

Fractionally Subadditive Maximization under an Incremental Knapsack Constraint

Yann Disser¹, Max Klimm², and David Weckbecker¹

¹ TU Darmstadt, {disser|weckbecker}@mathematik.tu-darmstadt.de

² TU Berlin, klimm@math.tu-berlin.de

Abstract. We consider the problem of maximizing a fractionally subadditive function under a knapsack constraint that grows over time. An incremental solution to this problem is given by an order in which to include the elements of the ground set, and the competitive ratio of an incremental solution is defined by the worst ratio over all capacities relative to an optimum solution of the corresponding capacity. We present an algorithm that finds an incremental solution of competitive ratio at most $\max\{3.293\sqrt{M}, 2M\}$, under the assumption that the values of singleton sets are in the range $[1, M]$, and we give a lower bound of $\max\{2.449, M\}$ on the attainable competitive ratio. In addition, we establish that our framework captures potential-based flows between two vertices, and we give a tight bound of 2 for the incremental maximization of classical flows with unit capacities.

1 Introduction

We consider an incremental knapsack problem of the form

$$\max_{S \subseteq E} f(S) \quad \text{s.t.} \quad w(S) \leq C, \quad (1)$$

with a monotone objective function $f: 2^E \rightarrow \mathbb{R}_{\geq 0}$ over a finite ground set E of size $m := |E|$, weights $w: E \rightarrow \mathbb{R}_{\geq 0}$ associated with the elements of the ground set, and a growing capacity bound $C > 0$. We denote an optimum solution to (1) for a fixed capacity C by S_C^* and let $f^*(C)$ be its value as a function of C .

The growing capacity bound models the increase of available resources over time, e.g., the spending budget of consumers with a steady income, and corporations with a regular cashflow. In these scenarios, it is natural to fix an order in which elements of the ground set should be purchased. As an example, think of a consumer with a steady income of 1 per time unit. There are three items $E = \{c, s, t\}$, a compact camera c at a price of $w(c) = 1$, a system camera s at a price of $w(s) = 2$, and a telephoto lens t at a price of $w(t) = 2$. For a subset $S \subseteq E$, suppose that $f(S) = 1$, if S contains the compact camera, but not the system camera, $f(S) = 2$ if it contains the system camera, but not the telephoto lens, $f(S) = 3$ if it contains both the system camera and the telephoto lens, and $f(S) = 0$, otherwise. If the buyer chooses to buy the compact camera first, they receive a utility of 1 which is optimal for the budget of 1, i.e., $f^*(1) = 1$. However, if the buyer delays their purchase until they can afford the system camera, the value at time 1 is 0 but the value at time 2 is 2 which is optimal, i.e., $f^*(2) = 2$. As this simple example illustrates, there is a tradeoff between the attained values at different times and, in particular, there is no purchase order that yields the optimal value at all times.

In this paper, we address this tradeoff from a perspective of competitive analysis. An incremental solution to our incremental knapsack problem is given by an ordering $\pi = (e_{\pi(1)}, e_{\pi(2)}, \dots, e_{\pi(m)})$ of the ground set. For capacity $C > 0$, let $\pi(C)$ be the items contained in the maximal prefix of π that fits into the capacity C , i.e., $\pi(C) = \{e_{\pi(1)}, e_{\pi(2)}, \dots, e_{\pi(k)}\}$ for some $k \in \mathbb{N}$ such that $\sum_{i=1}^k w(e_{\pi(i)}) \leq C$ and either $k = m$ or $\sum_{i=1}^{k+1} w(e_{\pi(i)}) > C$. We say that the ordering π is ρ -competitive with $\rho \geq 1$ if

$$f^*(C) \leq \rho \cdot f(\pi(C)) \quad \text{for all } C > 0.$$

We call an ordering *competitive* if it is ρ -competitive for some constant $\rho \geq 1$. It is easy to see that without any further assumptions, there is no hope to obtain a competitive ordering π . For illustration, consider the example above where the value of the system camera is changed to $M \in \mathbb{R}_{\geq 0}$. Since any competitive ordering is forced to buy the compact camera first in order to be competitive for $C = 1$, no ordering can be better than M -competitive at time 2. To avoid this issue, we will assume in the following that the value of each singleton set falls in a bounded interval $[1, M]$ for some constant $M \geq 1$, i.e., $f(e) \in [1, M]$ for all $e \in E$. We call such valuations *M-bounded*. Observe that even for 1-bounded valuations, complementarities between the items may prevent a competitive ordering. For illustration, adapt our example such that every non-empty subset of items yields a value of 1 unless it contains both the system camera and the telephoto lens for which the value is $N > M$. Again, any competitive ordering has to put the compact camera first and then cannot be better than N -competitive for budget $C = 4$. To avoid this issue, we additionally require that f is fractionally subadditive. Fractional subadditivity is a generalization of submodularity and a standard assumption in the combinatorial auction literature (cf. Nisan [24], Lehmann et al. [17]). Formally, a function $f: 2^E \rightarrow \mathbb{R}_{\geq 0}$ is called *fractionally subadditive* if $f(A) \leq \sum_{i=1}^k \alpha_i f(B_i)$ for all $A, B_1, B_2, \dots, B_k \in 2^E$ and all $\alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{R}_{\geq 0}$ such that $\sum_{i \in \{1, \dots, k\}: e \in B_i} \alpha_i \geq 1$ for all $e \in A$. Observe that fractional subadditivity implies regular subadditivity, i.e., $f(A \cup B) \leq f(A) + f(B)$, but not vice-versa.

To summarize, we consider incremental solutions to (1), assuming that

- (MO) f is monotone, i.e., $f(A) \geq f(B)$ for $A \supseteq B$,
- (MB) f is M -bounded, i.e., $f(e) \in [1, M]$ for all $e \in E$,
- (FS) f is fractionally subadditive.

Before stating our results, we illustrate the applicability of our framework to different settings.

Example 1. Every *submodular objective* is fractionally subadditive. This means that, for example, our framework captures the maximum coverage problem, where we are given a weighted set of sets $E \subseteq 2^U$ over a universe U . Every element of U has a value $v: U \rightarrow \mathbb{R}_{\geq 0}$ associated with it, and $f(S) = v(\bigcup_{X \in S} X)$ for all $S \subseteq E$ where we write $v(X) := \sum_{x \in X} v(x)$ for a set $X \in 2^U$. In this context, the M -boundedness condition demands that $v(X) \in [1, M]$ for all $X \in E$. Further examples include maximization versions of clustering and location problems.

