving their guarantees. For example, Singh et al. [36] give a constant-factor approximation algorithm for a class of bisubmodular functions. The authors refer to bisubmodularity that we consider in this paper as directed bisubmodularity. They show that a bisubmodular function can be embedded into a submodular function over an extended ground set, a set containing two copies of each element in the original ground set. For each subset of the extended ground set that contains both copies of an element, the submodular function value can be recursively obtained by solving discrete optimization problems with exponentially sized decision spaces. This construction is computationally expensive but theoretically interesting. Iwata et al. [37] and Ward and Živný [38] independently show that a randomized greedy algorithm attains the approximation guarantee of $1/2$ for unconstrained bisubmodular maximization (see also [39]). For k-submodular maximization with $k \geq 3$, Ward and Živný [39] achieve a $\max(1/3, 1/(1+a))$-approximation, where $a = \max(1, \sqrt{(k-1)/4})$. Iwata et al. [40] improve this result to a factor of $1/2$ and show that there is a $(k/(2k-1))$-approximation algorithm for unconstrained monotone k-submodular maximization. The authors further prove that their algorithms are asymptotically tight. In terms of *constrained* k-submodular maximization, researchers predominantly focus on non-negative monotone k-submodular functions. For example, Ohsaka and Yoshida [41] propose greedy algorithms for maximizing non-negative monotone k-submodular functions subject to a total cardinality constraint on all selected items, and to individual cardinality constraints on each of the k subsets. Their algorithm achieves a 1/2 approximation ratio under a total cardinality constraint, and this ratio is asymptotically tight. For the latter, the proposed algorithm achieves 1/3-approximation. Sakaue [42] studies maximization of non-negative monotone k-submodular functions under matroid constraints. The proposed greedy algorithm yields a 1/2-approximate solution.

To the best of our knowledge, there is no algorithm for maximizing possibly non-monotone k-submodular functions under general constraints. There has also been a paucity of research on *exact* solution methods for k-submodular maximization, both constrained and unconstrained. Therefore, we are interested in developing a versatile exact solution method. Practically, in situations where we have limited computing resources and an approximate solution suffices, approximation algorithms are extremely valuable. On the other hand, for problems such as high capital investments and strategic decision-making, where optimality is important and more computing resources are available, it is desirable to apply an exact method that solves the problem within a reasonable amount of time. Given the formidable computational burden of exhaustive search, there are no known efficient exact methods for constrained k-submodular maximization. To bridge this gap, our paper takes a polyhedral approach and proposes the first computationally attractive exact solution method that handles non-monotone k-submodular functions, as well as arbitrary linear constraints.

Before we summarize our results, we provide a few examples from a wide range of applications of k-submodular maximization.

1.1. Multi-type sensor placement

Sensor networks – enabled by internet of things (IoT) technology – provide real-time monitoring and control of systems to operate smart cities [43], smart homes [44], and smart grids [45]. These applications often call for multiple types of sensors in the network. For example, in smart water distribution networks, multiple types of sensors are placed to measure different aspects of water quality in real time [46]. Suppose we have k types of sensors and a set N of n locations to place them. If at most one sensor is allowed in each location, then every k-tuple of pairwise disjoint subsets of N corresponds to a multi-type sensor placement plan. The effectiveness of a sensor deployment plan can usually be evaluated using k-submodular functions such as the entropy function. Thus a multi-type sensor placement problem can be expressed in terms of constrained k-submodular maximization.

We provide more details of *coupled sensor placement*, in which we have two types of sensors for different measurements, such as temperature and humidity. Here, $k = 2$ is an arbitrary choice for illustration purposes. The description below can be generalized to the cases with $k \geq 3$. Every biset $(S_1, S_2) \in 3^N$ corresponds to a coupled sensor placement plan, in which the type-1 sensors are placed at the locations in S_1 and the type-2 sensors are placed at S_2. Due to a limited budget, we can place at most B_1 type-1 and B_2 type-2 sensors. We evaluate each sensor deployment plan using entropy, which measures how much uncertainty in the environment the sensors can capture [41]. The entropy of a discrete random variable X with support $\mathcal{X}$ is computed by

$$H(X) = -\sum_{x \in \mathcal{X}} \mathbb{P}(X = x) \log \mathbb{P}(X = x).$$

The entropy of X is high if multiple outcomes occur with similar probabilities, making it difficult for us to predict what we may observe. For instance, it is harder for us to guess the outcome of throwing a fair die correctly than that of a biased die, so the entropy in the case of a fair die is higher than the latter. In the context of coupled sensor placement, a discrete random variable X_{S_1,S_2} captures the possible observations reported by sensors installed at $(S_1, S_2) \in 3^N$, and the set $\mathcal{X}_{S_1,S_2}$ contains all possible observations. The entropy of X_{S_1,S_2} is

$$H(X_{S_1,S_2}) = -\sum_{x \in \mathcal{X}_{S_1,S_2}} \mathbb{P}(X_{S_1,S_2} = x) \log \mathbb{P}(X_{S_1,S_2} = x).$$

In an ideal coupled sensor placement plan, sensors are installed at locations where the corresponding observations are the most unpredictable. In other words, a placement (S_1^*, S_2^*) is the best when $H(X_{S_1^*,S_2^*})$ is maximal among all feasible $(S_1, S_2) \in 3^N$.

As mentioned earlier, the description above holds for $k \geq 2$ types of sensors. The function $f : (k+1)^N \to \mathbb{R}$, defined by $f(S_1, \ldots, S_k) = H(X_{S_1,\ldots,S_k})$ for all $(S_1, \ldots, S_k) \in (k+1)^N$, is monotone and k-submodular [41]. Thus the multi-type sensor placement problem is a cardinality-constrained k-submodular maximization problem with objective function f.

1.2. Multi-topic influence maximization

Social networks have allowed information to be spread faster than ever. In applications such as viral marketing, we may find a class of k-submodular maximization problems in which we aim to maximize the spread of information on k topics over a social network. Suppose one would like to promote k types of products. He or she may select influential network users to share their experiences of the products with their followers. As these followers share again with their own followers, messages about the products gradually diffuse across the network and reach a possibly large population. Kempe et al. [1] propose a diffusion model called independent cascade to describe the diffusion mechanism of a single type of influence. The authors also show that the expected total number of reached network users is a submodular function of the initial source of the spread. Ohsaka and Yoshida [41] generalize this model to allow $k \geq 2$ types of influence. In their model, a social network is represented by a digraph $G = (N, A)$, in which $N = \{1, 2, \ldots, n\}$ is the set of network users, and the arcs A capture how the users interact with each other. Every arc $(i, j) \in A$ is associated with probabilities $p^q_{(i,j)}$ for every $q \in \{1, 2, \ldots, k\}$. Each probability $p^q_{(i,j)}$ is the likelihood of j accepting i's information on the qth topic. Once j adopts a new piece of information, this user is ready to influence his or her own neighbors. Let $S_q \subseteq N$ be the group of influencers responsible for promoting the qth type of products and $A_q(S_q)$ be the individuals influenced by the initial spreaders S_q about product q under the stochastic model described above. Each influencer is restricted to accepting at most one type of sponsored product for fairness. Thus the initial influencers form a k-tuple of pairwise disjoint subsets of N. The function $f : (k+1)^N \to \mathbb{R}$ defined by $f(S_1, \ldots, S_k) = \mathbb{E}|\bigcup_{q=1}^k A_q(S_q)|$ computes the expected total number of influenced individuals given k sets of initial influencers. This function is shown to be monotone and k-submodular [41].

1.3. Multi-class feature selection

Feature selection plays a key role in multiple fields of research including machine learning [47], bioinformatics [48], and data mining [49]. This process improves the analysis of large datasets by reducing the dimensionality of data. In the resulting multi-class feature selection problems, there are k uncorrelated prediction variables, with their associated features mixed in a pool. The task is not only to find the most informative features, but also to classify the features with respect to the prediction variables, giving rise to a k-submodular optimization problem.

