FRACTIONALLY SUBADDITIVE MAXIMIZATION UNDER AN INCREMENTAL KNAPSACK CONSTRAINT WITH APPLICATIONS TO INCREMENTAL FLOWS*

YANN DISSER†, MAX KLIMM‡, ANNETTE LUTZ§, AND DAVID WECKBECKER¶

Abstract. We consider the problem of maximizing a fractionally subadditive function under an increasing knapsack constraint. 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 $\max\{2.618, M\}$ on the attainable competitive ratio. In addition, we establish that our framework captures potential-based flows between two vertices, and we give a lower bound of $\max\{2, M\}$ and an upper bound of $2M$ for the incremental maximization of classical flows with capacities in $[1, M]$ which is tight for the unit capacity case.

Key words. incremental optimization, competitive analysis, fractional subadditivity, flow

MSC codes. 68W27, 68W40, 90C17

1. Introduction. The decisions involved in large-scale infrastructure projects or in scheduling expensive investments usually have an impact over a prolonged period of time. This paper examines the question how investment or construction decisions can be made over time as the total budget grows, in such a way that the resulting solution is good at every point in time. This is a natural question for long-term infrastructure projects such as the construction of road networks, public transport systems, and energy networks. In addition, it is relevant for the investment decisions of businesses in manufacturing or distribution infrastructure.

Mathematically, we model the above settings in terms of the *incremental optimization problem*. Formally, we are given a ground set E of *elements* that can be invested in. Each element $e \in E$ has a weight $w(e)$ that models the time or money that has to be spent on realizing the element. In the following, for a set $S \subseteq E$, we write $w(S) := \sum_{e \in S} w(e)$. The value of having realized a subset $S \subseteq E$ of elements is given by a monotone objective function $f\colon 2^E \to \mathbb{R}_{\geq 0}$. Given a capacity bound $C \in \mathbb{R}_{\geq 0}$, the maximum value that can be obtained with elements up to this total size is given by an optimum solution to the following mathematical optimization problem:

$$f^*(C) := \max\big\{ f(S) \;\big|\; S \subseteq E, w(S) \leq C \big\}. \tag{P}$$

Given $C \in \mathbb{R}_{\geq 0}$, we denote the optimal value of this optimization problem by $f^*(C)$ and a set $S \subseteq E$ for which the optimum is attained by $S^*(C)$, where for the later we break ties in an arbitrary but fixed manner in order to obtain a unique set $S^*(C)$.

We are interested in obtaining an *incremental* solution to the optimization problem (P) that yields a good value for *all* capacity bounds $C \in \mathbb{R}_{\geq 0}$. Formally, an *(incremental) solution* is an ordering $\pi = \big(e_{\pi(1)}, e_{\pi(2)}, \ldots, e_{\pi(m)}\big)$ of the elements of

* Submitted to the editors DATE. A preliminary version of this article appeared in [6].

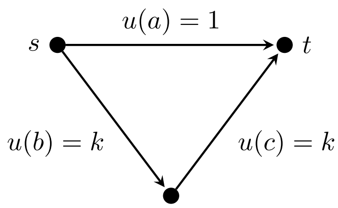
Funding: We acknowledge funding by the DFG through a short time project and subproject A007 of the CRC/TRR 154, as well as through grant DI 2041/2.

†TU Darmstadt (disser@mathematik.tu-darmstadt.de)

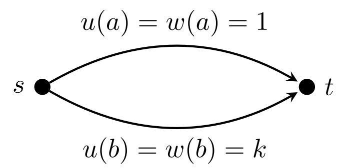
‡TU Berlin (klimm@math.tu-berlin.de)

§TU Darmstadt (lutz@mathematik.tu-darmstadt.de)

¶TU Darmstadt (weckbecker@mathematik.tu-darmstadt.de)



(a)



(b)

FIGURE 1.1. *Two examples of the incremental flow problem where no ρ-competitive solution with $\rho < k$ exists. In the left example, we have unit weights, i.e., $w(a) = w(b) = w(c) = 1$.*

the ground set E with $m = |E|$. For capacity $C \in \mathbb{R}_{\geq 0}$, let $\pi(C)$ be the items contained in the maximal prefix of π that fits into the capacity C, i.e.,

$$\pi(C) = \left\{e_{\pi(1)}, e_{\pi(2)}, \dots, e_{\pi(k)}\right\}$$

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 incremental solution π is *ρ-competitive* for some $\rho \geq 1$ if

$$f^*(C) \leq \rho\, f(\pi(C)) \quad \text{for all } C \in \mathbb{R}_{\geq 0}.$$

We call π *competitive* if it is ρ-competitive for some constant $\rho \geq 1$.

As an example, let us consider the special case of the *incremental maximum flow problem*. Here, the ground set E corresponds to the set of edges of an undirected graph $G = (V, E)$ with two designated vertices $s, t \in V$. Each edge has a weight $w(e) \in \mathbb{R}_{\geq 0}$ and a capacity $u(e) \in \mathbb{R}_{\geq 0}$. The value $f(S)$ of a subset $S \subseteq E$ is defined as the maximum value of an s-t-flow in $G_S = (V, S)$. Even in this special case, a competitive solution may fail to exist. For illustration, consider the graph in Figure 1.1a. Every solution π has to put edge a first in order to be competitive for $C = 1$. On the other hand, every solution that puts edge a first is not better than k-competitive for $C = 2$. Likewise, for the graph in Figure 1.1b, every competitive solution has to put edge a first in order to be competitive for $C = 1$, and every solution that puts edge a first is not better than k-competitive for $C = k$.

A closer inspection of these two examples reveals that there are (at least) two effects that prevent the existence of competitive solutions. The first is the *complementarity* of elements. In the graph in Figure 1.1a, edges b and c are complementary in the sense that both edges together support a flow of k while a single one of these edges alone cannot support any flow. For the graph in Figure 1.1b, no two edges are complementary since the total flow supported by a subset of edges is here simply equal to the sum of the capacities of the edges. In this example, the non-existence of a competitive solution is caused by the fact that the edges are too heterogenous. More specifically, we have $f(\{a\}) = 1$, but $f(\{b\}) = k$, i.e., there are two singleton sets whose value differs by a factor of k.

As we will show, these are essentially the only two effects that prevent the existence of competitive solutions. More specifically, we will make two assumptions that exclude the two effects shown in Figure 1.1a and Figure 1.1b. First, to avoid complementarities between elements, we assume that f is *fractionally subadditive*. Formally, a function $f\colon 2^E \to \mathbb{R}_{\geq 0}$ is called *fractionally subadditive* if

$$f(A) \leq \sum_{i=1}^{k} \alpha_i f(B_i) \quad \text{for all } A, B_1, \dots, B_k \in 2^E \text{ and all } \alpha_1, \dots, \alpha_k \in \mathbb{R}_{\geq 0}$$
$$\text{such that} \sum_{i \in \{1,\dots,k\}: e \in B_i} \alpha_i \geq 1 \text{ for all } e \in A.$$

Observe that fractional subadditivity implies regular subadditivity, i.e., we have $f(A \cup B) \leq f(A) + f(B)$, but not vice-versa. Second, to avoid that there exist singleton sets that differ too much in their values, we assume that there is a constant $M \in \mathbb{R}_{\geq 0}$, $M \geq 1$ such that $f(\{e\}) \in [1, M]$ for all $e \in E$. We call such valuations *M-bounded.*

Summarizing the discussion, this paper considers incremental solutions to (P) under the following assumptions.