Example 2. An objective function $f: E \rightarrow \mathbb{R}$ is called *XOS* if it can be written as the pointwise maximum of linear functions, i.e., there are $k \in \mathbb{N}$ and values $v_{e,i} \in \mathbb{R}$ for all $e \in E$ and $i \in \{1, \dots, k\}$ such that $f(S) = \max_{\{1, \dots, k\}} \sum_{e \in S} v_{e,i}$ for all $S \subseteq E$. The set of fractionally subadditive functions and the set of XOS functions coincide (Feige [7]). XOS functions are a popular way to encode the valuations of buyers in combinatorial auctions since they often give rise to a succinct representation (cf. Nisan [24] and Lehman et al. [17]).

Example 3. The *weighted rank function of an independence system* is fractionally subadditive (Amanatidis et al. [1]). An independence system is a tuple (E, \mathcal{I}) , where $\mathcal{I} \subseteq 2^E$ is closed under taking subsets and $\emptyset \in \mathcal{I}$. For a given weight function $w': E \rightarrow \mathbb{R}_{\geq 0}$, the weighted rank function of (E, \mathcal{I}) is given by $f(S) = \max\{\sum_{e \in I} w'(e) \mid I \in \mathcal{I} \cap 2^S\}$. This setting captures well-known problems such as weighted d -dimensional matching for any $d \in \mathbb{N}$, weighted set packing, weighted maximum independent set, and knapsack (i.e., a secondary knapsack problem on the same set ground set).

Example 4. It turns out that a *potential-based flow between two vertices s and t along a set E of parallel edges* gives rise to a fractionally subadditive flow function (Proposition 2). A potential-based flow is given by a classical flow $f: E \rightarrow \mathbb{R}_{\geq 0}$ together with vertex potentials p_s and p_t , coupled via $p_s - p_t = \beta_e \psi(f(e))$ for all $e \in E$. Here, $\beta_e \in \mathbb{R}$ are parameters and ψ is a continuous, strictly

increasing, and odd potential loss function. Different choices of ψ allow to model gas flows, water flows, and electrical flows, see Groß et al. [9]. In our incremental framework, $w: E \rightarrow \mathbb{R}_{\geq 0}$ are interpreted as construction costs of pipes or cables and the objective is to maximize the flow from s to t in terms of the objective

$$f(S \subseteq E) = \max_{S' \subseteq S} \max_{p \in \mathbb{R}_{\geq 0}} \sum_{e \in S'} \psi^{-1}(p/\beta_e) \quad \text{s.t.} \quad \psi^{-1}(p/\beta_e) \leq \mu_e \text{ for all } e \in S', \quad (2)$$

where $p := p_s - p_t$ and μ_e are edge capacities. Note that we need to allow turning off edges (by considering $S' \subseteq S$) in order to make f monotone. The M -boundedness condition corresponds to the assumption that $\mu_e \in [1, M]$ and a ρ -competitive ordering determines a construction order of pipes or cables such that the flow that can be sent through the network is at least a $1/\rho$ -fraction of the optimum at all times.

Our results Our main results are bounds on the best possible competitive ratio for incremental solutions to (1) for objectives satisfying (MO), (MB), and (FS). In other words, we bound the loss in solution quality that we have to accept when asking for incremental solutions that optimize for all capacities simultaneously. Note that, as customary in online optimization, we do not impose restrictions on the computational complexity of finding incremental solutions. We show the following.

Theorem 1. *For monotone, M -bounded, and fractionally subadditive objectives, the best possible ρ for which the knapsack problem (1) admits a ρ -competitive incremental solution satisfies $\rho \in [\max\{2.449, M\}, \max\{3.293\sqrt{M}, 2M\}]$.*

In particular, for $M \geq 2.71$, the best possible competitive ratio is between M and $2M$, while the bounds for 1-bounded objectives simplify as follows.

Corollary 1. *For monotone, 1-bounded, and fractionally subadditive objectives, the best ρ for which the knapsack problem (1) admits a ρ -competitive incremental solution satisfies $\rho \in [2.449, 3.293]$.*

Most interestingly, our upper bound uses a simultaneous capacity- and value-scaling approach. In each phase, we increase our capacity and value thresholds and pick the smallest capacity for which the optimum solution exceeds our thresholds. This solution is then assembled by adding one element at a time in a specific order. The order is chosen based on a primal-dual LP formulation that relies on fractional subadditivity.

In Section 2, we describe our algorithmic approach in detail and give a proof of the upper bound. In Section 3, we complement our result with two lower bounds. As an additional motivation, in Section 4, we show that our framework captures potential-based flows as described in Example 4. In this context, a 1-bounded objective corresponds to unit capacities. As a contrast, we also show that classical s - t -flows with unit capacities admit 2-competitive incremental solutions, and this is best-possible.

Related Work Bernstein et al. [3] considered a closely related framework for incremental maximization. Their framework assumes a growing cardinality constraint, which is a special case of our problem in (1) when all elements $e \in E$ have unit weight $w(e) = 1$. A natural incremental approach for a growing cardinality constraint is the greedy algorithm that includes in each step the element that increases the objective the most. This algorithm is well known to yield a $e/(e-1)$ approximation for submodular objectives [23]. Several generalizations of this result to broader classes of functions are known. Recently, Disser and Weckbecker [5] unified these results by giving a tight bound for the

approximation ratio of the greedy algorithm for γ - α -augmentable functions, which interpolates between known results for weighted rank functions of independence systems of bounded rank quotient, functions of bounded submodularity ratio, and α -augmentable functions. Sviridenko [26] showed that for a submodular function under a knapsack constraint, the greedy algorithm yields a $(1 - 1/e)$ -approximation when combined with a partial enumeration procedure. This approximation guarantee is best possible as shown by Feige [6]. Yoshida [28] generalized the result of Sviridenko to submodular functions of bounded curvature.

Another closely related setting is the robust maximization of a linear function under a knapsack constraint. Here, the capacity of the knapsack is revealed in an online fashion while packing, and we ask for a packing order that guarantees a good solution for every capacity. Megow and Mestre [21] considered this setting under the assumption that we have to stop packing once an item exceeds the knapsack capacity and showed that no bounded competitive ratio is possible. Navarra and Pinotti [22] added the mild assumption that all items fit in the knapsack and devised competitive solutions for this model. Disser et al. [4] allowed to discard items that do not fit and showed tight competitive ratios for this case. Kawase et al. [15] studied a generalization of this model in which the objective is submodular and devised a randomised competitive algorithm for this case. Since these models allow to discard items, these competitive ratios do not translate to our model. Kobayashi and Takazawa [16] studied randomized strategies for cardinality robustness in the knapsack problem. Other online versions of the knapsack problem assume that items are revealed over time, e.g., see Matchetti-Spaccamela and Vercellis [19]. Thielen et al. [27] combined both settings and assumed that items appear over time while the capacity grows.