With two prediction variables, such problem is referred to as *coupled feature selection* in [36]. More formally, suppose that a Gaussian graphical model and a set of features N are given. Let C_1, C_2 be two variables to be predicted. The goal is to partition the features in N into two sets S_1 and S_2, such that S_1 is used to predict C_1, and S_2 is used to predict C_2. It is assumed that S_1, S_2 are mutually conditionally independent given $C = \{C_1, C_2\}$. The total number of selected features, $|S_1| + |S_2|$, is no more than a given number B. Next, we describe the score function that evaluates the mutual information obtained by a coupled feature selection. Suppose X and Y are discrete random variables, and $\mathcal{X}, \mathcal{Y}$ are the respective supports. Then the conditional entropy of X given Y is

$$H(X|Y) = -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \mathbb{P}(X = x, Y = y) \log \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(Y = y)}.$$

The biset mutual information is computed by

$$I(S_1, S_2; C) = H(S_1 \cup S_2) - \sum_{i\in S_1} H(i \mid C_1) - \sum_{j\in S_2} H(j \mid C_2),$$

where H is the entropy function discussed in Section 1.1 and the conditional entropy is defined above. Intuitively, the features with higher mutual information are more informative about both prediction tasks. Let $f(S_1, S_2) = I(S_1, S_2; C)$. The function f is monotone and bisubmodular [36], and the best features can be found by maximizing f.

1.4. *Drug-drug interaction detection*

Drug-drug interactions (DDIs) detection is an important application in the healthcare domain which exploits bisubmodularity. DDIs are the reactions resulting from using multiple drugs concomitantly. DDIs are a major cause of morbidity and mortality [50] – adverse drug events cause 770,000 injuries and deaths every year, and as much as 30% of these adverse drug events are due to DDIs [51,52]. Hu et al. [53] show that the correlations among the combinations of drugs and associated symptoms can be captured by a bisubmodular function, and the potential DDIs are determined by maximizing this function.

The aforementioned applications are solved using approximation algorithms due to the lack of exact solution methods.

1.5. *Our contributions*

Despite the developments in approximation algorithms for unconstrained and a few classes of constrained k-submodular maximization, there is no known exact method other than exhaustive search for general k-submodular maximization. To bridge this gap, we propose the first polyhedral approach to study k-submodular function maximization and provide an exact algorithm to maximize any k-submodular function subject to general constraints. We propose a new class of valid linear inequalities called k-submodular inequalities. These inequalities are non-trivial extensions of the submodular inequalities introduced by Nemhauser and Wolsey [15], in that the proposed k-submodular inequalities account for the interchanges of elements among the k subsets. With these valid inequalities, we develop an exact cutting-plane algorithm for constrained k-submodular maximization problems, which does not require the k-submodular objective function to satisfy any restrictive assumptions, such as monotonicity and non-negativity, nor does it restrict the structure of the constraints. We demonstrate the effectiveness of our algorithm by experimenting on the multi-type sensor placement problem, which has a highly nonlinear k-submodular objective function. The computational experiments show that our algorithm significantly outperforms the exhaustive search method.

1.6. *Outline*

The outline of this paper is as follows. In Section 2, we provide formal definitions of k-submodularity and review its known properties. In Section 3, we state and prove additional properties of k-submodular functions that have not been studied in the literature. These properties are used to establish our main results. Next, in Section 4, we propose a class of valid linear inequalities which we call the *k-submodular inequalities* for the hypograph of any k-submodular function. In particular, we show that maximizing a k-submodular function is equivalent to solving a mixed-integer linear program with exponentially many k-submodular inequalities. In Section 5, we give a cutting plane algorithm to solve the constrained maximization problems with k-submodular objective functions. We demonstrate the efficiency of our proposed algorithm and compare it against the exhaustive search method on the multi-type sensor placement problem in Section 6. Lastly, we conclude with a few remarks in Section 7.

2. Preliminaries

Let $N = \{1, 2, \ldots, n\}$ be a non-empty finite set. For any integer $k \geq 1$, we let

$$(k+1)^N = \{(S_1, S_2, \ldots, S_k) : S_q \subseteq N, S_q \cap S_{q'} = \emptyset, \text{ for all } q, q' \in \{1, 2, \ldots, k\} \text{ with } q \neq q'\}$$

be the collection of all k-sets, which are k-tuples of pairwise disjoint subsets of N. For brevity, we denote any $(S_1, S_2, \ldots, S_k) \in (k+1)^N$ by $\mathbf{S}$. We call $\mathbf{S} \in (k+1)^N$ a *partition*, or an *orthant*, of N, if $\bigcup_{q=1}^{k} S_q = N$.

Definition 2.1. For any integer $k \geq 1$, a function $f : (k+1)^N \to \mathbb{R}$ is *k-submodular* if for any $\mathbf{X} = (X_1, X_2, \ldots, X_k), \mathbf{Y} = (Y_1, Y_2, \ldots, Y_k) \in (k+1)^N$,

$$f(\mathbf{X}) + f(\mathbf{Y}) \geq \ f(\mathbf{X} \sqcap \mathbf{Y}) + f(\mathbf{X} \sqcup \mathbf{Y}),$$

where

$$\mathbf{X} \sqcap \mathbf{Y} = (X_1 \cap Y_1, X_2 \cap Y_2, \ldots, X_k \cap Y_k),$$

and

$$\mathbf{X} \sqcup \mathbf{Y} = \left((X_1 \cup Y_1) \backslash \bigcup_{q=2}^{k} (X_q \cup Y_q), \ldots, (X_k \cup Y_k) \backslash \bigcup_{q=1}^{k-1} (X_q \cup Y_q) \right).$$

In particular, the functions satisfying Definition 2.1 when $k = 1$ are called *submodular functions*, and when $k = 2$ such functions are referred to as *bisubmodular functions*. In the following discussion, we assume $k \geq 2$ unless specified otherwise. For any $q \in \{1, 2, \ldots, k\}$, $i \in N \backslash \bigcup_{q' \in \{1,\ldots,k\} \backslash \{q\}} X_{q'}$ and $\mathbf{X} \in (k+1)^N$, we define

$$\rho_{q,i}(\mathbf{X}) = f(X_1, \ldots, X_q \cup \{i\}, \ldots, X_k) - f(\mathbf{X}).$$

Intuitively, $\rho_{q,i}(\mathbf{X})$ represents the marginal contribution of adding $i \in N$ to the qth subset of $\mathbf{X}$. Ando et al. [54] provide an alternative definition of bisubmodularity that involves the notion of marginal contribution. Ward and Živný [39] generalize this result to k-submodularity. Before explaining this equivalent definition of k-submodular functions, we first establish a new term.

Definition 2.2. A function $f : (k+1)^N \to \mathbb{R}$ is *submodular over a partition* $\mathbf{S} = (S_1, S_2, \ldots, S_k)$ if the function

$$\hat{f}_{\mathbf{S}}(X) := f(X \cap S_1, X \cap S_2, \ldots, X \cap S_k) \tag{1}$$

is submodular over $X \subseteq N$.

Lemma 2.3 (*[39]*)**.** *For an integer $k \geq 2$, a function $f : (k+1)^N \to \mathbb{R}$ is k-submodular if and only if*

(C1) the function f is submodular over every partition of N, and
(C2) given any $\mathbf{X} \in (k+1)^N$ and any $i \in N \backslash \bigcup_{p=1}^{k} X_p$, $\rho_{q,i}(\mathbf{X}) + \rho_{q',i}(\mathbf{X}) \geq 0$ for every pair of $q, q' \in \{1, 2, \ldots, k\}$ such that $q \neq q'$.

Although Ward and Živný [39] assume the k-set functions to be non-negative, shifting such functions by a constant does not affect their k-submodularity. The next corollary immediately follows from condition (C1). It captures the diminishing marginal return property of k-submodular functions over every partition.

Corollary 2.4. *If f is a k-submodular function, then for any $\mathbf{X}, \mathbf{Y} \in (k+1)^N$ that satisfy $X_p \subseteq Y_p$ for all $p \in \{1, \dots, k\}$, $\rho_{q,i}(\mathbf{X}) \geq \rho_{q,i}(\mathbf{Y})$ for all $i \in N \setminus \bigcup_{p=1}^k Y_p$ and $q \in \{1, \dots, k\}$.*

Definition 2.5. A k-submodular function f over a ground set N is *monotone non-decreasing* if for any $\mathbf{X}, \mathbf{Y} \in (k+1)^N$ such that $X_q \subseteq Y_q$ for all $q \in \{1, \dots, k\}$, the property $f(\mathbf{Y}) \geq f(\mathbf{X})$ holds.

Equivalently, f is monotone non-decreasing if for any $\mathbf{X} \in (k+1)^N$ and $i \in N \setminus \bigcup_{p=1}^k X_p$, $\rho_{q,i}(\mathbf{X}) \geq 0$ for all $q \in \{1, \dots, k\}$. We call a monotone non-decreasing function simply a *monotone* function.