Assumption 1.1. *The function $f : 2^E \to \mathbb{R}_{\geq 0}$ has the following properties:*

(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 1.2 (Submodular objective). A function $f : 2^E \to \mathbb{R}_{\geq 0}$ is called *submodular* if $f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$ for all $A, B \in 2^E$. It was shown by Lehmann et al. [22] that every monotone submodular function is also fractionally subadditive.

As a consequence our framework captures, e.g., the *maximum coverage problem*, where we are given a weighted family of sets $E \subseteq 2^U$ over a universe U. Every element of U has a value $v\colon U \to \mathbb{R}_{\geq 0}$ associated with it, and $f(S) = v\left(\bigcup_{X \in S} X\right)$ 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 1.3 (XOS objective). An objective function $f\colon 2^E \to \mathbb{R}$ is called *XOS* if it can be written as the pointwise maximum of modular 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

$$f(S) = \max\left\{ \sum_{e \in S} v_{e,i} \;\middle|\; i \in \{1, \dots, k\} \right\} \quad \text{for all } S \subseteq E.$$

As shown by Feige [11], the set of fractionally subadditive functions and the set of XOS functions coincide. 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 [30] and Lehman et al. [22]).

Example 1.4 (Weighted rank function of an independence system). An independence system is a tuple $(E, \mathcal{I})$, where $\emptyset \in \mathcal{I}$ and $\mathcal{I} \subseteq 2^E$ is closed under taking subsets, i.e., $A \in \mathcal{I}$ whenever $A \subseteq B$ and $B \in \mathcal{I}$. For a given weight function $v\colon E \to \mathbb{R}_{\geq 0}$, the weighted rank function of $(E, \mathcal{I})$ is given by

$$f(S) = \max\left\{ \sum_{e \in I} v(e) \;\middle|\; I \in \mathcal{I} \cap 2^S \right\} \quad \text{for all } S \subseteq E.$$

As shown by Amanatidis et al. [1], the weighted rank function of an independence system is fractionally subadditive.

This setting captures well-known problems such as *weighted d-dimensional matching* for any $d \in \mathbb{N}$. Here, we are given sets $V_1, \dots, V_d$ such that $E \subseteq V_1 \times \dots \times V_d$ and a function $v : E \to \mathbb{R}_{\geq 0}$. The objective $f \colon 2^E \to \mathbb{R}_{\geq 0}$ maps a set $S \subseteq E$ to the maximum weight of a d-dimensional matching in S, i.e,

$$f(S) = \max\left\{ \sum_{e \in M} v(e) \;\middle|\; \begin{array}{l} M \subseteq S \text{ such that } v_i \neq v_i' \text{ for all } i \in \{1, \dots, d\} \\ \quad \text{for all } e = (v_1, \dots, v_d), e' = (v_1', \dots, v_d') \in M \text{ with } e \neq e' \end{array} \right\}.$$

In a similar vein, this setting also includes weighted set packing and weighted maximum independent set.

Example 1.5 (Potential-based flows). Consider a variant of the incremental flow problem on parallel edges as in Figure 1.1b. As before, every edge e has a capacity $u(e) \in \mathbb{R}_{\geq 0}$. In addition, we are given a continuous and strictly increasing potential-loss function $\psi : \mathbb{R} \to \mathbb{R}$ with $\lim_{x \to \infty} \psi(x) = \infty$ that describes the physical properties of the network. Every edge e has a *resistance* $\beta(e) \in \mathbb{R}_{\geq 0}$. A vector $x \in \mathbb{R}^E_{\geq 0}$ is a flow if $x_e \leq u(e)$ for all $e \in E$, and it is called a potential-based flow if there are vertex potentials $p_s, p_t \in \mathbb{R}_{\geq 0}$ such that

$$p_s - p_t = \beta(e)\psi(x_e) \quad \text{for all } e \in E.$$

The potentials correspond to physical properties at the nodes such as pressures or voltages; different choices of ψ allow to model gas flows, water flows, and electrical flows, see Groß et al. [13]. In our incremental framework, $w \colon E \to \mathbb{R}_{\geq 0}$ 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 $f \colon 2^E \to \mathbb{R}_{\geq 0}$ that maps a set $S \subseteq E$ to the value

$$f(S) = \max\left\{ \sum_{e \in T} \psi^{-1}\left(\frac{p}{\beta(e)}\right) \;\middle|\; T \subseteq S, p \in \mathbb{R}_{\geq 0} \text{ with } \psi^{-1}\left(\frac{p}{\beta(e)}\right) \leq u(e) \text{ for all } e \in T \right\},$$

where $p := p_s - p_t$. The value of the objective is the maximum value of a feasible potential-based s-t-flow where we allow turning off edges in $S \setminus T$ in order to make f monotone. The M-boundedness condition corresponds to the assumption that $u(e) \in [1, M]$. As we will show in Proposition 4.1, this objective is fractionally subadditive.

1.1. Our results. Our main results are bounds on the best possible competitive ratio for incremental solutions to (P) 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. To state our result, we denote by $\varphi = \frac{1}{2}(1 + \sqrt{5}) \approx 1.618$ the golden ratio.

THEOREM 1.6. *For monotone, M-bounded, and fractionally subadditive objectives, the best possible ρ for which the optimization problem* (P) *admits a ρ-competitive solution satisfies $\rho \in \big[\max\{\varphi+1, M\}, \max\{3.293\sqrt{M}, 2M\}\big]$.*

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 1.7. *For monotone, 1-bounded, and fractionally subadditive objectives, the best ρ for which the optimization problem* (P) *admits a ρ-competitive solution satisfies $\rho \in \big[\varphi+1, 3.293\big]$.*

Our upper bounds are shown by an algorithm that 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.

For the definition of the algorithm, we need access to two oracles. On the one hand, we need oracle access to the optimal solution of a given capacity; on the other hand, we need access to an XOS oracle. More information on this can be found in Remark 2.1.

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 1.5. In this context, a 1-bounded objective corresponds to unit capacities. As a contrast, we also show that classical s-t-flows with capacities in $[1, M]$ admit $2M$-competitive incremental solutions, and this is best-possible for the unit capacity case.

1.2. Related Work. Bernstein et al. [4] considered a closely related framework for incremental maximization. Their framework assumes a growing cardinality constraint, which is a special case of our problem in (P) 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 [29]. Several generalizations of this result to broader classes of functions are known. Recently, Disser and Weckbecker [7] unified these results by giving a tight bound for the approximation ratio of the greedy algorithm for γ-α-augmentable functions[1], 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 [33] 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 [10]. Yoshida [35] generalized the result of Sviridenko to submodular functions of bounded curvature.

Another closely related setting is the robust maximization of a modular 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 [26] considered this setting under the assumption that we have to stop packing once an item exceeds the knapsack

[1] A function is called γ-α-augmentable if, for all sets $X, Y \subseteq E$, there exists $y \in Y$ with $f(X \cup \{y\}) - f(X) \geq (\gamma f(X \cup Y) - \alpha f(X))/|Y|$.

objective f	**setting**	**lower bound**		**upper bound**	
additive		∞	[26]		
additive	with discarding	2	[5]	2	[5]
additive	with discarding, $f = w$	φ	[5]	φ	[5]
additive	$C \geq \max w(e)$	2	[28]	2	[28]
additive	$C \geq \max w(e),\ f = w$	φ	[28]	1.756	[28]
submodular	with discarding	2	[5]	2.794	[20]
frac. subadditive	$f(\{e\}) = 1$	$\varphi + 1$		3.293	
accountable	$w(e) = 1$	2.183	[4]	$\varphi + 1$	[4]

FIGURE 1.2. *Competitive ratios for deterministic incremental maximization of a monotone objective under a knapsack constraint in different settings where $\varphi = \frac{1}{2}(1+\sqrt{5}) \approx 1.617$ is the golden ratio. Note that additivity implies fractional subadditivity (XOS), which implies accountability.*

capacity and presented a polynomial time algorithm that has an instance-sensitive near-optimal competitive ratio. Navarra and Pinotti [28] added the mild assumption that all items fit in the knapsack and devised competitive solutions for this model. Disser et al. [5] allowed to discard items that do not fit and showed tight competitive ratios for this case. Kawase et al. [19] studied a generalization of this model in which the objective is submodular and devised a randomized competitive algorithm for this case. Klimm and Knaack [20] gave a deterministic competitive algorithm with improved competitive ratio for this case. Since the models in [5, 19, 20] allow to discard items, these competitive ratios do not translate to our model. Kobayashi and Takazawa [21] 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 [24]. Thielen et al. [34] combined both settings and assumed that items appear over time while the capacity grows. An overview over a selection of the aforementioned results can be found in Figure 1.2.