In terms of incremental minimization, Lin et al. [18] introduced a general framework, based on a problem-specific augmentation routine, that subsumes several earlier results. A minimization problem with growing cardinality constraint that received particular attention is the so-called robust matching problem introduced by Hassin and Rubinstein [11]. Here, we ask for a weighted matching such that the heaviest k edges of the matching approximate a maximum weight matching of cardinality k , for all cardinalities k . Hassin and Rubinstein [11] gave tight bounds on the deterministic competitive ratio of this problem, and Matuschke et al. [20] gave bounds on the randomized competitive ratio. Fujita et al. [8] and Kakimura et al. [12] considered extensions of this problem to independence systems. A similar variant of the knapsack problem where the k most valuable items are compared to an optimum solution of cardinality k was studied by Kakimura et al. [13].

Incremental optimization has also been considered from an offline perspective, i.e., without uncertainty in items or capacities. Kalinowski et al. [14] and Hartline and Sharp [10] considered incremental flow problems where the average flow over time needs to be maximized (in contrast to the worst flow over time). Anari et al. [2] and Orlin et al. [25] considered general robust submodular maximization problems.

The class of fractionally additive valuations was introduced by Nisan [24] and Lehman et al. [17] under the name of XOS-valuations as a compact way to represent the utilities of bidders in combinatorial auctions. In a combinatorial auction, a set of elements E is auctioned off to a set of n bidders who each have a private utility function $f_i: 2^E \rightarrow \mathbb{R}_{\geq 0}$. In this context, a natural question is to maximize social welfare, i.e., to partition E into sets E_1, E_2, \dots, E_n with the objective to maximize $\sum_{i=1}^n f_i(E_i)$. Feige [7] gave a $(1 - 1/e)$ -approximation for this problem.

2 Upper bound

In the following, we fix a ground set E , a monotone, M -bounded and fractionally subadditive objective $f: 2^E \rightarrow \mathbb{R}_{\geq 0}$, and weights $w: E \rightarrow \mathbb{R}_{\geq 0}$. We present a refined variant of the incremental algorithm introduced in [3]. On a high level, the idea is to consider optimum solutions of increasing

sizes, and to add all elements in these optimum solutions one solution at a time. By carefully choosing the order in which we add elements of a single solution, we ensure that elements contributing the most to the objective are added first. In this way, we can guarantee that either the solution we have assembled most recently, or the solution we are currently assembling provides sufficient value to stay competitive. While the algorithm of [3] only scales the capacity, we simultaneously scale capacities and solution values. In addition, we use a more sophisticated order in which we assemble solutions, based on a primal-dual LP formulation. We now describe our approach in detail.

Let $\lambda \approx 3.2924$ be the unique real root of the equation $0 = \lambda^7 - 2\lambda^6 - 3\lambda^5 - 3\lambda^4 - 3\lambda^3 - 2\lambda^2 - \lambda - 1$, $\delta := \frac{\lambda^3}{\lambda^2+1} \approx 3.0143$ and $\rho := \max\{\lambda\sqrt{M}, 2M\}$. Our algorithm $\text{ALG}_{\text{scale}}$ operates in phases of increasing capacities $C_1, \dots, C_N \in \mathbb{R}_{\geq 0}$ with $C_1 := \min_{e \in E} w(e)$, $C_i = \min\{C \geq \delta C_{i-1} \mid f(S_C^*) \geq \rho f(S_{C_{i-1}}^*)\}$ for $i \in \{2, \dots, N-1\}$, and $C_N := \sum_{e \in E} w(e) \in (C_{N-1}, \delta C_{N-1}]$, where $N \in \mathbb{N}$ is chosen accordingly. In phase $i \in \{1, \dots, N\}$, $\text{ALG}_{\text{scale}}$ adds the elements of the set $S_{C_i}^*$ one at a time. We may assume that previously added elements are added again (without any benefit), since this only hurts the algorithm.

To specify the order in which the elements of $S_{C_i}^*$ are added, consider the following linear program (LP_X) parameterized by $X \subseteq E$ (cf. [7]):

$$\begin{aligned} \min \quad & \sum_{B \subseteq E} \alpha_B f(B) \\ \text{s.t.} \quad & \sum_{B \subseteq E: e \in B} \alpha_B \geq 1, \quad \forall e \in X, \\ & \alpha_B \geq 0, \quad \forall B \subseteq E, \end{aligned}$$

and its dual

$$\begin{aligned} \max \quad & \sum_{e \in X} \gamma_e \\ \text{s.t.} \quad & \sum_{e \in B} \gamma_e \leq f(B), \quad \forall B \subseteq E, \\ & \gamma_e \geq 0, \quad \forall e \in E. \end{aligned}$$

Fractional subadditivity of f translates to $f(X) \leq \sum_{B \subseteq E} \alpha_B f(B)$ for all $\alpha \in \mathbb{R}^{2^E}$ feasible for (LP_X). The solution $\alpha^* \in \mathbb{R}^{2^E}$ with $\alpha_X^* = 1$ and $\alpha_B^* = 0$ for $X \neq B \subseteq E$ is feasible and satisfies $f(X) = \sum_{B \subseteq E} \alpha_B^* f(B)$. Together this implies that α^* is an optimum solution to (LP_X). By strong duality, there exists an optimum dual solution $\gamma^*(X) \in \mathbb{R}^E$ with

$$f(X) = \sum_{e \in X} \gamma_e^*(X). \quad (3)$$

In phase 1, the algorithm $\text{ALG}_{\text{scale}}$ adds the unique element in $S_{C_1}^*$. In phase 2, $\text{ALG}_{\text{scale}}$ adds an element of largest weight in $S_{C_2}^*$ first and the other elements in an arbitrary order. And in phase $i \in \{3, 4, \dots, N\}$, $\text{ALG}_{\text{scale}}$ adds the elements of $S_{C_i}^*$ in an order $(e_1, \dots, e_{|S_{C_i}^*|})$ such that, for all $j \in \{1, \dots, |S_{C_i}^*| - 1\}$,

$$\frac{\gamma_{e_j}^*(S_{C_i}^*)}{w(e_j)} \geq \frac{\gamma_{e_{j+1}}^*(S_{C_i}^*)}{w(e_{j+1})}. \quad (4)$$

For $C \in [0, C_i]$ and with $j := \max\{j \in \{1, \dots, |S_{C_i}^*|\} \mid w(\{e_1, \dots, e_j\}) \leq C\}$, we let $S_{C_i, C}^* := \{e_1, \dots, e_j\}$ denote the prefix of $S_{C_i}^*$ of capacity C . Furthermore, by π^A , we refer to the the permutation of E that represents the order in which the algorithm $\text{ALG}_{\text{scale}}$ adds the elements of E .