Without loss of generality, we assume that $f(\boldsymbol{\emptyset}) = 0$ where $\boldsymbol{\emptyset}$ is the k-set $(\emptyset, \dots, \emptyset)$. By slightly abusing notation, we let $f(\mathbf{X}) = f(\mathbf{x})$, where $\mathbf{x} = [x^1, \dots, x^k]^\top$ and $x_q^\top \in \{0,1\}^n = \mathbb{B}^n$ for every $q \in \{1, \dots, k\}$. To be more precise, $x_i^q = 1$ if $i \in X_q$, and $x_i^q = 0$ otherwise for $i \in N$ and $q \in \{1, \dots, k\}$. This is a unique one-to-one mapping between $(k+1)^N$ and $\{\mathbf{x} \in \mathbb{B}^{kn} : \sum_{q=1}^k x_i^q \leq 1 \text{ for all } i \in N\}$. The hypograph of f is

$$\mathcal{T}_f = \left\{ (\mathbf{x}, \eta) \in \mathbb{B}^{kn} \times \mathbb{R} : \eta \leq f(\mathbf{x}), \sum_{q=1}^k x_i^q \leq 1 \text{ for all } i \in N \right\}.$$

In this study, we consider maximization problems with k-submodular objective functions, namely

$$\max_{\mathbf{X} \in \mathcal{X}} f(\mathbf{X}), \tag{2}$$

where f is k-submodular and $\mathcal{X} \subseteq (k+1)^N$ denotes the collection of feasible k-sets. When the problem is unconstrained, $\mathcal{X}$ is $(k+1)^N$. Let $\mathcal{K}$ be the set of incidence vectors $\mathbf{x}$ that correspond to the feasible k-sets in $\mathcal{X}$. Problem (2) can be rewritten as

$$\max\{\eta : (\mathbf{x}, \eta) \in \mathcal{T}_f, \mathbf{x} \in \mathcal{K}\}. \tag{3}$$

In Section 4, we propose a set of valid linear inequalities for $\mathcal{T}_f$. By using these inequalities in a cutting plane framework, we propose the first computationally feasible exact method to solve problem (3) in Section 5. Before we do so, we first identify additional properties of k-submodular functions in the next section.

3. New properties of k-submodular functions

In this section, we establish a few properties of k-submodular functions that are not previously discussed in the literature to the best of our knowledge. These properties are useful for deriving valid linear inequalities for $\mathcal{T}_f$ in Section 4.

Lemma 3.1. *Given a ground set $N = \{1, 2, \dots, n\}$, a function $f : (k+1)^N \to \mathbb{R}$ is k-submodular and monotone if and only if*

$$\hat{f}_{\mathbf{S}}(Y) \leq \hat{f}_{\mathbf{S}}(X) + \sum_{i \in Y \setminus X} [\hat{f}_{\mathbf{S}}(X \cup \{i\}) - \hat{f}_{\mathbf{S}}(X)] \tag{4}$$

for any $X, Y \subseteq N$ over any partition $\mathbf{S}$ of N.

Proof. Nemhauser and Wolsey [15] show that a set function $g : 2^N \to \mathbb{R}$ is submodular and monotone if and only if

$$g(T) \leq g(S) + \sum_{j \in T \setminus S} [g(S \cup \{j\}) - g(S)] \text{ for any } S, T \subseteq N.$$

Suppose f is k-submodular and monotone. Given any partition $\mathbf{S}$, $\hat{f}_{\mathbf{S}}$ is submodular by (C1). For any $P \subseteq Q \subseteq N$, we construct $\mathbf{P}, \mathbf{Q}$ such that $P_q = P \cap S_q$ and $Q_q = Q \cap S_q$ for all $q \in \{1, \ldots, k\}$. Since $P_q \subseteq Q_q$ for all q and f is monotone, $\hat{f}_{\mathbf{S}}(P) = f(\mathbf{P}) \leq f(\mathbf{Q}) = \hat{f}_{\mathbf{S}}(Q)$, which implies that $\hat{f}_{\mathbf{S}}$ is monotone. Thus property (4) holds. Conversely, suppose (4) is true. Then $\hat{f}_{\mathbf{S}}$ is submodular and monotone over any partition $\mathbf{S}$ of N, and (C1) immediately follows. Let any $\mathbf{X} \in (k+1)^N$ and $i \in N \setminus \bigcup_{p=1}^k X_p$ be given. For any $q \in \{1, \ldots, k\}$, we construct a partition $\mathbf{S}^q$ such that $S^q_p = X_p$ for all $p \in \{1, \ldots, k\} \setminus \{q\}$, and $S^q_q = N \setminus \bigcup_{p \in \{1,\ldots,k\} \setminus \{q\}} X_p$. We note that $\mathbf{X}$ and $(X_1, \ldots, X_q \cup \{i\}, \ldots, X_k)$ are both in the partition $\mathbf{S}^q$. Now $\rho_{q,i}(\mathbf{X}) = \hat{f}_{\mathbf{S}^q}(X \cup \{i\}) - \hat{f}_{\mathbf{S}^q}(X) \geq 0$. It follows that $\rho_{q,i}(\mathbf{X}) + \rho_{q',i}(\mathbf{X}) \geq 0$ for any $q, q' \in \{1, \ldots, k\}$ with $q \neq q'$. Therefore, f is monotone and (C2) holds. We conclude that f is k-submodular and monotone. $\square$

Given a non-monotone submodular function g defined over a ground set N, Nemhauser and Wolsey [15] show that $g^*(S) := g(S) - \sum_{i \in S}(f(N) - f(N \setminus \{i\}))$ is monotone and submodular. Lemma 3.2 generalizes this result to non-monotone k-submodular functions.

Lemma 3.2. *Let $f : (k+1)^N \to \mathbb{R}$ be a k-submodular function. For every $i \in N$ and $q \in \{1, \ldots, k\}$, we define*

$$\xi_i^q = \min \left\{ \rho_{q,i}(\mathbf{S}) : \mathbf{S} \in (k+1)^{N \setminus \{i\}}, \bigcup_{p=1}^k S_p = N \setminus \{i\} \right\}.$$

The function

$$f^*(\mathbf{X}) := f(\mathbf{X}) - \sum_{q=1}^k \sum_{i \in X_q} \xi_i^q$$

is k-submodular and monotone.

Proof. By Lemma 3.1, it suffices to show that $\hat{f}^*_{\mathbf{T}}$ is submodular and monotone for any partition $\mathbf{T}$. Consider any $X, Y \subseteq N$ and any partition $\mathbf{T}$ of N. Let $\mathbf{X}$ and $\mathbf{Y}$ be the corresponding k-sets over this partition. In other words, $X_q = X \cap T_q$ and $Y_q = Y \cap T_q$ for all $q \in \{1, \ldots, k\}$. Then

$$\hat{f}^*_{\mathbf{T}}(X) + \sum_{i \in Y \setminus X} [\hat{f}^*_{\mathbf{T}}(X \cup \{i\}) - \hat{f}^*_{\mathbf{T}}(X)] \tag{5a}$$

$$= f^*(\mathbf{X}) + \sum_{q=1}^k \sum_{i \in Y_q \setminus X_q} [f^*(X_1, \ldots, X_q \cup \{i\}, \ldots, X_k) - f^*(\mathbf{X})] \tag{5b}$$

$$= f(\mathbf{X}) - \sum_{q=1}^k \sum_{i \in X_q} \xi_i^q + \sum_{q=1}^k \sum_{i \in Y_q \setminus X_q} [\rho_{q,i}(\mathbf{X}) - \xi_i^q] \tag{5c}$$

$$= f(\mathbf{X}) + \sum_{q=1}^k \sum_{i \in Y_q \setminus X_q} \rho_{q,i}(\mathbf{X}) - \sum_{q=1}^k \sum_{i \in X_q \cup Y_q} \xi_i^q \tag{5d}$$

$$\geq f(X_1 \cup Y_1, \ldots, X_k \cup Y_k) - \sum_{q=1}^k \sum_{i \in X_q \cup Y_q} \xi_i^q \tag{5e}$$

$$\geq f(\mathbf{Y}) + \sum_{q=1}^k \sum_{i \in X_q \setminus Y_q} \rho_{q,i}(T_1, \ldots, T_q \setminus \{i\}, \ldots, T_k) - \sum_{q=1}^k \sum_{i \in X_q \cup Y_q} \xi_i^q \tag{5f}$$

$$\geq f(\mathbf{Y}) + \sum_{q=1}^{k} \sum_{i \in X_q \setminus Y_q} \xi_i^q - \sum_{q=1}^{k} \sum_{i \in X_q \cup Y_q} \xi_i^q \tag{5g}$$