In terms of incremental minimization, Lin et al. [23] introduced a general framework, based on a problem-specific augmentation routine, that subsumes several earlier results. A maximization problem with growing cardinality constraint that received particular attention is the so-called robust matching problem introduced by Hassin and Rubinstein [15]. 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 [15] gave tight bounds on the deterministic competitive ratio of this problem, and Matuschke et al. [25] gave bounds on the randomized competitive ratio. Fujita et al. [12] and Kakimura et al. [16] 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. [17].

Incremental optimization has also been considered from an offline perspective, i.e., without uncertainty in items or capacities. Kalinowski et al. [18] and Hartline and Sharp [14] 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. [31] considered general robust submodular maximization problems. For maximization of (fractionally) subadditive objectives the approximation ratio is known to be $|E|^{-1/2}$ in the value oracle setting due to Dobzinski et al. [8], Singer [32], and Mirrokni et al. [27], and $2+\epsilon$ in the demand oracle setting due to Badanidiyuru [3].

The class of fractionally subadditive valuations was introduced by Nisan [30] and Lehman et al. [22] 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 $f_i\colon 2^E \to \mathbb{R}_{\geq 0}$. 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 $\sum_{i=1}^n f_i(E_i)$. Dobzinski and Schapira [9] 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 $f\colon 2^E \to \mathbb{R}_{\geq 0}$, and weights $w\colon E \to \mathbb{R}_{\geq 0}$. We present a refined variant of the incremental algorithm introduced in [4]. 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 [4] only scales the capacity, our algorithm $\textsc{Alg}_{\text{scale}}$ simultaneously scales 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

$$0 = \lambda^7 - 2\lambda^6 - 3\lambda^5 - 3\lambda^4 - 3\lambda^3 - 2\lambda^2 - \lambda - 1.$$

This yields

$$\left(\frac{1}{\lambda} + \frac{1}{\lambda^2}\right)\frac{\lambda^3}{\lambda^2+1} = \frac{\lambda^2}{\lambda+1} - 1 - \frac{\lambda^2+1}{\lambda^3}. \tag{2.1}$$

Furthermore, let $\delta := \frac{\lambda^3}{\lambda^2+1} \approx 3.0143$ and

$$\rho := \max\bigl\{\lambda\sqrt{M}, 2M\bigr\}. \tag{2.2}$$

Algorithm $\textsc{Alg}_{\text{scale}}$ operates in phases of increasing capacities $C_1, \dots, C_N \in \mathbb{R}_{\geq 0}$ with

$$\begin{aligned} C_1 &:= \min_{e\in E} w(e), \\ C_i &:= \min\bigl\{C \geq \delta C_{i-1} \bigm| f^*(C) \geq \rho f^*(C_{i-1})\bigr\} \qquad \text{for all } i \in \mathbb{N}, \end{aligned}$$

where we set $\min\emptyset = \sum_{e\in E} w(e)$. Let $N \in \mathbb{N}$ be the minimal index such that $C_N = \sum_{e\in E} w(e)$. In phase $i \in \{1,\dots,N\}$, $\textsc{Alg}_{\text{scale}}$ adds the elements of the set $S^*(C_i)$ one at a time. Recall that $S^*(C)$ is the optimum solution to (P) for capacity C. 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. [11]):

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

and its dual

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

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 ($\mathrm{LP_X}$). 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 ($\mathrm{LP_X}$). By strong duality, there exists an optimum dual solution $\gamma^*(X) \in \mathbb{R}^E$ with

$$f(X) = \sum_{e \in X} \gamma^*(X)_e. \tag{2.3}$$

In phase 1, the algorithm $\textsc{Alg}_{\text{scale}}$ adds the single element in $S^*(C_1)$. In phase 2, $\textsc{Alg}_{\text{scale}}$ adds an element $e \in S^*(C_2)$ first that maximizes $\gamma^*(S^*(C_2))_e$ and the other elements in an arbitrary order. In phase $i \in \{3, 4, \dots, N\}$, $\textsc{Alg}_{\text{scale}}$ adds the elements of $S^*(C_i)$ in an order $(e_1, \dots, e_{|S^*(C_i)|})$ such that, for all $j \in \{1, \dots, |S^*(C_i)| - 1\}$,

$$\frac{\gamma^*(S^*(C_i))_{e_j}}{w(e_j)} \geq \frac{\gamma^*(S^*(C_i))_{e_{j+1}}}{w(e_{j+1})}. \tag{2.4}$$

The reason why we do not use (2.4) in phase 2 is that so early on we want to increase the objective value as fast as possible which is not necessarily guaranteed by choosing the order of elements in $S^*(C_2)$ according to (2.4).

By π^{A}, we refer to the the permutation of E that represents the order in which the algorithm $\textsc{Alg}_{\text{scale}}$ adds the elements of E. Let $0 \leq C \leq C' \leq w(E)$, $k := |S^*(C')|$, and let $(e_1, \dots, e_k)$ be an ordering of all elements in $S^*(C')$ such that (2.4) holds for all $j \in \{1, \dots k-1\}$. With $j := \max\{j \in \{1, \dots k\} \mid w(\{e_1, \dots, e_j\}) \leq C\}$, we let $S^*(C', C) := \{e_1, \dots, e_j\}$ denote the largest prefix of the optimum solution $S^*(C')$ with capacity at most C.

Roughly, we show that this algorithm is competitive as follows: In the first phase $\textsc{Alg}_{\text{scale}}$ obviously performs optimally. In all other phases, the solution added in the previous phase is large enough to be competitive until the solution added currently has a larger value. From this point until the end of the phase, the current solution is competitive.

Remark 2.1. In the construction of our algorithm, we assume to have oracle access to an optimum solution $S^*(C)$ of a given capacity $C \in \mathbb{R}_{\geq 0}$. Finding such a solution may not be possible in polynomial time. Badanidiyuru et al. [3], give a $(2+\varepsilon)$-approximation algorithm and show that no polynomial time algorithm can have an approximation ratio of less than 2, unless $\mathsf{P} = \mathsf{NP}$. Our algorithm $\textsc{Alg}_{\text{scale}}$ can use an α-approximation oracle instead of an oracle for the optimum solution, for a loss of factor α in its competitive ratio. Furthermore, we assume to have access to an XOS oracle. For a given set $X \subseteq E$ and $x \in X$, an XOS oracle gives the value of x within the set X, which corresponds to the solution of the dual LP mentioned above. Instead of an XOS oracle, our algorithm can use an β-approximation oracle for a loss of factor β in its competitive ratio.

For all $X \subseteq E$, the dual variables $\gamma^*(X)$ are a feasible solution for the dual of (LP_X). Thus, for all $Y \subseteq E$, we have

$$\sum_{e \in Y} \gamma^*(X)_e \leq f(Y). \tag{2.5}$$

i.e., $\gamma^*(X)$ associates a contribution to the overall objective to each element $e \in E$, and this association is consistent for all sets $Y \subseteq E$.