We first show that the dual variables $\gamma_e^*(X)$ associate a contribution to the overall objective to each element $x \in X$, and that this association is consistent under taking subsets of X .

Lemma 1. *Let $X \subseteq Y \subseteq E$. Then,*

$$f(X) \geq \sum_{e \in X} \gamma_e^*(Y).$$

Proof. Since $\gamma^*(Y)$ is a feasible solution for the dual of (LP_X) , it is also a feasible solution for the dual of (LP_Y) . Thus, since $\gamma^*(X)$ is an optimum solution of (LP_X) ,

$$\sum_{e \in X} \gamma_e^*(Y) \leq \sum_{e \in X} \gamma_e^*(X) \stackrel{(3)}{=} f(X). \quad \square$$

The following lemma establishes that the order in which we add the elements of each optimum solution are decreasing in density, in an approximate sense.

Lemma 2. *Let $C, C' \in \mathbb{R}_{\geq 0}$ with $C \leq C' \leq w(E)$. Then*

$$f^*(C') \leq \frac{C'}{C}(f(S_{C',C}^*) + M).$$

Proof. If $S_{C'}^* = S_{C',C}^*$, the statement holds trivially. Suppose $|S_{C'}^*| > |S_{C',C}^*|$. Let $j := |S_{C',C}^*|$, and let $S_{C'}^* = \{e_1, \dots, e_{|S_{C'}^*|}\}$ such that (4) holds. Note that, by definition, $S_{C',C}^* = \{e_1, \dots, e_j\}$ and

$$w(\{e_1, \dots, e_j\}) \leq C < w(\{e_1, \dots, e_{j+1}\}). \quad (5)$$

We have

$$\begin{aligned} f^*(C') &\stackrel{(3)}{=} \sum_{i=1}^{|S_{C'}^*|} w(e_i) \frac{\gamma_{e_i}^*(S_{C'}^*)}{w(e_i)} \\ &\stackrel{(4)}{\leq} \left(\sum_{i=1}^{j+1} \gamma_{e_i}^*(S_{C'}^*) \right) + \frac{\sum_{i=1}^{j+1} w(e_i)}{w(\{e_1, \dots, e_{j+1}\})} \sum_{i=j+2}^{|S_{C'}^*|} w(e_i) \frac{\gamma_{e_{j+1}}^*(S_{C'}^*)}{w(e_{j+1})} \\ &\stackrel{(4)}{\leq} \left(\sum_{i=1}^{j+1} \gamma_{e_i}^*(S_{C'}^*) \right) + \frac{\left(\sum_{i=1}^{j+1} \gamma_{e_i}^*(S_{C'}^*) \right)}{w(\{e_1, \dots, e_{j+1}\})} \sum_{i=j+2}^{|S_{C'}^*|} w(e_i) \\ &\stackrel{(5)}{<} \left(\sum_{i=1}^{j+1} \gamma_{e_i}^*(S_{C'}^*) \right) + \frac{\left(\sum_{i=1}^{j+1} \gamma_{e_i}^*(S_{C'}^*) \right)}{C} (C' - C) \\ &= \frac{C'}{C} \left(\left(\sum_{i=1}^j \gamma_{e_i}^*(S_{C'}^*) \right) + \gamma_{e_{j+1}}^*(S_{C'}^*) \right) \\ &\stackrel{\text{Lemma 1}}{\leq} \frac{C'}{C} (f(\{e_1, \dots, e_j\}) + f(\{e_{j+1}\})) \\ &\leq \frac{C'}{C} (f(S_{C',C}^*) + M). \end{aligned}$$

□

Since every set $S \subseteq E$ with $w(S) \leq C$ satisfies $f(S) \leq f^*(C)$, and since we have $w(S_{C',C}^*) \leq C$, we immediately obtain the following.

Corollary 2. Let $C, C' \in \mathbb{R}_{\geq 0}$ with $C \leq C' \leq w(E)$. Then

$$f^*(C') \leq \frac{C'}{C}(f^*(C) + M).$$

With this, we are now ready to show the upper bound of our main result.

Theorem 2. The incremental solution computed by $\text{ALG}_{\text{scale}}$ is ρ -competitive for $\rho = \max\{\lambda\sqrt{M}, 2M\} \approx \max\{3.2924\sqrt{M}, 2M\}$.

Proof. We have to show that, for all sizes $C \in \mathbb{R}_{\geq 0}$, we have $f^*(C) \leq \rho f(\pi^A(C))$. We will do this by analyzing the different phases of the algorithm. Observe that, for all $i \in \{2, \dots, N-1\}$, we have

$$\begin{aligned} f^*(C_i) &\stackrel{\text{Def. of ALG}_{\text{scale}}}{\geq} \rho f^*(C_{i-1}) \\ &\geq \rho^{i-1} f^*(C_1) \\ &\stackrel{(\text{MB})}{\geq} \rho^{i-1} \\ &\stackrel{\text{Def. of } \rho}{\geq} (\lambda\sqrt{M})^{i-1}. \end{aligned} \tag{6}$$

In phase 1, we have $C \in (0, C_1]$. Since C_1 is the minimum weight of all elements and we start by adding $S_{C_1}^*$, $\pi^A(C)$ is optimal.

Consider phase 2, and suppose $C \in (C_1, C_2)$. If $C_2 > \delta C_1$ holds, then C_2 is the smallest value such that $f^*(C_2) \geq \rho f^*(C_1)$, i.e., by monotonicity of f , we have $f(\pi^A(C)) \geq f(\pi^A(C_1)) = f^*(C_1) > \frac{1}{\rho} f^*(C)$. Now assume $C_2 = \delta C_1$. If $C \in (C_1, 3C_1)$, i.e., any solution of size C cannot contain more than two elements, or if $C \in (C_1, C_2)$ and $S_{C_2}^*$ contains at most 2 elements, by fractional subadditivity and M -boundedness of f , we have $f^*(C) \leq |S_{C_2}^*|M \leq 2M$ and thus, by (6), $f(\pi^A(C)) \geq f^*(C_1) \geq 1 \geq \frac{1}{2M} f^*(C) \geq \frac{1}{\rho} f^*(C)$. Now suppose $C \in [3C_1, C_2 = \delta C_1)$ and that $S_{C_2}^*$ contains at least 3 elements. The prefix $\pi^A(C_1 + C_2)$ contains all elements from $S_{C_1}^* \cup S_{C_2}^*$, the prefix $\pi^A(C_2) = \pi^A(C_1 + C_2 - C_1)$ contains at least all but one element of $S_{C_2}^*$, and the prefix $\pi^A(C_2 - C_1)$ contains at least all but 2 elements from $S_{C_2}^*$ because C_1 is the weight of any element is at least C_1 . Since $C \geq 3C_1 > (\delta - 1)C_1 = C_2 - C_1$, $\pi^A(C)$ contains at least all but 2 elements from $S_{C_2}^*$. Recall that in phase 2 the algorithm adds the element with the highest objective value first. Therefore, and because $|S_{C_2}^*| \geq 3$, we have $f(\pi^A(C)) \geq \frac{1}{3} f(S_{C_2}^*) \geq \frac{1}{\rho} f^*(C)$.