$$= f(\mathbf{Y}) - \sum_{q=1}^{k} \sum_{i \in Y_q} \xi_i^q \tag{5h}$$

$$= f^*(\mathbf{Y}) = \hat{f}^*_{\mathbf{T}}(Y). \tag{5i}$$

Eqs. (5b)–(5d) rewrite f^* in terms of f. Inequality (5e) is a consequence of Corollary 2.4 as we show next. For every $q \in \{1, \ldots, k\}$, we fix an ordering of the elements in $Y_q \setminus X_q$ to be $(\alpha^q(1), \alpha^q(2), \ldots, \alpha^q(|Y_q \setminus X_q|))$. Then

$$\begin{aligned} & f(X_1 \cup Y_1, \ldots, X_k \cup Y_k) \\ = & f(\mathbf{X}) + \sum_{q=1}^{k} \sum_{j=1}^{|Y_q \setminus X_q|} \rho_{q,\alpha^q(j)}(X_1 \cup Y_1, \ldots, X_q \cup \{\alpha^q(r)\}_{r=1}^{j-1}, X_{q+1}, \ldots, X_k) \\ \leq & f(\mathbf{X}) + \sum_{q=1}^{k} \sum_{j=1}^{|Y_q \setminus X_q|} \rho_{q,\alpha^q(j)}(\mathbf{X}) \\ = & f(\mathbf{X}) + \sum_{q=1}^{k} \sum_{i \in Y_q \setminus X_q} \rho_{q,i}(\mathbf{X}). \end{aligned}$$

Similarly, inequality (5f) holds because

$$\rho_{q,i}(X_1 \cup Y_1, \ldots, X_q \cup Y_q \setminus \{i\}, \ldots, X_k \cup Y_k) \geq \rho_{q,i}(T_1, \ldots, T_q \setminus \{i\}, \ldots, T_k)$$

for any $i \in X_q \cup Y_q \subseteq T_q$ where $q \in \{1, \ldots, k\}$. Inequality (5g) follows from the definitions of ξ_i^1 and ξ_i^2. Eqs. (5i) follow from the definitions of f^* and $\hat{f}^*$. $\square$

Lemma 3.3. *Let f be a monotone k-submodular function. Given any $\mathbf{X}, \mathbf{S} \in (k+1)^N$,*

$$f(\mathbf{X}) \leq f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \in X_q \setminus \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) + \sum_{q=1}^{k} \sum_{p \in \{1,\ldots,k\} \setminus \{q\}} \sum_{i \in X_q \cap S_p} \rho_{q,i}(\emptyset).$$

Proof. Let

$$X_q = \bigcup_{p=1}^{k} L_p^q \cup J_q, S_p = \bigcup_{q=1}^{k} L_p^q \cup K_p$$

where J_q, K_p, L_p^q are pairwise disjoint subsets of N for all $p, q \in \{1, \ldots, k\}$. Observe that $L_p^q = X_q \cap S_p$, $J_q = X_q \setminus \bigcup_{r=1}^{k} S_r$, and $K_p = S_p \setminus \bigcup_{r=1}^{k} X_r$, for all p and q. Furthermore,

$$f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \in X_q \setminus \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) + \sum_{q=1}^{k} \sum_{p \in \{1,\ldots,k\} \setminus \{q\}} \sum_{i \in X_q \cap S_p} \rho_{q,i}(\emptyset) \tag{6a}$$

$$= f\left(\bigcup_{q=1}^{k} L_1^q \cup K_1, \ldots, \bigcup_{q=1}^{k} L_k^q \cup K_k\right) + \sum_{q=1}^{k} \sum_{i \in J_q} \rho_{q,i}(\mathbf{S}) + \sum_{q=1}^{k} \sum_{p=1, p \neq q}^{k} \sum_{i \in L_p^q} \rho_{q,i}(\emptyset) \tag{6b}$$

$$\geq f\left(\bigcup_{q=1}^{k} L_1^q \cup K_1 \cup J_1, \ldots, \bigcup_{q=1}^{k} L_k^q \cup K_k \cup J_k\right) + f\left(\bigcup_{p=2}^{k} L_p^1, \ldots, \bigcup_{p=1}^{k-1} L_p^k\right) \tag{6c}$$

$$\geq f\left(L_1^1 \cup K_1 \cup J_1, \ldots, L_k^k \cup K_k \cup J_k\right) + f\left(\bigcup_{p=2}^{k} L_p^1, \ldots, \bigcup_{p=1}^{k-1} L_p^k\right) \tag{6d}$$

$$\geq f\left(\bigcup_{p=1}^{k} L_p^1 \cup K_1 \cup J_1, \ldots, \bigcup_{p=1}^{k} L_p^k \cup K_k \cup J_k\right) \tag{6e}$$

$$\geq f\left(\bigcup_{p=1}^{k} L_p^1 \cup J_1, \ldots, \bigcup_{p=1}^{k} L_p^k \cup J_k\right) \tag{6f}$$

$$= f(\mathbf{X}). \tag{6g}$$

Inequality (6c) follows from Corollary 2.4. Inequalities (6d) and (6f) are due to the monotonicity of f, and inequality (6e) holds because f is k-submodular. □

Lemma 3.3 applies to all monotone k-submodular functions. By using the relationship between any general k-submodular function f and its monotone counterpart f^* as stated in Lemma 3.2, we obtain the following result.

Corollary 3.4. *Let f be a k-submodular function. Given any $\mathbf{X}, \mathbf{S} \in (k+1)^N$,*

$$f(\mathbf{X}) \leq f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \in X_q \setminus \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) + \sum_{q=1}^{k} \sum_{p \in \{1,\ldots,k\} \setminus \{q\}} \sum_{i \in X_q \cap S_p} \rho_{q,i}(\emptyset) - \sum_{q=1}^{k} \sum_{i \in S_q \setminus X_q} \xi_i^q.$$

With these properties of k-submodular functions, we propose valid linear inequalities for the hypograph of any k-submodular function in the next section.

4. *k*-submodular inequalities

Let f be a k-submodular function defined over N. Recall that $\mathcal{T}_f$ is the hypograph of f. In this section, we propose two classes of valid linear inequalities for $\mathcal{T}_f$ depending on whether f is monotone. We refer to these inequalities as the *k-submodular inequalities.*

Proposition 4.1. *Let f be a monotone k-submodular function. For a given $\mathbf{S} \in (k+1)^N$, the inequality*

$$\eta \leq f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \notin \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) x_i^q + \sum_{q=1}^{k} \sum_{p \in \{1,\ldots,k\} \setminus \{q\}} \sum_{i \in S_p} \rho_{q,i}(\emptyset) x_i^q \tag{7}$$

is valid for $\mathcal{T}_f$.

Proof. Consider any $(\mathbf{x}, \eta) \in \mathcal{T}_f$. Recall that $\mathbf{x} = [x^1, \ldots, x^k]^\top \in \mathbb{B}^{kn}$. Let $\mathbf{X} \in (k+1)^N$ be the k-set represented by $\mathbf{x}$. For any $\mathbf{S} \in (k+1)^N$,

$$\eta \leq f(\mathbf{X}) \tag{8a}$$

$$\leq f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \in X_q \setminus \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) + \sum_{q=1}^{k} \sum_{p \in \{1,\ldots,k\} \setminus \{q\}} \sum_{i \in X_q \cap S_p} \rho_{q,i}(\emptyset) \tag{8b}$$

$$= f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \notin \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) x_i^q + \sum_{q=1}^{k} \sum_{p \in \{1,\ldots,k\} \setminus \{q\}} \sum_{i \in S_p} \rho_{q,i}(\emptyset) x_i^q \tag{8c}$$

Inequality (8a) follows from the definition of $\mathcal{T}_f$. Inequality (8b) holds by Lemma 3.3. Eq. (8c) uses the characteristic vector $\mathbf{x}$ to equivalently state the set relations. To see this, for every $q \in \{1, \dots, k\}$ and $i \in N$, $x_i^q = 1$ exactly when $i \in X_q$. $\square$

Proposition 4.2. *Let f be any k-submodular function. For a given $\mathbf{S} \in (k+1)^N$, the inequality*

$$\eta \le f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \notin \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) x_i^q + \sum_{q=1}^{k} \sum_{p \in \{1,\dots,k\} \setminus \{q\}} \sum_{i \in S_p} \rho_{q,i}(\mathbf{\emptyset}) x_i^q - \sum_{q=1}^{k} \sum_{i \in S_q} \xi_i^q (1 - x_i^q) \tag{9}$$

is valid for $\mathcal{T}_f$.