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

$$f^*(C') \leq \frac{C'}{C}\big(f(S^*(C', C)) + M\big). \tag{2.6}$$

Proof. If $S^*(C') = S^*(C', C)$, the statement holds trivially. Suppose that we have $|S^*(C')| > |S^*(C', C)|$. Let $j := |S^*(C', C)|$, and let $S^*(C') = \big\{e_1, \ldots, e_{|S^*(C')|}\big\}$ such that (2.4) holds. Note that, by definition, $S^*(C', C) = \{e_1, \ldots, e_j\}$ and

$$w(\{e_1, \ldots, e_j\}) \leq C < w(\{e_1, \ldots, e_{j+1}\}). \tag{2.7}$$

We have

$$
\begin{aligned}
f^*(C') &\overset{(2.3)}{=} \sum_{i=1}^{|S^*(C')|} w(e_i)\frac{\gamma^*(S^*(C'))_{e_i}}{w(e_i)} \\
&\overset{(2.4)}{\leq} \left(\sum_{i=1}^{j+1}\gamma^*(S^*(C'))_{e_i}\right) + \frac{\sum_{i=1}^{j+1} w(e_i)}{w(\{e_1,\dots,e_{j+1}\})}\sum_{i=j+2}^{|S^*(C')|} w(e_i)\frac{\gamma^*(S^*(C'))_{e_{j+1}}}{w(e_{j+1})} \\
&= \left(\sum_{i=1}^{j+1}\gamma^*(S^*(C'))_{e_i}\right) + \frac{\sum_{i=1}^{j+1} w(e_i)\frac{\gamma^*(S^*(C'))_{e_{j+1}}}{w(e_{j+1})}}{w(\{e_1,\dots,e_{j+1}\})}\sum_{i=j+2}^{|S^*(C')|} w(e_i) \\
&\overset{(2.4)}{\leq} \left(\sum_{i=1}^{j+1}\gamma^*(S^*(C'))_{e_i}\right) + \frac{\left(\sum_{i=1}^{j+1}\gamma^*(S^*(C'))_{e_i}\right)}{w(\{e_1,\dots,e_{j+1}\})}\sum_{i=j+2}^{|S^*(C')|} w(e_i) \\
&\overset{(2.7)}{<} \left(\sum_{i=1}^{j+1}\gamma^*(S^*(C'))_{e_i}\right) + \frac{\left(\sum_{i=1}^{j+1}\gamma^*(S^*(C'))_{e_i}\right)}{C}(C'-C) \\
&= \frac{C'}{C}\left[\left(\sum_{i=1}^{j}\gamma^*(S^*(C'))_{e_i}\right) + \gamma^*(S^*(C'))_{e_{j+1}}\right] \\
&\overset{(2.5)}{\leq} \frac{C'}{C}\big(f(\{e_1,\dots,e_j\}) + f(\{e_{j+1}\})\big) \\
&\leq \frac{C'}{C}\big(f(S^*(C',C)) + M\big),
\end{aligned}
$$

completing the proof. ∎

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

$$
f^*(C') \leq \frac{C'}{C}\big(f^*(C) + M\big). \tag{2.8}
$$

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

THEOREM 2.4. *For $\rho = \max\{\lambda\sqrt{M}, 2M\} \approx \max\{3.2924\sqrt{M}, 2M\}$, the incremental solution computed by $\textsc{Alg}_{\text{scale}}$ is ρ-competitive.*

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

$$
f^*(C_i) \geq \rho f^*(C_{i-1}) \geq \rho^{i-1} f^*(C_1) \overset{\text{(MB)}}{\geq} \rho^{i-1} \geq (\lambda\sqrt{M})^{i-1}, \tag{2.9}
$$

where for the first inequality, we use the definition of the algorithm $\textsc{Alg}_{\text{scale}}$, and for the last inequality we use the definition of ρ in (2.2).

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 $S^*(C_1)$, i.e., the optimum solution of size C_1, the value of $\pi^{\mathrm{A}}(C)$ 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

$$f(\pi^{\mathrm{A}}(C)) \geq f(\pi^{\mathrm{A}}(C_1)) = f^*(C_1) > \frac{1}{\rho} f^*(C).$$

Now assume $C_2 \leq \delta C_1$. 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,

$$f(\pi^{\mathrm{A}}(C)) \geq f^*(C_1) \geq 1 \geq \frac{1}{2M} f^*(C) \geq \frac{1}{\rho} f^*(C).$$

Now suppose $C \in [3C_1, C_2)$ and that $S^*(C_2)$ contains at least 3 elements. The prefix $\pi^{\mathrm{A}}(C_1 + C_2)$ contains all elements from $S^*(C_1) \cup S^*(C_2)$, the prefix $\pi^{\mathrm{A}}(C_2) = \pi^{\mathrm{A}}(C_1 + C_2 - C_1)$ contains at least all but one elements of $S^*(C_2)$, and the prefix $\pi^{\mathrm{A}}(C_2 - C_1)$ contains at least all but 2 elements of $S^*(C_2)$ because the weight of any element is at least C_1. Since

$$C \geq 3C_1 > (\delta - 1)C_1 \geq C_2 - C_1,$$

$\pi^{\mathrm{A}}(C)$ contains at least all but 2 elements from $S^*(C_2)$. Recall that in phase 2, the algorithm adds the element $e \in S^*(C_2)$ that maximizes $\gamma^*(S^*(C_2))_e$ first. Therefore, and because $|S^*(C_2)| \geq 3$, we have $f(\pi^{\mathrm{A}}(C)) \geq \frac{1}{3} f(S^*(C_2)) \geq \frac{1}{\rho} f^*(C)$.

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

$$f^*(C_1 + C_2) \leq f^*(C_2) + M \tag{2.10}$$

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

$$f\big(\pi^{\mathrm{A}}(C_2)\big) \geq f^*(C_2) - M \geq \rho - M \geq M \tag{2.11}$$

where the first inequality follows from subadditivity of f and the fact that the prefix $\pi^{\mathrm{A}}(C_2)$ contains at least all but one element from $S^*(C_2)$. We obtain

$$f\big(S^{\mathrm{A}}(C_2)\big) \overset{(2.11)}{\geq} f^*(C_2) - M \overset{(2.10)}{\geq} f^*(C_1 + C_2) - 2M \overset{(2.11)}{\geq} f^*(C_1 + C_2) - 2f(S^{\mathrm{A}}(c_2)),$$

i.e., by monotonicity,

$$f^*(C) \leq f^*(C_1 + C_2) \leq 3f(\pi^{\mathrm{A}}(C_2)) \leq \rho f(\pi^{\mathrm{A}}(C_2)) \leq \rho f(\pi^{\mathrm{A}}(C)).$$

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

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

This yields $\sum_{j=1}^{i-1} C_j \leq C_i \leq \sum_{j=1}^{i} C_j$. If $i = N$ and $\sum_{j=1}^{N-1} C_j \geq C_N = w(E)$, we have nothing left to show because $C \geq w(E)$. Thus, suppose that $\sum_{j=1}^{N-1} C_j < C_N$. Furthermore, if $i = N$ and $\rho f^*(C_{N-1}) > f^*(C_N) = f^*(w(E)) = f(E)$, we again have nothing to show as the prefix $\pi^{\mathrm{A}}\big(\sum_{j=1}^{N-1} C_j\big) \subseteq \pi^{\mathrm{A}}(C)$ contains the set $S^*(C_{N-1})$ and has value at least $f^*(C_{N-1})$. Thus, assume that $f^*(C_N) \geq \rho f^*(C_{N-1})$. This implies that (2.9) also holds for $i = N$.