Consider phase 2 and suppose $C \in [C_2, C_1 + C_2]$. We have

$$f^*(C_1 + C_2) \leq f^*(C_2) + M \tag{7}$$

because f is subadditive and because C_1 is the minimum weight of all elements. Furthermore, we have

$$f(\pi^A(C_2)) \geq f^*(C_2) - M \geq \rho - M \geq M \tag{8}$$

where the first inequality follows from subadditivity of f and the fact that the prefix $\pi^A(C_2)$ contains at least all but one element from $S_{C_2}^*$. Combining (7) and (8), we obtain $f(\pi^A(C_2)) \geq f^*(C_1 + C_2) - 2M \geq f^*(C_1 + C_2) - 2f(\pi^A(C_2))$, i.e., by monotonicity, $f^*(C) \leq f^*(C_1 + C_2) \leq 3f(\pi^A(C_2)) \leq \rho f(\pi^A(C_2)) \leq \rho f(\pi^A(C))$.

Now consider phase $i \in \{3, \dots, N\}$ and $C \in (\sum_{j=1}^{i-1} C_j, \sum_{j=1}^i C_j]$. Note that, for $j \in \{1, \dots, i\}$, $C_i \geq \delta^{i-j} C_j$ and hence

$$\sum_{j=1}^{i-1} \frac{C_j}{C_i} \leq \sum_{j=1}^{i-1} \frac{1}{\delta^{i-j}} < \sum_{j=1}^{\infty} \frac{1}{\delta^j} = \frac{1}{\delta - 1} < 1,$$

i.e., we have $\sum_{j=1}^{i-1} C_j \leq C_i \leq \sum_{j=1}^i C_j$. If $i = N$ and $\sum_{j=1}^{i-1} C_j \geq C_i$, we have nothing left to show. Thus, suppose that $\sum_{j=1}^{N-1} C_j \leq C_N$.

Suppose $C \in (\sum_{j=1}^{i-1} C_j, C_i)$. If $C_i > \delta C_{i-1}$ holds, then C_i is the smallest integer such that $f^*(C_i) \geq \rho f^*(C_{i-1})$, i.e., we have $f(\pi^A(C)) \stackrel{\text{(MO)}}{\geq} f(\pi^A(C_{i-1})) \geq f^*(C_{i-1}) > \frac{1}{\rho} f^*(C)$. For the case that $C_i = \delta C_{i-1}$, we distinguish between two different cases:

Case 1 ($i = 3$): Let $c := (\frac{1}{\lambda\sqrt{M}} + \frac{1}{\lambda^2})\delta C_2$. Note that $(\frac{1}{\lambda} + \frac{1}{\lambda^2})\delta = \frac{\lambda^2}{\lambda+1} - 1 - \frac{1}{\delta}$ by definition of λ and δ and thus

$$\begin{aligned}
c &\leq \left(\frac{1}{\lambda} + \frac{1}{\lambda^2}\right)\delta C_2 \\
&= \left(\frac{\lambda^2}{\lambda+1} - 1 - \frac{1}{\delta}\right)C_2 \\
&\leq \frac{\lambda^2}{\frac{\lambda}{\sqrt{M}} + 1}C_2 - C_2 - C_1 \\
&= \frac{\lambda^2 M}{\lambda\sqrt{M} + M}C_2 - C_2 - C_1.
\end{aligned} \tag{9}$$

We will show that $\pi^A(C_1 + C_2)$ is competitive up to size $C_1 + C_2 + c$, and that $\pi^A(C_1 + C_2 + c)$ is competitive up to size C_3 . We have

$$\begin{aligned}
f^*(C_1 + C_2 + c) &\stackrel{\text{Corollary 2}}{\leq} \frac{C_1 + C_2 + c}{C_2}(f^*(C_2) + M) \\
&\stackrel{(9)}{\leq} \frac{C_1 + C_2 + \left(\frac{\lambda^2 M}{\lambda\sqrt{M} + M}C_2 - C_2 - C_1\right)}{C_2}(f^*(C_2) + M) \\
&= \frac{\lambda^2 M}{\lambda\sqrt{M} + M} \left(1 + \frac{M}{f^*(C_2)}\right) f^*(C_2) \\
&\stackrel{(6)}{\leq} \frac{\lambda^2 M}{\lambda\sqrt{M} + M} \left(1 + \frac{M}{\lambda\sqrt{M}}\right) f^*(C_2) \\
&= \lambda\sqrt{M} \frac{\lambda\sqrt{M}}{\lambda\sqrt{M} + M} \left(\frac{\lambda\sqrt{M} + M}{\lambda\sqrt{M}}\right) f^*(C_2) \\
&\leq \rho f^*(C_2) \\
&\leq \rho f(\pi^A(C_1 + C_2)),
\end{aligned}$$

where the last inequality follows from the fact that the algorithm starts by packing $S_{C_1}^*$ and $S_{C_2}^*$ before any other elements and needs capacity $C_1 + C_2$ to assemble both sets, i.e., $S_{C_2}^* \subseteq \pi^A(C_1 + C_2)$.

Since $\text{ALG}_{\text{scale}}$ adds the elements from $S_{C_3}^*$ after those from $S_{C_1}^*$ and $S_{C_2}^*$, we have $S_{C_3,c}^* \subseteq \pi^A(C_1 + C_2 + c)$, and thus

$$\begin{aligned}
f(\pi^A(C_1 + C_2 + c)) &\stackrel{\text{(MO)}}{\geq} f(S_{C_3,c}^*) \\
&\stackrel{\text{Lemma 2}}{\geq} \frac{c}{C_3} f^*(C_3) - M \\
&\stackrel{C_3 = \delta C_2}{=} \left(\left(\frac{1}{\lambda \sqrt{M}} + \frac{1}{\lambda^2} \right) - \frac{M}{f^*(C_3)} \right) f^*(C_3) \\
&\stackrel{(6)}{\geq} \left(\frac{1}{\lambda \sqrt{M}} + \frac{1}{\lambda^2} - \frac{M}{\lambda^2 M} \right) f^*(C_3) \\
&= \frac{1}{\lambda \sqrt{M}} f^*(C_3) \\
&\geq \frac{1}{\rho} f^*(C_3).
\end{aligned}$$

This, together with monotonicity of f , implies $f^*(C) \leq \rho f(\pi^A(C))$ for all $C \in (C_1 + C_2, C_3]$.