Proof. Consider any $(\mathbf{x}, \eta) \in \mathcal{T}_f$. Let $\mathbf{X} \in (k+1)^N$ be the k-set represented by $\mathbf{x}$. For any $\mathbf{S} \in (k+1)^N$,

$$\eta \le f(\mathbf{X}) \tag{10a}$$

$$\le f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \in X_q \setminus \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) + \sum_{q=1}^{k} \sum_{p \in \{1,\dots,k\} \setminus \{q\}} \sum_{i \in X_q \cap S_p} \rho_{q,i}(\mathbf{\emptyset}) - \sum_{q=1}^{k} \sum_{i \in S_q \setminus X_q} \xi_i^q \tag{10b}$$

$$= f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \notin \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) x_i^q + \sum_{q=1}^{k} \sum_{p \in \{1,\dots,k\} \setminus \{q\}} \sum_{i \in S_p} \rho_{q,i}(\mathbf{\emptyset}) x_i^q - \sum_{q=1}^{k} \sum_{i \in S_q} \xi_i^q (1 - x_i^q) \tag{10c}$$

Inequality (10a) holds due to the definition of $\mathcal{T}_f$, and (10b) follows from Corollary 3.4. Lastly, for every $q \in \{1, \dots, k\}$ and $i \in N$, $x_i^q = 1$ or equivalently $1 - x_i^q = 0$, exactly when $i \in X_q$. This justifies equation (10c). $\square$

We call inequalities (7) and (9) k-submodular inequalities associated with $\mathbf{S} \in (k+1)^N$. Intuitively, the first summation term on the right-hand side of a k-submodular inequality represents the marginal contribution made by appending additional elements to S_q, $q \in \{1, \dots, k\}$. The second nested summation term gives the upper bounds for the change in functional value when some elements in S_q are switched to $S_{q'}$ for any $q' \neq q$. When $k = 2$, we call the proposed inequalities *bisubmodular inequalities*. In the next remark, we show that our proposed inequalities subsume the submodular inequalities.

Remark 4.3. Notice that the submodular inequalities proposed by Nemhauser and Wolsey [55] is a special case of the k-submodular inequalities when $k = 1$. Let $g : 2^N \to \mathbb{R}$ be a submodular function defined on N. We denote the hypograph of g by

$$\{(y, \eta_g) \in \mathbb{B}^n \times \mathbb{R} \mid \eta_g \le g(y)\}.$$

For any $j \in N$ and $S \subseteq N$, we let $\gamma_j = g(N) - g(N \setminus \{j\})$ and $\rho_j(S) = g(S \cup \{j\}) - g(S)$. The submodular inequality associated with S is

$$\eta_g \le g(S) + \sum_{i \notin S} \rho_i(S) y_i - \sum_{j \in S} \gamma_j (1 - y_j),$$

which is exactly inequality (9) when $k = 1$. In this submodular inequality, the first summation accounts for marginal returns from appending items to S, and the second summation estimates the change in function g if items are removed from S. The natural extension of a submodular inequality to the k-submodular setting is an inequality that accounts for the change in function value by adding an unselected item to, or removing an item from each of the k subsets. However, the resulting inequalities are usually invalid because the function value also changes by switching a selected item from one subset to another. The k-submodular inequalities account for this complication.

Now let us consider the polyhedron

$$\mathcal{P}_f = \{(\mathbf{x},\eta) \in \mathbb{R}^{kn+1} : \eta \le f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \notin \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) x_i^q + \sum_{q=1}^{k} \sum_{p\in\{1,\dots,k\}\setminus\{q\}} \sum_{i\in S_p} \rho_{q,i}(\emptyset) x_i^q$$
$$- \sum_{q=1}^{k} \sum_{i\in S_q} \xi_i^q (1 - x_i^q), \forall \mathbf{S} \in (k+1)^N, \sum_{q=1}^{k} x_i^q \le 1, \forall i \in N\}.$$

Theorem 4.4. *Given any k-submodular (not necessarily monotone) function f and any $(\mathbf{x},\eta) \in \mathbb{B}^{kn} \times \mathbb{R}$, we have $(\mathbf{x},\eta) \in \mathcal{P}_f$ if and only if $\eta \le f(\mathbf{X})$, where $\mathbf{X}$ is the k-set represented by $\mathbf{x}$.*

Proof. Suppose $(\mathbf{x},\eta) \in \mathcal{P}_f$. Due to the second set of constraints in $\mathcal{P}_f$ and the fact that $\mathbf{x} \in \mathbb{B}^{kn}$, $x_i^q = 1$ exactly when $i \in X_q$ for any $i \in N$ and $q \in \{1,\dots,k\}$. In addition, $x_i^p = 0$ for all $p \in \{1,\dots,k\}\setminus\{q\}$. Therefore,

$$\eta \le f(\mathbf{X}) + \sum_{q=1}^{k} \sum_{i \notin \bigcup_{r=1}^{k} X_r} \rho_{q,i}(\mathbf{X}) \cdot 0 + \sum_{q=1}^{k} \sum_{p\in\{1,\dots,k\}\setminus\{q\}} \sum_{i\in X_p} \rho_{q,i}(\emptyset) \cdot 0 - \sum_{q=1}^{k} \sum_{i\in X_q} \xi_i^q (1-1)$$
$$= f(\mathbf{X}).$$

Conversely, suppose $\eta \le f(\mathbf{X})$. Let $\mathbf{x}$ be the characteristic vector of the k-set $\mathbf{X}$. The second set of constraints in $\mathcal{P}_f$ trivially holds at $(\mathbf{x},\eta)$. For any $\mathbf{S} \in (k+1)^N$,

$$\begin{aligned}
\eta &\le f(\mathbf{X}) \\
&\le f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \in X_q \setminus \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) + \sum_{q=1}^{k} \sum_{p\in\{1,\dots,k\}\setminus\{q\}} \sum_{i\in X_q \cap S_p} \rho_{q,i}(\emptyset) - \sum_{q=1}^{k} \sum_{i\in S_q\setminus X_q} \xi_i^q \\
&= f(\mathbf{S}) + \sum_{q=1}^{k} \sum_{i \notin \bigcup_{r=1}^{k} S_r} \rho_{q,i}(\mathbf{S}) x_i^q + \sum_{q=1}^{k} \sum_{p\in\{1,\dots,k\}\setminus\{q\}} \sum_{i\in S_p} \rho_{q,i}(\emptyset) x_i^q - \sum_{q=1}^{k} \sum_{i\in S_q} \xi_i^q (1 - x_i^q),
\end{aligned}$$

which follows from the argument in the proof of Proposition 4.2. Thus $(\mathbf{x},\eta)$ satisfies the first set of constraints in $\mathcal{P}_f$ as well. We conclude that $(\mathbf{x},\eta) \in \mathcal{P}_f$. □

Corollary 4.5. *Problem (3) is equivalent to*

$$\max\{\eta : (\mathbf{x},\eta) \in \mathcal{P}_f \cap \mathbb{B}^{kn} \times \mathbb{R}, x \in \mathcal{K}\}.$$

Proof. This result directly follows from Theorem 4.4. □

Remark 4.6. It may be difficult to compute ξ_i^q, where $i \in N$ and $q \in \{1,\dots,k\}$, for non-monotone k-submodular functions in practice. However, we do not require exact ξ_i^q values in the construction of the linear valid inequalities in our exact method. Proposition 4.2 still holds if we replace ξ_i^q by its lower bound. One lower bound is $\zeta = \underline{f} - \overline{f}$, where $\underline{f}$ and $\overline{f}$ are a lower and an upper bound of f, respectively. This estimate can be improved depending on the problem context. Similarly, we can replace the ξ_i^q values in $\mathcal{P}_f$ by their lower bounds that are cheaper to obtain. With the same proof, Theorem 4.4 and Corollary 4.5 hold for the modified $\mathcal{P}_f$.