Case 1: $C \in \big(\sum_{j=1}^{i-1} C_j, C_i\big)$. We will show that in this case, the value of the solution C_{i-1}, which is already added by the algorithm, is large enough to guarantee competitiveness. 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., using (MO), we have

$$f(\pi^{\mathrm{A}}(C)) \geq f\big(\pi^{\mathrm{A}}\big(\textstyle\sum_{j=1}^{i-1} C_j\big)\big) \geq f^*(C_{i-1}) > \tfrac{1}{\rho} f^*(C).$$

Now, consider the case that $C_i \leq \delta C_{i-1}$. Note that $C_i < \delta C_{i-1}$ is only possible if $i = N$. We distinguish between two different cases:

Case 1.1: $i = 3$. Let $c := \big(\frac{1}{\lambda\sqrt{M}} + \frac{1}{\lambda^2}\big)\delta C_2$. We have

$$\begin{aligned}
c &\leq \left(\frac{1}{\lambda} + \frac{1}{\lambda^2}\right)\delta C_2 \\
&\overset{(2.1)}{=} \left(\frac{\lambda^2}{\lambda+1} - 1 - \frac{1}{\delta}\right) C_2 \\
&\leq \frac{\lambda^2}{\frac{\lambda}{\sqrt{M}} + 1} C_2 - C_2 - C_1 \\
&= \frac{\lambda^2 M}{\lambda\sqrt{M} + M} C_2 - C_2 - C_1. \qquad (2.12)
\end{aligned}$$

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

$$\begin{aligned}
f^*(C_1 + C_2 + c) &\overset{(2.8)}{\leq} \frac{C_1 + C_2 + c}{C_2}\big(f^*(C_2) + M\big) \\
&\overset{(2.12)}{\leq} \frac{C_1 + C_2 + \left(\frac{\lambda^2 M}{\lambda\sqrt{M}+M} C_2 - C_2 - C_1\right)}{C_2}\big(f^*(C_2) + M\big) \\
&= \frac{\lambda^2 M}{\lambda\sqrt{M} + M}\left(1 + \frac{M}{f^*(C_2)}\right) f^*(C_2) \\
&\overset{(2.9)}{\leq} \frac{\lambda^2 M}{\lambda\sqrt{M} + M}\left(1 + \frac{M}{\lambda\sqrt{M}}\right) f^*(C_2) \\
&= \lambda\sqrt{M}\frac{\lambda\sqrt{M}}{\lambda\sqrt{M} + M}\left(\frac{\lambda\sqrt{M} + M}{\lambda\sqrt{M}}\right) f^*(C_2) \\
&\leq \rho f^*(C_2) \\
&\leq \rho f(\pi^{\mathrm{A}}(C_1 + C_2)),
\end{aligned}$$

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., $S^*(C_2) \subseteq \pi^{\mathrm{A}}(C_1 + C_2)$.

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

$$\begin{aligned}
f\big(\pi^{\mathrm{A}}(C_1 + C_2 + c)\big) &\overset{\text{(MO)}}{\geq} f(S^*(C_3, c)) \\
&\overset{(2.6)}{\geq} \frac{c}{C_3} f^*(C_3) - M \\
&\geq \left[\left(\frac{1}{\lambda\sqrt{M}} + \frac{1}{\lambda^2}\right) - \frac{M}{f^*(C_3)}\right] f^*(C_3) \\
&\overset{(2.9)}{\geq} \left(\frac{1}{\lambda\sqrt{M}} + \frac{1}{\lambda^2} - \frac{M}{\lambda^2 M}\right) f^*(C_3) \\
&= \frac{1}{\lambda\sqrt{M}} f^*(C_3) \\
&\geq \frac{1}{\rho} f^*(C_3),
\end{aligned}$$

where for the third inequality we use that $C_3 \leq \delta C_2$. This, together with monotonicity of f, implies $f^*(C) \leq \rho f(\pi^{\mathrm{A}}(C))$ for all $C \in (C_1 + C_2, C_3]$.

Case 1.2: $i \geq 4$. : Recall that $C \in \big(\sum_{j=1}^{i-1} C_j, C_i\big)$. We have

$$\begin{aligned}
f^*(C) &\overset{\text{(MO)}}{\leq} f^*(C_i) \\
&\overset{(2.8)}{\leq} \frac{C_i}{C_{i-1}} (f^*(C_{i-1}) + M) \\
&\leq \delta\left(1 + \frac{M}{f^*(C_{i-1})}\right) f^*(C_{i-1}) \\
&\overset{(2.9)}{\leq} \delta\left(1 + \frac{M}{\lambda^2 M}\right) f^*(C_{i-1}) \\
&= \frac{\lambda^3}{\lambda^2 + 1}\left(1 + \frac{1}{\lambda^2}\right) f^*(C_{i-1}) \\
&= \lambda f^*(C_{i-1}) \\
&\leq \rho f(\pi^{\mathrm{A}}(C)),
\end{aligned}$$

where for the third inequality we use that $C_i \leq \delta C_{i-1}$. Thus, also in this case, we find $f^*(C) \leq \rho f(\pi^{\mathrm{A}}(C))$ for all $C \in \big(\sum_{j=1}^{i-1} C_j, C_i\big)$.

Case 2: $C \in \big[C_i, \sum_{j=1}^{i} C_j\big]$. Since $C_N = w(E)$, we can assume that $i < N$. 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., $S^*(C_i, C - \sum_{j=1}^{i-1} C_j) \subseteq \pi^{\mathrm{A}}(C)$. We will show that the value of this set is large enough to guarantee competitiveness in this case. We have

$$\begin{aligned}
f\big(\pi^{\mathrm{A}}(C)\big) &\overset{\text{(MO)}}{\geq} f\Big(S^*\big(C_i, C - \textstyle\sum_{j=1}^{i-1} C_j\big)\Big) \\
&\overset{(2.6)}{\geq} \frac{C - \sum_{j=1}^{i-1} C_j}{C_i} f^*(C_i) - M \\
&\overset{(2.8)}{\geq} \frac{C - \sum_{j=1}^{i-1} C_j}{C_i}\left(\frac{C_i}{C} f^*(C) - M\right) - M \\
&= \left(\frac{C - \sum_{j=1}^{i-1} C_j}{C} - \frac{C - \sum_{j=1}^{i-1} C_j}{C_i} \cdot \frac{M}{f^*(C)} - \frac{M}{f^*(C)}\right) f^*(C)
\end{aligned}$$

$$
\begin{aligned}
&\geq \left(1 - \sum_{j=1}^{i-1} \frac{C_j}{C_i} - 1 \cdot \frac{M}{f^*(C)} - \frac{M}{f^*(C)}\right) f^*(C) \\
&\overset{(2.9)}{\geq} \left(1 - \sum_{j=1}^{i-1} \frac{1}{\delta^{i-j}} - \frac{2M}{\rho^{i-1}}\right) f^*(C) \\
&\geq \left(1 - \sum_{j=1}^{\infty} \frac{1}{\delta^{j}} - \frac{2M}{\rho^{i-1}}\right) f^*(C) \\
&\geq \left[1 - \left(\frac{1}{1-\delta^{-1}} - 1\right) - \frac{2M}{\lambda^2 M}\right] f^*(C) \\
&\geq 0.319 \cdot f^*(C) \\
&\geq \frac{1}{\rho} f^*(C),
\end{aligned}
$$

where for the fourth inequality we use $C_i \leq C \leq \sum_{j=1}^{i} C_j$ and for the fifth inequality we use $C_{j+1} \geq \delta C_j$. ∎

For 1-bounded objectives, Theorem 2.4 immediately yields the following.