Case 2 ($i \geq 4$): Recall that $C \in (\sum_{j=1}^{i-1} C_j, C_i)$. We have

$$\begin{aligned}
f^*(C) &\stackrel{\text{(MO)}}{\leq} f^*(C_i) \\
&\stackrel{\text{Corollary 2}}{\leq} \frac{C_i}{C_{i-1}} (f^*(C_{i-1}) + M) \\
&\stackrel{C_i = \delta C_{i-1}}{=} \delta \left(1 + \frac{M}{f^*(C_{i-1})} \right) f^*(C_{i-1}) \\
&\stackrel{(6)}{\leq} \delta \left(1 + \frac{M}{\lambda^2 M} \right) f^*(C_{i-1}) \\
&= \frac{\lambda^3}{\lambda^2 + 1} \left(1 + \frac{1}{\lambda^2} \right) f^*(C_{i-1}) \\
&= \lambda f^*(C_{i-1}) \\
&\leq \rho f(\pi^A(C)).
\end{aligned}$$

Thus, also in this case, we found $f^*(C) \leq \rho f(\pi^A(C))$ for all $C \in (\sum_{j=1}^{i-1} C_j, C_i)$.

Now, consider $C \in [C_i, \sum_{j=1}^i C_j]$. Up to this budget, the algorithm had a capacity of $C - \sum_{j=1}^{i-1} C_j > C - C_i \geq 0$ to pack elements from $S_{C_i}^*$, i.e., $S_{C_i, C - \sum_{j=1}^{i-1} C_j}^* \subseteq \pi^A(C)$. This yields

$$\begin{aligned}
f(\pi^A(C)) &\stackrel{\text{(MO)}}{\geq} f(S_{C_i, C - \sum_{j=1}^{i-1} C_j}^*) \\
&\stackrel{\text{Lemma 2}}{\geq} \frac{C - \sum_{j=1}^{i-1} C_j}{C_i} f^*(C_i) - M \\
&\stackrel{\text{Corollary 2}}{\geq} \frac{C - \sum_{j=1}^{i-1} C_j}{C_i} \left(\frac{C_i}{C} f^*(C) - M \right) - M \\
&= \left(\frac{C - \sum_{j=1}^{i-1} C_j}{C} - \frac{C - \sum_{j=1}^{i-1} C_j}{C_i} \cdot \frac{M}{f^*(C)} - \frac{M}{f^*(C)} \right) f^*(C) \\
&\stackrel{C_i \leq C \leq \sum_{j=1}^i C_i}{\geq} \left(1 - \sum_{j=1}^{i-1} \frac{C_j}{C_i} - 1 \cdot \frac{M}{f^*(C)} - \frac{M}{f^*(C)} \right) f^*(C) \\
&\stackrel{(6), C_{j+1} \geq \delta C_j}{\geq} \left(1 - \sum_{j=1}^{i-1} \frac{1}{\delta^{i-j}} - \frac{2M}{\rho^{i-1}} \right) f^*(C) \\
&\geq \left(1 - \sum_{j=1}^{\infty} \frac{1}{\delta^j} - \frac{2M}{\rho^{i-1}} \right) f^*(C) \\
&= \left(1 - \left(\frac{1}{1 - \delta^{-1}} - 1 \right) - \frac{2M}{\lambda^2 M} \right) f^*(C) \\
&\geq 0.319 \cdot f^*(C) \\
&\geq \frac{1}{\rho} f^*(C).
\end{aligned}$$

□

For 1-bounded objectives, Theorem (2) immediately yields the following.

Corollary 3. *If $M = 1$, the incremental solution computed by $\text{ALG}_{\text{scale}}$ is 3.2924-competitive.*

3 Lower bound

In this section, we show the second part of Theorem 1, i.e., we give a lower bound on the competitive ratio for the incremental knapsack problem (1) with monotone, M -bounded, and fractionally subadditive objectives, and we show a lower bound for the special case with $M = 1$.

Proposition 1. *For monotone, M -bounded, and fractionally subadditive objectives, the knapsack problem (1) does not admit a ρ -competitive incremental solution for $\rho < M$.*

Proof. Consider the set $E = \{e_1, e_2\}$ with weights $w(e_i) = i$ for $i \in \{1, 2\}$ and the values $v(e_1) = 1$ and $v(e_2) = M$. We define the objective $f(X \subseteq E) := \sum_{e \in X} v(e)$. It is easy to see that f is monotone, M -bounded and modular and thus fractionally subadditive.

Consider some competitive algorithm with competitive ratio $\rho \geq 0$ for the knapsack problem (1). In order to be competitive for capacity 1, the algorithm has to add element e_1 first. Thus, the solution of the algorithm of size 2 cannot contain element e_2 , i.e., the value of the solution of capacity 2 given by the algorithm has value 1. The optimal solution of capacity 2 has value M , and thus $\rho \geq M$. □

For $M \in [1, \sqrt{6})$, we obtain a stronger lower bound.

Theorem 3. *For monotone, 1-bounded, and fractionally subadditive objectives, the knapsack problem (1) does not admit a ρ -competitive incremental solution for $\rho < \sqrt{6}$.*

Proof. Consider the set $E = E_1 \cup E_2 \cup E_3 = \{e_1\} \cup \{e_2, e_3, e_4\} \cup \{e_5, \dots, e_{10}\}$ the weights

$$w(e) = \begin{cases} 101, & \text{for } e \in E_1, \\ 102, & \text{for } e \in E_2, \\ 103, & \text{else,} \end{cases}$$

and the objective $f: E \rightarrow \mathbb{R}_{\geq 0}$ with

$$f(X \subseteq E) = \max\{|X \cap E_1|, \frac{\sqrt{6}}{3}|X \cap E_2|, |X \cap E_3|, |X \cap \{e_2\}|, |X \cap \{e_3\}|, |X \cap \{e_4\}|\}.$$

It is easy to see that $f(e) = 1$ for all $e \in E$ and that f is monotone. Furthermore, f is fractionally subadditive because it is an XOS-valuation since all terms in the maximum of the definition of f are modular set functions on E .