5. A cutting plane algorithm for k-submodular maximization

We incorporate our proposed k-submodular inequalities in a cutting plane algorithm to tackle constrained k-submodular maximization problems in the form of (2), or equivalently (3). Following the results in Section 4, problem (3) can be rewritten as

$$\max \quad \eta \tag{11a}$$
$$\text{s.t.} \quad (\mathbf{x}, \eta) \in \mathcal{C}, \tag{11b}$$
$$\mathbf{x} \in \mathcal{K}. \tag{11c}$$

The polyhedral set $\mathcal{C}$ in constraint (11b) is defined by the k-submodular inequalities, which provide a piecewise linear representation of the objective function f. The set $\mathcal{K}$ in constraint (11c) contains the characteristic vectors $\mathbf{x}$ that are associated with the feasible k-sets in $\mathcal{X}$. By abusing notation, $\mathcal{K}$ here also embeds the binary restriction $\mathbf{x} \in \mathbb{B}^{kn}$ and the constraints $\sum_{q=1}^{k} x_i^q \leq 1$, for all $i \in N$.

We propose Algorithm 1 to solve problem (11). In this algorithm, we start with a relaxed set $\mathcal{C}$ and repeat the following subroutine until the optimality gap is within the given tolerance ϵ. We solve a relaxed version of (11) to obtain $\overline{\mathbf{x}}$ and $\overline{\eta}$ using a branch-and-bound algorithm. The current solution $\overline{\eta}$ is an upper bound for the optimal objective, and $f(\overline{\mathbf{x}})$ serves as a lower bound. Let $\overline{\mathbf{X}}$ be the k-set that corresponds to $\overline{\mathbf{x}}$. If $\overline{\eta}$ overestimates $f(\overline{\mathbf{x}})$, then we restrict $\mathcal{C}$ by adding the k-submodular inequality (9) associated with $\overline{\mathbf{X}}$. We repeat the same procedure in the next iteration.

Algorithm 1: Delayed Constraint Generation

```
1  Input initial $\mathcal{C}$, LB $= -\infty$, UB $= \infty$;
2  while $(UB - LB)/UB > \epsilon$ do
3      Solve problem (11) by a branch-and-bound algorithm to get $(\overline{\mathbf{x}}, \overline{\eta})$;
4      if $UB > \overline{\eta}$ then
5          UB $\leftarrow \overline{\eta}$;
6      end
7      compute $f(\overline{\mathbf{x}})$;
8      if $\overline{\eta} > f(\overline{\mathbf{x}})$ then
9          Add a $k$-submodular inequality (9) associated with $\overline{\mathbf{x}}$ to $\mathcal{C}$;
10     end
11     if $LB < f(\overline{\mathbf{x}})$ then
12         LB $\leftarrow f(\overline{\mathbf{x}})$;
13         Update the incumbent solution to $\overline{\mathbf{x}}$ ;
14     end
15 end
16 Output $\overline{\eta}$, $\overline{\mathbf{x}}$.
```

Corollary 5.1. *Algorithm 1 converges to an optimal solution of Problem (11) in finitely many iterations.*

Proof. This result follows from the fact that the number of feasible solutions is finite and from Theorem 4.4. □

6. Numerical study

In this numerical study, we demonstrate the effectiveness of our proposed Delayed Constraint Generation (DCG) Algorithm 1 by solving constrained k-submodular maximization problems with $k = 2$ and 3.

Specifically, we run computational experiments on the multi-type sensor placement problem, described in Section 1.1. We refer the readers to Example 5.1 in [35] for a small numerical example. To summarize, let a set N of n potential sensor deployment locations, and t pairs of measurements made by all types of sensors at each location be given. Our goal is to determine a multi-type sensor placement plan $\boldsymbol{S} \in (k+1)^N$, subject to cardinality constraints $|S_q| \leq B_q$ for $q \in \{1, 2, \dots, k\}$, such that the entropy is maximized. Since the entropy function is highly nonlinear, we cannot formulate the multi-type sensor placement problem as a compact mixed-integer linear program. Therefore, we compare our DCG algorithm against the exhaustive search (ES) method, which is the only available benchmark for an exact solution.

Using the DCG approach, we formulate the multi-type sensor placement problem as

$$\max \ \eta \tag{12a}$$

$$\text{s.t.} \ (\boldsymbol{x}, \eta) \in \mathcal{C}, \tag{12b}$$

$$\sum_{q=1}^{k} x_i^q \leq 1, \qquad \text{for all } i \in N, \tag{12c}$$

$$\sum_{i \in N} x_i^q \leq B_q, \qquad \text{for all } q \in \{1, 2, \dots, k\}, \tag{12d}$$

$$x_i^q \in \mathbb{B}, \qquad \text{for all } i \in N, q \in \{1, 2, \dots, k\}. \tag{12e}$$

The variables x^q and η are consistent with the notation in (11). Constraint (12b) gives the piecewise linear representation of the entropy function by exploiting its k-submodularity. The inequalities (12d) ensure that the cardinality requirements are satisfied.

We create random problem instances using the Intel Berkeley research lab dataset [56]. This dataset includes the sensor readings of three environmental factors – light, temperature, and humidity – at 54 locations in the Intel Berkeley Research lab from February 28th to April 5th in 2004. We discretize the temperature data into three equal-width bins. Both light and humidity data are discretized into two equal-width bins. For the set of experiments with $k = 2$, we aim to find the best placement plan for light and temperature sensors. When $k = 3$, our goal is to determine the optimal placement plan for light, temperature and humidity sensors. The experiments are executed on two threads of a Linux server with Intel Haswell E5-2680 processor at 2.5 GHz and 128 GB of RAM. Our algorithms are implemented in Python 3.6 and Gurobi Optimizer 7.5.1 with default settings and one-hour time limit for each instance.

First, we explore how the changes in the number of deployable locations, n, affect the computational performance of the DCG algorithm in both sets of experiments with $k = 2$ and 3. We randomly select $n \in \{20, 30, 40, 50\}$ out of the 54 locations in the dataset. At each of the n locations, we randomly select $t \in \{50, 100, 150, 200\}$ tuples of light, temperature and humidity measurements for evaluating the entropy. We set $B_q = \lfloor n/10 \rfloor$ for $q \in \{1, \dots, k\}$, so that the cardinality bound for each type of sensors increases proportionally with n. The computational results are summarized in Table 1 for $k = 2$ and Table 2 for $k = 3$. The first two columns in these tables list the numbers of deployable sensor locations and the numbers of observations at each location. Columns 3–5 present the relevant computational statistics, namely the running time in seconds, the number of k-submodular inequalities added, and the number of branch-and-bound nodes visited when solving the relaxed master problems. The end optimality gap is computed by (UB−LB)/UB, where UB and LB are the best upper and lower bounds on the objective respectively. The last column reports the runtime of ES. At the time limit, ES does not provide end gap information because it produces no lower bounds and has to essentially go through each feasible solution to prove optimality.

In this set of experiments, DCG solves all the instances when $k = 2$ and solves all but one test case when $k = 3$, within the one hour time limit. The test case that DCG fails to solve attains a small end gap of 6.1%. Based on the runtime differences, the instances with $k = 3$ are in general more challenging than those with $k = 2$, when the other parameters are kept the same. Overall, the computational statistics for $k = 2$ and

Table 1
Computational performance of DCG and exhaustive search in the coupled sensor placement problem. The statistics are averaged across 3 trials. The superscript $^{\ell}$ means that out of the three trials, ℓ instances reach the time limit of one hour.

n	t	Time (s)	# cuts	# nodes	ES time (s)
20	50	0.73	49.67	53.00	9.01
	100	1.09	38.67	42.33	17.68
	150	1.76	51.67	54.00	21.89
	200	0.97	16.33	20.00	34.07
30	50	7.67	285.33	289.67	$-^3$
	100	13.04	250.33	253.67	$-^3$
	150	15.07	224.67	230.00	$-^3$
	200	23.70	241.67	245.00	$-^3$
40	50	36.05	810.33	814.67	$-^3$
	100	161.80	1783.33	1791.00	$-^3$
	150	145.67	1255.33	1261.00	$-^3$
	200	283.81	1676.67	1685.00	$-^3$
50	50	104.21	1549.00	1560.00	$-^3$
	100	478.09	3559.33	3566.33	$-^3$
	150	1372.18	6911.67	6920.67	$-^3$
	200	2474.35	9268.67	9275.67	$-^3$

Table 2
Computational performance of DCG and exhaustive search in the sensor placement problem with three types of sensors. The statistics are averaged across 3 trials. The superscript $^{\ell}$ means that out of the three trials, ℓ instances reach the time limit of one hour.