Corollary 2.5. *If $M = 1$, the incremental solution computed by* $\textsc{Alg}_{\text{scale}}$ *is 3.2924-competitive.*

3. Lower bound. In this section, we show the second part of Theorem 1.6, i.e., we give a lower bound on the competitive ratio for the incremental optimization problem (P) with monotone, M-bounded, and fractionally subadditive objectives, and we show a lower bound for the special case with $M = 1$.

Theorem 3.1. *For monotone, M-bounded, and fractionally subadditive objectives, the knapsack problem (P) 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 $f(S) := \sum_{e \in S} v(e)$ for all $S \subseteq E$. 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 (P). 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 $\rho \geq M$. ∎

We proceed to give a stronger lower bound for $M \in [1, \varphi+1)$ with $\varphi = \frac{1}{2}(1+\sqrt{5})$. To this end, for a value $n \in \mathbb{N}$, consider an instance I with $\sum_{i=1}^{n} i = \frac{1}{2}n(n+1)$ elements partitioned into sets $E_1, E_2, \ldots, E_n$ such that $|E_i| = i$ for all $i \in \{1, \ldots, n\}$. We set

$$
f(S) = \max_{i \in \{1,\ldots,n\}} |S \cap E_i| \quad \text{for all } S \subseteq E. \tag{3.1}
$$

The elements' weights are defined as $w(e) = b + i!$ for all $e \in E_i$ with base weight $b = (n+2)!$.

We note that the problem instance I is built in such a way that the elements in all sets $E_1, \ldots, E_n$ have roughly the same relative weight because b is very large.

We show first, for a given capacity $C \in \mathbb{N}$, the number of elements that can be packed without exceeding this capacity can vary by at most 1, regardless of which elements are packed. Yet, the weights of elements in E_i increase quickly enough with increasing i such that, for capacity $C = i(b + i!)$ it is only possible to pack i elements if all i elements are from the set $E_1 \cup \dots \cup E_i$.

PROPOSITION 3.2. *Let π be a solution to the instance defined above. Consider capacity $C = i(b + i!)$ for $i \in \{1, \dots, \frac{1}{2}n(n+1)\}$. Then, we have*

$$|\pi(C)| = \begin{cases} i & \text{if } \{e_{\pi(1)}, \dots, e_{\pi(i)}\} \subseteq E_1 \cup \dots \cup E_i, \\ i-1 & \text{else} \end{cases}$$

with $E_j = \emptyset$ for $j > n$.

Proof. First we observe that $|\pi(C)| \in \{i, i-1\}$: Assume $|\pi(C)| < i - 1$. Then, we have

$$\begin{aligned} C - w(\pi(C)) &\geq i(b + i!) - (i-2)(b + n!) \geq 2b - in! \\ &\geq b + (n+2)! - n(n+1)n! = b + 2(n+1)! \geq \max_{e \in E} w(e), \end{aligned}$$

contradicting that $\pi(C)$ is the maximum prefix of π which fits into capacity C. Assume $|\pi(C)| > i$. Then, we have $w(\pi(C)) > (i+1)b > ib + (i+1)! > i(b + i!)$, which contradicts $w(\pi(C)) \leq C$.

If $\pi(C)$ contains $e \in E_j$ with $j > i$, we have

$$\begin{aligned} C - w(e) &= i(b + i!) - (b + j!) = (i-1)b + ii! - j! \\ &< (i-1)b < (i-1) \min_{e \in E} w(e). \end{aligned}$$

So $\pi(C) \setminus e$ contains at most $i - 2$ elements and $|\pi(C)| \leq i - 1$. If the first i elements in π are in the sets $E_1, \dots E_i$, we have $w(\{e_{\pi(1)}, \dots, e_{\pi(i)}\}) \leq i(b+i!) = C$. Therefore, we have $|\pi(C)| \geq i$. ∎

We say an incremental solution to the problem instance I given above *is represented by* a set of numbers $A = \{a_1, \dots, a_\ell\}$ with $a_i < a_{i+1}$ and $a_\ell = n$ if the solution adds first all elements from E_{a_1}, then from E_{a_2}, and so on until adding all elements from E_{a_ℓ}. Only afterwards all remaining elements are added in an arbitrary order. Note, that elements added after the last element of E_n in any solution do not influence the objective value for any capacity, since when they are added the incremental solution has already reached the maximum value of n. First we will observe, that every incremental solution of problem instance I can be transformed into a solution that can be represented by a set $\{a_1, ..., a_\ell\}$ without decreasing the objective value for any capacity.

LEMMA 3.3. *For every incremental solution π there is a set $A = \{a_1, \dots, a_\ell\}$ with $a_i < a_{i+1}$ and $a_\ell = n$ representing an incremental solution with objective value at least $f(\pi(C))$ for all capacities $C \geq 0$.*

Proof. First, we show that there is a solution π', whose objective value is at least $f(\pi(C))$ for every capacity, such that, for all $i \in \{1, ..., n-1\}$, if at least one element from the set E_i is added before the last element from E_n is added, then this is true for all elements from E_i. Furthermore, if $h \in \mathbb{N}$ is the index of the last element from E_i in π', we have $f(\{e_{\pi'(1)}, ..., e_{\pi'(h-1)}\}) + 1 = i = f(\{e_{\pi'(1)}, ..., e_{\pi'(h)}\})$. We will do this by altering the solution π to obtain the desired solution π'.

Fix some $i \in \mathbb{N}$ such that at least one element from the set E_i is added before the last element from E_n is added in the solution π. Let $j \in \mathbb{N}$, $j \leq i$ be the largest number such that, when the j-th element from the set E_i is added, the value of the solution increases from $j-1$ to j. If this does not exist, we set $j = 0$. If $j = i$, E_i is completely added before the last element from E_n and when the last element from E_i is added to the incremental solution its value increases by one to the value of i. Thus, suppose that $j < i$. All elements from E_i that are added after the j-th element do not increase the value of the solution and can thus be moved to the end of the whole order π. Since now, there are only j elements from the set E_i added before the last element from the set E_n is added, it makes sense to add the elements from the set E_j instead of these j elements, as they have a smaller weight (if they are not already added). We can then move the j elements from E_i to the end of the order. After all these changes the solution obtains all values at least as fast as before. By doing this for all $i \in \{1, ..., n-1\}$, we obtain the desired solution π'.

Now, we will show that we can reorder the elements in the solution π' such that the elements that are added before the last element from E_n is added are ordered by the index of the set they belong to. Consider any two sets E_i, E_j, $i < j$ that are added before the last element from set E_n is added. Recall that, when the last element from E_i is added, the value of the solution is i. This implies that at that point at most $i-1$ elements from the set E_j are added. Thus, swapping the elements of E_i and E_j in the order π' until all elements from E_i are added before the elements from E_j, does not decrease the value of the solution for any capacity. By doing this for all pairs (i, j), we obtain a solution that can be represented by a set $A = \{a_1, ..., a_\ell\}$.□

Utilizing the properties of the weights we mentioned before, we can find a collection of conditions which are necessary and sufficient for a set of numbers $\{a_1, \dots, a_\ell\}$ to represent a ρ-competitive solution for the problem instance I. In the following, we denote by ℓ' the index with $\rho a_{\ell'} < n$ and $\rho a_{\ell'+1} \geq n$ and set $z_i := \sum_{j=1}^{i} a_j$. The index ℓ' is needed because all indices $i > \ell'$ satisfy $\rho a_i \geq n$, i.e., after a solution has added the set E_i, it is ρ-competitive for all capacities.