Suppose there was an algorithm ALG with competitive ratio $\rho < \sqrt{6}$. Let $\pi: \{1, \dots, 10\} \rightarrow \{1, \dots, 10\}$ be a permutation that represents the order in which ALG adds the elements of E . To be competitive for capacity $C = 101$, we must have $\pi(1) = 1$. For capacity $C = 306$, the optimum solution E_2 has value $\sqrt{6}$. To be competitive for this capacity, ALG has to achieve an objective value of $\frac{\sqrt{6}}{\rho} > 1$. This is not possible by adding one element of E_2 and one of E_3 . Furthermore, adding two elements of E_3 is not possible with a capacity of C , since the elements in E_3 have weight 103, i.e., we must have $\{\pi(2), \pi(3)\} \subset \{2, 3, 4\}$. Without loss of generality, we can assume $(\pi(2), \pi(3)) = (2, 3)$. We distinguish between two different cases:

Case 1: $\pi(4) = 4$. We consider the capacity $C = 6 \cdot 103 = 618$. The solution of capacity at most C given by ALG can contain at most two elements of E_3 . Thus, the value of the solution of ALG of capacity $C = 618$ is $\sqrt{6}$ which is not ρ -competitive since the optimum solution E_3 of capacity $C = 618$ has value 6 and $\rho\sqrt{6} < 6$.

Case 2: $\pi(4) \in \{5, \dots, 10\}$. We consider the capacity $C = 5 \cdot 103 = 515$. The solution of capacity at most C given by ALG can contain at most two elements of E_3 . Thus, the value of the solution of ALG of capacity $C = 515$ is 2, which is not ρ -competitive since the optimum solution E_3 of capacity $k = 515$ has value 5, and we have $2\rho < 5$.

This is a contradiction to the fact that ALG has competitive ratio $\rho < \sqrt{6}$. □

4 Application to flows

In this section we show that our algorithm $\text{ALG}_{\text{scale}}$ can be used to solve problems as given in Example 4, where we are given a graph $G = (V, E)$ consisting of two nodes s and t with a collection of edges between them, and want to determine an order in which to build the edges while maintaining a potential-based flow between s and t that is as large as possible. As mentioned in Example 4, the corresponding objective (2) is monotone and M -bounded for $\mu_e \in [1, M]$. Now we show fractional subadditivity.

Proposition 2. *The function*

$$f(S \subseteq E) = \max_{S' \subseteq S} \max_{p \in \mathbb{R}_{\geq 0}} \sum_{e \in S'} \psi^{-1}(p/\beta_e) \quad \text{s.t.} \quad \psi^{-1}(p/\beta_e) \leq \mu_e \text{ for all } e \in S'$$

is fractionally subadditive.

Proof. For $i \in \{1, \dots, |E|\}$, let

$$p_i := \beta_e \psi(\mu_e)$$

be the maximum potential difference between s and t such that the flow along p_i is still feasible. For $i \in \{1, \dots, k\}$ and $e \in E$, we define $x_e(p_i)$ to be the flow value along e induced by a potential difference of p_i between s and t if this flow is feasible and 0 otherwise. For $S \subseteq E$, we have

$$\begin{aligned} f(S) &= \max_{S' \subseteq S} \max_{p \in \mathbb{R}_{\geq 0}} \sum_{e \in S'} \psi^{-1}(p/\beta_e) \quad \text{s.t.} \quad \psi^{-1}(p/\beta_e) \leq \mu_e \text{ for all } e \in S' \\ &= \max_{i \in \{1, \dots, |E|\}} \sum_{e \in S} x_e(p_i), \end{aligned}$$

i.e., f is an XOS-function and thus fractionally subadditive (see Example (2)). \square

We now turn to an incremental variant of classical s - t -flows. In the problem INCREMENTAL MAXIMUM s - t -FLOW we are given a graph $G = (V, E)$ and consider the objective

$$f(X \subseteq E) = \max\{F \mid \text{there exists an } s\text{-}t\text{-flow of value } F \text{ in } (V, X)\}.$$

We restrict ourselves to the case where all edges have unit weight and unit capacity.

To solve this problem, we describe the algorithm QUICKEST-INCREMENT from [14]: The algorithm starts by adding the shortest path and then iteratively adds the smallest set of edges that increase the maximum flow value by at least 1. Let $r \in \mathbb{N}$ be the number of iterations until QUICKEST-INCREMENT terminates. For $i \in \{0, 1, \dots, r\}$, let λ_i be the size of the set added in iteration i , i.e., λ_0 is the length of the shortest s - t -path, λ_1 the size of the set added in iteration 1, and so on. For $k \in \{1, \dots, |E|\}$, we denote the solution of size k of the algorithm by S_k^A .

With $X_{\max} \in \mathbb{R}_{\geq 0}$ defined as the maximal possible s - t -flow value in the underlying graph, for $j \in \{1, \dots, \lfloor X_{\max} \rfloor\}$, we denote by c_j the minimum number of edges required to achieve a flow value of at least j .

In [14] the authors relate the values λ_i and c_j in the following way.

Lemma 3. *In the unit capacity case, we have $\lambda_i \leq c_j/(j - i)$ for all $i, j \in \mathbb{N}$ with $0 \leq i < j \leq r$.*

Using this estimate, we can find a bound on the competitive ratio of QUICKEST-INCREMENT.

Theorem 4. *The algorithm QUICKEST-INCREMENT has a competitive ratio of at most 2 for the problem INCREMENTAL MAXIMUM s - t -FLOW with unit capacities.*

Proof. Note that, since we consider the unit capacity case, we have $X_{\max} - 1 = r \in \mathbb{N}$ because QUICKEST-INCREMENT increases the value of the solution by exactly 1 in each iteration.

Consider some size $k \in \{1, \dots, |E|\}$. If $k < c_1$, we have $f(S_k^*) = 0$, i.e., every solution is competitive. If $k \geq c_1$, let $j := f(S_k^*)$. Note that $f(S_{c_j}^*) = j = f(S_k^*)$ and therefore $k \geq c_j$. By Lemma 3, we

have

$$\begin{aligned}
\sum_{i=0}^{\lceil \frac{j}{2} \rceil - 1} \lambda_i &\leq \sum_{i=0}^{\lceil \frac{j}{2} \rceil - 1} \frac{c_j}{j-i} \\
&= c_j \sum_{i=0}^{\lceil \frac{j}{2} \rceil - 1} \frac{1}{j-i} \\
&\leq c_j \sum_{i=0}^{\lceil \frac{j}{2} \rceil - 1} \frac{1}{j - \lceil \frac{j}{2} \rceil + 1} \\
&= c_j \lceil \frac{j}{2} \rceil \frac{1}{\lceil \frac{j}{2} \rceil + 1} \\
&\leq c_j.
\end{aligned} \tag{10}$$

This implies $f(S_k^A) \geq f(S_{c_j}^A) \stackrel{(10)}{\geq} \lceil \frac{j}{2} \rceil \geq \frac{1}{2}j = \frac{1}{2}f(S_k^*)$. □

As it turns out, there exists no algorithm with a better competitive ratio than QUICKEST-INCREMENT.