n	t	Time (s)	# cuts	# nodes	End gap	ES time (s)
20	50	0.90	33.33	35.33	–	1633.00
	100	1.58	32.67	35.33	–	2953.18
	150	1.29	18.00	21.33	–	$-^3$
	200	3.68	36.67	40.00	–	$-^3$
30	50	7.54	142.67	147.33	–	$-^3$
	100	20.27	222.33	226.67	–	$-^3$
	150	22.48	156.67	160.33	–	$-^3$
	200	46.55	243.33	248.33	–	$-^3$
40	50	82.10	947.33	952.67	–	$-^3$
	100	164.97	1043.00	1049.33	–	$-^3$
	150	395.54	1599.67	1606.33	–	$-^3$
	200	640.38	1902.67	1912.00	–	$-^3$
50	50	253.29	1878.33	1886.00	–	$-^3$
	100	1779.49	6907.67	6920.00	–	$-^3$
	150	2719.26	6640.00	6646.33	–	$-^3$
	200	$-^3$	6496.00	6500.67	6.10%	$-^3$

$k = 3$ display the same trend. The runtime, the number of branch-and-bound nodes as well as the number of k-submodular inequalities added increase as n increases. Variations in t for small n values do not significantly impact the computational statistics. When $n = 50$, all the statistics increase at a greater rate in response to increments in t compared with the case of $n = 20$. On the other hand, unsurprisingly, ES struggles for $n \geq 30$ when $k = 2$ and 3. In the test cases with $n = 20$ that ES solves, the computing time drastically increases as k goes from 2 to 3, reflecting the exponential growth of the decision space. For $n = 20, 30$ and 40, all instances are solved by DCG under 11 min; while ES hits the time limit for $n \geq 30$. In fact, when $k = 2$, $n = 50$ and $t = 100$, exhaustive search needs to enumerate $50!/(5!5!40!) \approx 2.59 \times 10^{12}$ feasible bisets to find an exact optimal solution. We find that objective function evaluation alone takes 1.6×10^{-4} s on average when $t = 100$ for each biset. Thus, the total function evaluation time is equivalent to 13.13 years. In contrast, our algorithm finds an optimal solution in 8 min.

Next, we explore the effects of the cardinality bounds B_q, $q \in \{1, \dots, k\}$, on the computational performance of DCG. We consider all the placeable sensor locations; that is, $n = 54$. Again, at each location, we randomly select $t \in \{50, 100, 150, 200\}$ k-tuples of sensor readings. We set $B_q = B$ for $q \in \{1, \dots, k\}$, where B is an integer between 1 and 5. The computational results are summarized in Table 3 for $k = 2$ and Table 4 for $k = 3$. In either table, the first column shows the upper bounds on the number of each type

Table 3
Computational performance of DCG and exhaustive search in the coupled sensor placement problem. The statistics are averaged across 3 trials. The superscript $^{\ell}$ means that out of the three trials, ℓ instances reach the time limit of one hour.

B	t	Time (s)	# cuts	# nodes	End gap	ES time (s)
1	50	1.09	38.00	41.00	–	0.48
	100	1.16	18.33	21.33	–	1.04
	150	1.45	16.67	20.00	–	1.33
	200	1.97	14.67	17.33	–	1.98
2	50	11.03	283.33	286.67	–	546.64
	100	27.05	333.33	337.67	–	1158.15
	150	22.14	209.00	212.33	–	1596.22
	200	47.81	283.33	286.00	–	2307.06
3	50	31.90	669.33	674.33	–	$-^{3}$
	100	92.53	939.00	943.33	–	$-^{3}$
	150	149.39	997.00	1000.00	–	$-^{3}$
	200	335.15	1535.67	1541.33	–	$-^{3}$
4	50	67.92	1106.67	1114.33	–	$-^{3}$
	100	404.21	3422.33	3428.33	–	$-^{3}$
	150	754.14	3934.00	3940.00	–	$-^{3}$
	200	1308.30	5154.00	5159.33	–	$-^{3}$
5	50	149.37	1773.67	1782.33	–	$-^{3}$
	100	636.18	4467.33	4473.00	–	$-^{3}$
	150	2068.81	8792.00	8800.33	–	$-^{3}$
	200	2941.30^{1}	11 385.00	11 394.33	2.78%	$-^{3}$

Table 4
Computational performance of DCG and exhaustive search in the sensor placement problem with three types of sensors. The statistics are averaged across 3 trials. The superscript $^{\ell}$ means that out of the three trials, ℓ instances reach the time limit of one hour.

B	t	Time (s)	# cuts	# nodes	End gap	ES time (s)
1	50	1.14	19.33	22.00	–	37.56
	100	3.05	31.00	33.67	–	64.11
	150	4.80	34.00	36.67	–	94.67
	200	5.33	26.00	28.00	–	134.80
2	50	22.63	267.33	271.00	–	$-^{3}$
	100	41.45	294.33	297.00	–	$-^{3}$
	150	50.65	246.00	248.67	–	$-^{3}$
	200	88.52	308.67	311.67	–	$-^{3}$
3	50	103.74	923.00	927.67	–	$-^{3}$
	100	184.81	1405.00	1410.67	–	$-^{3}$
	150	355.86	1315.33	1319.67	–	$-^{3}$
	200	524.26	1420.67	1424.00	–	$-^{3}$
4	50	215.59	1624.00	1630.67	–	$-^{3}$
	100	787.15	3315.00	3320.67	–	$-^{3}$
	150	1618.03	4851.67	4857.00	–	$-^{3}$
	200	2467.97	5447.00	5453.00	–	$-^{3}$
5	50	386.29	2330.33	2339.67	–	$-^{3}$
	100	1449.97	5362.33	5372.67	–	$-^{3}$
	150	2871.68^{1}	8067.00	8073.00	4.34%	$-^{3}$
	200	$-^{3}$	7037.33	7044.00	6.26%	$-^{3}$

of sensors. The second column lists the numbers of observations at each of the 54 locations for entropy evaluations. The next four columns are the relevant computational statistics, including the runtime in seconds, the number of k-submodular inequalities added, the number of branch-and-bound nodes visited and the end optimality gaps. The last column reports the running time of ES in seconds.

In both Tables 3 and 4, higher cardinality bounds B make the multi-type sensor placement problem more challenging, with longer running time, more cuts added, and more branch-and-bound nodes visited. In particular, when $B \geq 4$, the computational statistics increase at a higher rate as t increases than that with $B \leq 3$. These trends are true for both $k = 2$ and $k = 3$, since the decision space consisting of all the plausible deployment plans grows rapidly as more sensors are allowed. If, in addition, the number of observations is high, then each entropy evaluation becomes expensive, resulting in a significant increase in the running time. When $k = 2$, even in the most challenging instances where $B = 5$ and $t = 200$, DCG solves two test instances within one hour, and obtains a feasible solution within 3% optimality in the third trial. Similarly, when $k = 3$, DCG attains small optimality gap under 6.26% for the most challenging test case with $B = 5$ and $t = 200$. When the cardinality bounds are below three, DCG solves all the instances within six minutes for $k = 2$ and nine minutes for $k = 3$. On the contrary, ES fails due to the time limit for all the instances with $B \geq 3$ and $k = 2$. ES struggles more when $k = 3$ and fails as soon as B exceeds 1.

7. Concluding remarks

In this paper, we propose a polyhedral approach to solve constrained maximization problems with k-submodular objective functions. We propose valid linear inequalities, referred to as k-submodular inequalities, for the hypograph of any k-submodular function. This development leads us to construct the first exact method – a delayed constraint generation algorithm based on k-submodular inequalities – to solve general k-submodular maximization problems other than the trivially available exhaustive search method. Our numerical experiments on a highly nonlinear multi-type sensor placement problem show that the proposed delayed constraint generation algorithm is effective when handling challenging k-submodular maximization problems that cannot be solved exactly by existing methods.

Acknowledgments

We thank the editor and the reviewers for providing comments that improved this paper. This research is supported, in part, by NSF, United States grant 2007814. This research is also supported in part through the computational resources and staff contributions provided for the Quest high performance computing facility at Northwestern University, United States, which is jointly supported by the Office of the Provost, the Office for Research, and Northwestern University Information Technology.