LEMMA 3.4. *Let $A = \{a_1, \dots, a_\ell\}$ with $a_i < a_{i+1}$ and $a_\ell = n$ be a set of numbers that represents an incremental solution for instance I. Then, the incremental solution is ρ-competitive if and only if the following three conditions are satisfied:*

(i) $a_1 = 1$,

(ii) the following two conditions hold for all $i \in \{1, \dots, \ell'\}$:

  (iia) $z_i + a_i \leq \lfloor \rho a_i \rfloor$ *if* $a_{i+1} \leq \lfloor \rho a_i \rfloor + 1$,

  (iib) $z_i + a_i + 1 \leq \lfloor \rho a_i \rfloor$ *if* $a_{i+1} > \lfloor \rho a_i \rfloor + 1$.

Proof. We first show that for a ρ-competitive incremental solution represented by a set of numbers A conditions *(i)*, *(iia)* and *(iib)* have to be satisfied.

If $a_1 \neq 1$, the incremental solution is not competitive for capacity $C = b + 1$, i.e., *(i)* must hold.

Consider capacity $C = (\lfloor \rho a_i \rfloor + 1)(b + (\lfloor \rho a_i \rfloor + 1)!)$ for $i \in \{1, \dots, l'\}$. The optimum solution of capacity C is $S^*(C) = E_{\lfloor \rho a_i \rfloor + 1}$ and has value $f^*(C) = \lfloor \rho a_i \rfloor + 1$. The value of the incremental solution is at least $a_i + 1$, since $\frac{1}{\rho}(\lfloor \rho a_i \rfloor + 1) > a_i$. Thus, the incremental solution of capacity C contains at least $a_i + 1$ elements from $E_{a_{i+1}}$.

If $a_{i+1} \leq \lfloor \rho a_i \rfloor + 1$, the incremental solution contains $\lfloor \rho a_i \rfloor + 1 - z_i$ elements from $E_{a_{i+1}}$ by Proposition 3.2 at capacity C. Thus, we have $a_i + 1 \leq \lfloor \rho a_i \rfloor + 1 - z_i$ which implies *(iia)*. If $a_{i+1} > \lfloor \rho a_i \rfloor + 1$, the incremental solution contains $\lfloor \rho a_i \rfloor - z_i$ elements from $E_{a_{i+1}}$ by Proposition 3.2. Thus, we have $a_i + 1 \leq \lfloor \rho a_i \rfloor - z_i$ which implies *(iib)*.

We proceed to show that, conversely, an incremental solution represented by a set of numbers A satisfying conditions *(i)*, *(iia)* and *(iib)* is ρ-competitive. To this end, fix an arbitrary incremental solution with these properties. Since all elements have integer weight, it suffices to show ρ-competitiveness for all capacities $C \in \mathbb{N}$.

For capacities $C \in \{1, \dots, b+1\}$, the incremental solution is ρ-competitive because $b+1$ is the smallest weight of all elements and $a_1 = 1$ by *(i)*, i.e., the element of smallest weight is added first.

Let $i \in \{1, \dots, \ell'\}$. We will show that, for all capacities in

$$\big\{z_i(b + z_i!) + 1, \dots, z_{i+1}(b + z_{i+1}!)\big\},$$

the incremental solution is ρ-competitive. For all capacities

$$C \in \big\{z_i(b + z_i!) + 1, \dots, (\lfloor \rho a_i \rfloor + 1)(b + (\lfloor \rho a_i \rfloor + 1)!) - 1\big\}, \tag{3.2}$$

the value of the optimum solution is at most $\lfloor \rho a_i \rfloor$ while the value of the incremental solution is at least a_i because it contains at least all elements from $E_{a_1}, \dots, E_{a_i}$ at capacity $z_i(b+z_i!)$ by Proposition 3.2. Thus, the incremental solution is ρ-competitive for all values C as in (3.2). Next, suppose that

$$C \in \big\{(\lfloor \rho a_i \rfloor + 1)(b + (\lfloor \rho a_i \rfloor + 1)!), \dots, z_{i+1}(b + z_{i+1}!)\big\}.$$

Let $a^* \in \{\lfloor \rho a_i \rfloor + 1, \dots, z_{i+1}\}$ be the value of the optimum solution of capacity C. This implies that $C \geq a^*(b + a^*!)$. We consider two cases.

Case 1: $a_{i+1} \leq \lfloor \rho a_i \rfloor + 1$. By *(iia)*, we have $z_i + a_i \leq \lfloor \rho a_i \rfloor$ and thus

$$a^* - z_i \geq a_i + a^* - \lfloor \rho a_i \rfloor \geq \frac{1}{\rho}\lfloor \rho a_i \rfloor + \frac{1}{\rho}\big(a^* - \lfloor \rho a_i \rfloor\big) = \frac{1}{\rho}a^*. \tag{3.3}$$

By Proposition 3.2 the incremental solution contains at least $a^* - z_i$ elements from the set $E_{a_{i+1}}$ at capacity C since $a_{i+1} \leq \lfloor \rho a_i \rfloor + 1 \leq a^*$. This means that, by (3.3), the incremental solution of capacity C has value at least $a^* - z_i \geq \frac{1}{\rho}a^*$, i.e., the solution is ρ-competitive for capacity C.

Case 2: $a_{i+1} > \lfloor \rho a_i \rfloor + 1$. By *(iib)*, we have $z_i + a_i + 1 \leq \lfloor \rho a_i \rfloor$ and thus

$$a^* - z_i - 1 \geq a_i + a^* - \lfloor \rho a_i \rfloor \geq \frac{1}{\rho}\lfloor \rho a_i \rfloor + \frac{1}{\rho}\big(a^* - \lfloor \rho a_i \rfloor\big) = \frac{1}{\rho}a^*. \tag{3.4}$$

By Proposition 3.2 the incremental solution contains at least $a^* - z_i - 1$ elements from the set $E_{a_{i+1}}$ at capacity C. This means that, by (3.4), the incremental solution of capacity C has value at least $a^* - z_i - 1 \geq \frac{1}{\rho}a^*$, i.e., the solution is ρ-competitive for capacity C.

We conclude that for every capacity $C \in \big\{1, \dots, z_{\ell'+1}(b + z_{\ell'+1}!)\big\}$ the incremental solution is ρ-competitive. For all capacities $C > z_{\ell'+1}(b + z_{\ell'+1}!)$, the value of the incremental solution is at least $a_{\ell'+1}$, while the optimum solution has value at most n. By definition of ℓ' we have $\rho a_{\ell'+1} \geq n$. Therefore, the incremental solution is ρ-competitive. ∎

In the following we will show that, for $2 \leq \rho \leq \varphi + 1$ and given some set of numbers $\{a_1, \dots, a_i\}$, every algorithm is forced to choose $a_{i+1} \leq \lfloor \rho a_i \rfloor + 1$ to be ρ-competitive for capacity $\lfloor \rho a_i \rfloor + 1$.

If $z_i + a_i = \lfloor \rho a_i \rfloor$, then $z_i + a_i + 1 > \lfloor \rho a_i \rfloor$, i.e., contraposition of condition *(iib)* from Lemma 3.4 yields the following.