Theorem 5. *There is no algorithm with a competitive ratio of less than 2 for the problem INCREMENTAL MAXIMUM s - t -FLOW with unit capacities.*

Proof. Consider the graph $G = (V, E)$ with

$$\begin{aligned}
V &:= \{s, t, u_1, u_2, u_3, v_1, v_2, v_3\}, \\
E &:= \{(s, u_1), (s, v_1), (u_1, u_2), (u_2, u_3), (u_3, t), (v_1, v_2), (v_2, v_3), (v_3, t), (u_1, v_3)\},
\end{aligned}$$

with unit capacities (cf. Figure 1).

We consider the problem INCREMENTAL MAXIMUM s - t -FLOW and an algorithm ALG which finds an approximation for this problem. In order to be competitive for size $k = 3$, ALG has to add the blocking path s, u_1, v_3, t first. But this implies that the solution of size $k = 8$ does not contain both paths, s, u_1, u_2, u_3, t and s, v_1, v_2, v_3, t , i.e., the optimal solution of size $k = 8$ has flow value 2 while the solution of ALG has flow value 1. Thus, the competitive ratio of ALG cannot be smaller than 2. □

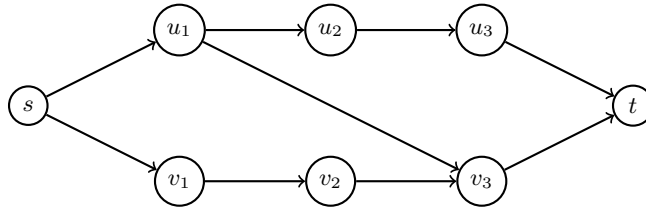


Fig. 1. A lower bound instance with best possible competitive ratio 2 for the problem INCREMENTAL MAXIMUM s - t -FLOW

References

1. Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. Coverage, matching, and beyond: New results on budgeted mechanism design. In *Proceedings of the 12 International Conference on Web and Internet Economics (WINE)*, pages 414–428, 2016.
2. Nima Anari, Nika Haghtalab, Seffi Naor, Sebastian Pokutta, Mohit Singh, and Alfredo Torrico. Structured robust submodular maximization: Offline and online algorithms. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89, pages 3128–3137, 2019.
3. Aaron Bernstein, Yann Disser, Martin Groß, and Sandra Himpburg. General bounds for incremental maximization. *Math. Program.*, 2020.
4. Yann Disser, Max Klimm, Nicole Megow, and Sebastian Stiller. Packing a knapsack of unknown capacity. *SIAM J. Discret. Math.*, 31(3):1477–1497, 2017.
5. Yann Disser and David Weckbecker. Unified greedy approximability beyond submodular maximization. arXiv:2011.00962, 2020.
6. Uriel Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.
7. Uriel Feige. On maximizing welfare when utility functions are subadditive. *SIAM J. Comput.*, 39(1):122–142, 2009.
8. Ryo Fujita, Yusuke Kobayashi, and Kazuhisa Makino. Robust matchings and matroid intersections. *SIAM J. Discret. Math.*, 27(3):1234–1256, 2013.
9. Martin Groß, Marc E. Pfetsch, Lars Schewe, Martin Schmidt, and Martin Skutella. Algorithmic results for potential-based flows: Easy and hard cases. *Networks*, 73(3):306–324, 2019.
10. Jeff Hartline and Alexa Sharp. Incremental flow. *Networks*, 50(1):77–85, 2007.
11. Refael Hassin and Shlomi Rubinfeld. Robust matchings. *SIAM J. Discret. Math.*, 15(4):530–537, 2002.
12. Naonori Kakimura and Kazuhisa Makino. Robust independence systems. *SIAM J. Discret. Math.*, 27(3):1257–1273, 2013.
13. Naonori Kakimura, Kazuhisa Makino, and Kento Seimi. Computing knapsack solutions with cardinality robustness. *Jpn. J. Ind. Appl. Math.*, 29(3):469–483, 2012.
14. Thomas Kalinowski, Dmytro Matsypura, and Martin W.P. Savelsbergh. Incremental network design with maximum flows. *Eur. J. Oper. Res.*, 242(1):51–62, 2015.
15. Yasushi Kawase, Hanna Sumita, and Takuro Fukunaga. Submodular maximization with uncertain knapsack capacity. *SIAM J. Discret. Math.*, 33(3):1121–1145, 2019.
16. Yusuke Kobayashi and Kenjiro Takazawa. Randomized strategies for cardinality robustness in the knapsack problem. *Theor. Comput. Sci.*, 699:53–62, 2017.
17. Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games Econ. Behav.*, 55(2):270–296, 2006.
18. Guolong Lin, Chandrashekar Nagarajan, Rajmohan Rajaraman, and David P. Williamson. A general approach for incremental approximation and hierarchical clustering. *SIAM J. Comput.*, 39(8):3633–3669, 2010.
19. Alberto Marchetti-Spaccamela and Carlo Vercellis. Stochastic on-line knapsack problems. *Math. Program.*, 68:73–104, 1995.
20. Jannik Matuschke, Martin Skutella, and José A. Soto. Robust randomized matchings. *Math. Oper. Res.*, 43(2):675–692, 2018.
21. Nicole Megow and Julián Mestre. Instance-sensitive robustness guarantees for sequencing with unknown packing and covering constraints. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 495–504, 2013.
22. Alfredo Navarra and Cristina M. Pinotti. Online knapsack of unknown capacity: How to optimize energy consumption in smartphones. *Theor. Comput. Sci.*, 697:98–109, 2017.
23. George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14:265–294, 1978.
24. Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC)*, pages 1–12, 2000.
25. James B. Orlin, Andreas S. Schulz, and Rajan Udvani. Robust monotone submodular function maximization. *Math. Program.*, 172(1-2):505–537, 2018.
26. Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32:41–43, 2004.
27. Clemens Thielen, Morten Tiedemann, and Stephan Westphal. The online knapsack problem with incremental capacity. *Math. Methods Oper. Res.*, 83(2):207–242, 2016.
28. Yuichi Yoshida. Maximizing a monotone submodular function with a bounded curvature under a knapsack constraint. *SIAM J. Discret. Math.*, 33(3):1452–1471, 2019.