References

[1] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, Theory Comput. 11 (4) (2015) 105–147.
[2] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, C. Faloutsos, Efficient sensor placement optimization for securing large water distribution networks, J. Water Resour. Plann. Manage. 134 (6) (2008) 516–526.
[3] I. Contreras, E. Fernández, Hub location as the minimization of a supermodular set function, Oper. Res. 62 (3) (2014) 557–570.
[4] L. Lovász, Submodular functions and convexity, in: Mathematical Programming the State of the Art, Springer, 1983, pp. 235–257.
[5] S. Iwata, L. Fleischer, S. Fujishige, A combinatorial strongly polynomial algorithm for minimizing submodular functions, J. ACM 48 (4) (2001) 761–777.
[6] A. Schrijver, A combinatorial algorithm minimizing submodular functions in strongly polynomial time, J. Combin. Theory Ser. B 80 (2) (2000) 346–355.
[7] J.B. Orlin, A faster strongly polynomial time algorithm for submodular function minimization, Math. Program. 118 (2) (2009) 237–251.
[8] Y.T. Lee, A. Sidford, S.C.-W. Wong, A faster cutting plane method and its implications for combinatorial and convex optimization, in: 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2015, pp. 1049–1065.
[9] Z. Svitkina, L. Fleischer, Submodular approximation: Sampling-based algorithms and lower bounds, SIAM J. Comput. 40 (6) (2011) 1715–1737.
[10] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions—I, Math. Program. 14 (1) (1978) 265–294.
[11] J. Edmonds, Submodular functions, matroids, and certain polyhedra, in: R. Guy, H. Hanani, N. Sauer, J. Schönheim (Eds.), Combinatorial Structures and their Applications, Gordon and Breach, New York, 1970, pp. 69–87.
[12] J. Edmonds, Submodular functions, matroids, and certain polyhedra, in: Combinatorial Optimization—Eureka, You Shrink!, Springer, 2003, pp. 11–26.
[13] A. Atamtürk, V. Narayanan, Polymatroids and mean-risk minimization in discrete optimization, Oper. Res. Lett. 36 (5) (2008) 618–622.

[14] A. Atamtürk, V. Narayanan, Submodular function minimization and polarity, Math. Program. (2021) 1–11.
[15] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, Wiley-Interscience, New York, 1988, p. 662.
[16] J. Yu, S. Ahmed, Polyhedral results for a class of cardinality constrained submodular minimization problems, Discrete Optim. 24 (2017) 87–102.
[17] Q. Yu, S. Küçükyavuz, Strong valid inequalities for a class of concave submodular minimization problems under cardinality constraints, 2021, arXiv preprint arXiv:2103.04398.
[18] S. Ahmed, A. Atamtürk, Maximizing a class of submodular utility functions, Math. Program. 128 (1) (2011) 149–169.
[19] J. Yu, S. Ahmed, Maximizing a class of submodular utility functions with constraints, Math. Program. 162 (1–2) (2017) 145–164.
[20] X. Shi, O.A. Prokopyev, B. Zeng, Sequence independent lifting for the set of submodular maximization problem, in: International Conference on Integer Programming and Combinatorial Optimization, Springer, 2020, pp. 378–390.
[21] H.-H. Wu, S. Küçükyavuz, Probabilistic partial set covering with an oracle for chance constraints, SIAM J. Optim. 29 (1) (2019) 690–718.
[22] H.-H. Wu, S. Küçükyavuz, A two-stage stochastic programming approach for influence maximization in social networks, Comput. Optim. Appl. 69 (3) (2018) 563–595.
[23] H.-H. Wu, S. Küçükyavuz, An exact method for constrained maximization of the conditional value-at-risk of a class of stochastic submodular functions, Oper. Res. Lett. 48 (3) (2020) 356–361.
[24] Y. Zhang, R. Jiang, S. Shen, Ambiguous chance-constrained binary programs under mean-covariance information, SIAM J. Optim. 28 (4) (2018) 2922–2944.
[25] W. Xie, On distributionally robust chance constrained programs with wasserstein distance, Math. Program. 186 (2021) 115–155.
[26] A. Gómez, Submodularity and valid inequalities in nonlinear optimization with indicator variables, 2018, http://www.optimization-online.org/DB_FILE/2018/11/6925.pdf.
[27] A. Atamtürk, A. Gómez, Submodularity in conic quadratic mixed 0–1 optimization, Oper. Res. 68 (2) (2020) 609–630.
[28] A. Atamtürk, A. Gómez, Supermodularity and valid inequalities for quadratic optimization with indicators, 2020, arXiv preprint arXiv:2012.14633.
[29] F. Kılınç-Karzan, S. Küçükyavuz, D. Lee, Conic mixed-binary sets: Convex hull characterizations and applications, 2020, arXiv preprint arXiv:2012.14698.
[30] A. Huber, V. Kolmogorov, Towards minimizing k-submodular functions, in: International Symposium on Combinatorial Optimization, Springer, 2012, pp. 451–462.
[31] R. Chandrasekaran, S.N. Kabadi, Pseudomatroids, Discrete Math. 71 (3) (1988) 205–217.
[32] L. Qi, Directed submodularity, ditroids and directed submodular flows, Math. Program. 42 (1–3) (1988) 579–599.
[33] S. Fujishige, S. Iwata, Bisubmodular function minimization, SIAM J. Discrete Math. 19 (4) (2005) 1065–1073.
[34] S.T. McCormick, S. Fujishige, Strongly polynomial and fully combinatorial algorithms for bisubmodular function minimization, Math. Program. 122 (1) (2010) 87–120.
[35] Q. Yu, S. Küçükyavuz, A polyhedral approach to bisubmodular function minimization, Oper. Res. Lett. 49 (1) (2021) 5–10.
[36] A. Singh, A. Guillory, J. Bilmes, On bisubmodular maximization, in: Artificial Intelligence and Statistics, 2012, pp. 1055–1063.
[37] S. Iwata, S.-i. Tanigawa, Y. Yoshida, Bisubmodular Function Maximization and Extensions, Technical report, Technical Report METR 2013-16, The University of Tokyo, 2013.
[38] J. Ward, S. Živný, Maximizing bisubmodular and k-submodular functions, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 1468–1481.
[39] J. Ward, S. Živný, Maximizing k-submodular functions and beyond, ACM Trans. Algorithms (TALG) 12 (4) (2016) 1–26.
[40] S. Iwata, S.-i. Tanigawa, Y. Yoshida, Improved approximation algorithms for k-submodular function maximization, in: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2016, pp. 404–413.
[41] N. Ohsaka, Y. Yoshida, Monotone k-submodular function maximization with size constraints, in: C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems 28, Curran Associates, Inc., 2015, pp. 694–702.
[42] S. Sakaue, On maximizing a monotone k-submodular function subject to a matroid constraint, Discrete Optim. 23 (2017) 105–113.
[43] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, IEEE Internet Things J. 1 (1) (2014) 22–32.
[44] H. Ghayvat, J. Liu, S.C. Mukhopadhyay, X. Gui, Wellness sensor networks: A proposal and implementation for smart home for assisted living, IEEE Sens. J. 15 (12) (2015) 7341–7348.
[45] M. Abujubbeh, F. Al-Turjman, M. Fahrioglu, Software-defined wireless sensor networks in smart grids: An overview, Sustainable Cities Soc. 51 (2019) 101754.
[46] Public Utilities Board Singapore, Managing the water distribution network with a smart water grid, Smart Water 1 (2016) 1–13.
[47] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.
[48] Y. Saeys, I.n. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[49] L. Rokach, O.Z. Maimon, Data Mining with Decision Trees: Theory and Applications, Vol. 69, World scientific, 2008.

[50] Y. Lu, D. Shen, M. Pietsch, C. Nagar, Z. Fadli, H. Huang, Y.-C. Tu, F. Cheng, A novel algorithm for analyzing drug-drug interactions from medline literature, Sci. Rep. 5 (2015) 17357.

[51] N.P. Tatonetti, G.H. Fernald, R.B. Altman, A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports, J. Am. Med. Inform. Assoc. 19 (1) (2012) 79–85.

[52] M. Pirmohamed, M. Orme, Drug interactions of clinical importance, in: Davies's Textbook of Adverse Drug Reactions, Chapman & Hall Medical London, 1998, pp. 888–912.

[53] Y. Hu, R. Wang, F. Chen, Bi-submodular optimization (BSMO) for detecting drug-drug interactions (DDIs) from on-line health forums, J. Healthc. Inform. Res. 3 (1) (2019) 19–42.

[54] K. Ando, S. Fujishige, T. Naitoh, A characterization of bisubmodular functions, Discrete Math. 148 (1–3) (1996) 299–303.

[55] G.L. Nemhauser, L.A. Wolsey, Maximizing submodular set functions: formulations and analysis of algorithms, in: North-Holland Mathematics Studies, Vol. 59, Elsevier, 1981, pp. 279–301.

[56] P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, R. Thibaux, Intel lab data, 2004, Online Dataset. http://db.csail.mit.edu/labdata/labdata.html.