COROLLARY 3.5. *If a set of numbers $\{a_1, \ldots, a_\ell\}$ represents a ρ-competitive incremental solution and $z_i + a_i = \lfloor \rho a_i \rfloor$ for some $i \in \{1, \ldots, \ell'\}$, then*

$$a_{i+1} \leq \lfloor \rho a_i \rfloor + 1. \tag{3.5}$$

PROPOSITION 3.6. *Let $\rho \in [2, \varphi + 1]$, and let $A = \{a_1, \ldots, a_\ell\}$ with $a_i < a_{i+1}$ be a set of numbers that represents an incremental solution. If the incremental solution is ρ-competitive, then $a_{i+1} \leq \lfloor \rho a_i \rfloor + 1$ for all $i \in \{1, \ldots, \ell'\}$.*

Proof. By Corollary 3.5, it suffices to show that we have $z_i + a_i = \lfloor \rho a_i \rfloor$ for all $i \in \{1, \ldots, \ell'\}$. We will prove this by induction. For $i = 1$, we have

$$z_1 + a_1 = 1 + 1 = 2 = \lfloor \rho \rfloor = \lfloor \rho a_1 \rfloor,$$

where we use the fact that $a_1 = 1$ by Lemma 3.4*(i)*.

Suppose that

$$z_i + a_i = \lfloor \rho a_i \rfloor \tag{3.6}$$

holds for some $i \in \{1, \ldots, \ell' - 1\}$. By Lemma 3.4*(iia)* and *(iib)* and the ρ-competitiveness of the incremental solution, we have $z_{i+1} + a_{i+1} \leq \lfloor \rho a_{i+1} \rfloor$. Thus, we only have to show that

$$z_{i+1} + a_{i+1} \geq \lfloor \rho a_{i+1} \rfloor. \tag{3.7}$$

To prove this, we first calculate for $\rho \in (2, \varphi + 1]$:

$$\begin{aligned} \frac{(3-\rho)(\rho-1)}{\rho-2} = \frac{-(\rho-2)^2+1}{\rho-2} &= \frac{1}{\rho-2} - (\rho-2) \\ &\geq \frac{1}{\varphi-1} - (\varphi-1) = \varphi - (\varphi-1) = 1, \end{aligned} \tag{3.8}$$

where for the inequality we use that $\rho \leq \varphi + 1$. In the case $\rho = 2$ we can calculate directly $\rho - 2 = 0 \leq 1 = (3-\rho)(\rho-1)$. We then obtain

$$\begin{aligned} (3-\rho)\lfloor(\rho-1)a_i\rfloor + 1 &> (3-\rho)((\rho-1)a_i - 1) + 1 \\ &= (3-\rho)(\rho-1)a_i + \rho - 2 \\ &\overset{(3.8)}{\geq} (\rho-2)a_i + \rho - 2 \\ &= (\rho-2)(a_i+1). \end{aligned} \tag{3.9}$$

Utilizing this inequality, we have

$$\begin{aligned} \lfloor(\rho-2)(\lfloor\rho a_i\rfloor+1)\rfloor &= \lfloor(\rho-2)(\lfloor(\rho-1)a_i\rfloor + a_i + 1)\rfloor \\ &= \lfloor\lfloor(\rho-1)a_i\rfloor + (\rho-3)\lfloor(\rho-1)a_i\rfloor + (\rho-2)(a_i+1)\rfloor \\ &= \lfloor(\rho-1)a_i\rfloor + \lfloor(\rho-3)\lfloor(\rho-1)a_i\rfloor + (\rho-2)(a_i+1)\rfloor \\ &\overset{(3.9)}{<} \lfloor(\rho-1)a_i\rfloor + \lfloor 1 \rfloor, \end{aligned}$$

where for the third equation we use that $\lfloor(\rho-1)a_i\rfloor \in \mathbb{N}$. Because both sides of this inequality are in $\mathbb{N}$, we have

$$\lfloor(\rho-2)(\lfloor\rho a_i\rfloor+1)\rfloor \leq \lfloor(\rho-1)a_i\rfloor. \tag{3.10}$$

This yields

$$\begin{aligned}
\lfloor \rho a_{i+1} \rfloor &= \lfloor (\rho-2)a_{i+1} \rfloor + 2a_{i+1} \\
&\overset{(3.5)}{\leq} \lfloor (\rho-2)(\lfloor \rho a_i \rfloor + 1) \rfloor + 2a_{i+1} \\
&\overset{(3.10)}{\leq} \lfloor (\rho-1)a_i \rfloor + 2a_{i+1} \\
&= \lfloor \rho a_i \rfloor - a_i + 2a_{i+1} \\
&\overset{(3.6)}{=} z_i + 2a_{i+1} \\
&= z_{i+1} + a_{i+1},
\end{aligned}$$

i.e., (3.7) holds. By induction $z_i + a_i = \lfloor \rho a_i \rfloor$ follows for all $i \in \{1, \dots, \ell'\}$, and therefore, by Corollary 3.5, the proposition holds. ∎

THEOREM 3.7. *For $\rho < \varphi + 1$, there is no ρ-competitive algorithm for problem instance I with sufficiently large $n \in \mathbb{N}$.*

Proof. Suppose, for $\rho < 2$, there was a ρ-competitive incremental solution represented by the set of numbers $\{a_1, \dots, a_\ell\}$. Without loss of generality we can assume that $a_i < a_{i+1}$ for all $i \in \{1, \dots, \ell - 1\}$. Yet, Lemma 3.4*(i)*, *(iia)* and *(iib)* imply that

$$2 = z_1 + a_1 \leq \lfloor \rho a_1 \rfloor = 1$$

which is a contradiction, i.e., for $\rho < 2$, there is no ρ-competitive incremental solution.

Next, suppose that for $\rho \in [2, \varphi + 1)$ there was a ρ-competitive incremental solution. Let the number of disjoint sets $n \in \mathbb{N}$ in the instance be sufficiently large, and let $\{a_1, \dots, a_\ell\}$ be the set of numbers representing a ρ-competitive incremental solution. Without loss of generality, we can assume that $a_{i+1} > a_i$ for all $i \in \{1, \dots, \ell - 1\}$. By Lemma 3.4 and Proposition 3.6, we know that the following conditions are satisfied:

(i) $a_1 = 1$,
(ii) $z_i + a_i \leq \lfloor \rho a_i \rfloor$ for all $i \in \{1, \dots, \ell'\}$,
(iii) $a_{i+1} \leq \lfloor \rho a_i \rfloor + 1$ for all $i \in \{1, \dots, \ell'\}$.

For $1 \leq j \leq i \leq \ell'$, from *(iii)* it follows that

$$a_j \geq \frac{1}{\rho}\lfloor \rho a_j \rfloor \overset{(iii)}{\geq} \frac{1}{\rho}(a_{j+1} - 1) \geq \frac{1}{\rho}\left[\frac{1}{\rho}\left(a_{j+2} - 1\right) - 1\right] \geq \cdots \geq \frac{1}{\rho^{i-j}} a_i - \sum_{k=1}^{i-j} \frac{1}{\rho^k}.$$

This implies

$$\begin{aligned}
z_i &= \sum_{j=1}^{i} a_j \\
&\geq \sum_{j=1}^{i} \left(\frac{1}{\rho^{i-j}} a_i - \sum_{k=1}^{i-j} \frac{1}{\rho^k} \right) \\
&= \left(\sum_{j=0}^{i-1} \frac{1}{\rho^j} \right) a_i - \sum_{j=1}^{i} \sum_{k=1}^{i-j} \frac{1}{\rho^k} \\
&= \frac{1-\rho^{-i}}{1-\rho^{-1}} a_i - \sum_{j=1}^{i} \left(\frac{1-\rho^{j-i-1}}{1-\rho^{-1}} - 1 \right) \\
&\geq \frac{1-\rho^{-i}}{1-\rho^{-1}} a_i - i \frac{1}{1-\rho^{-1}} \\
&= \frac{1}{1-\rho^{-1}} \big((1-\rho^{-i}) a_i - i \big). \qquad (3.11)
\end{aligned}$$

For $i \in \{2, \dots, \ell'\}$, we obtain

$$\begin{aligned}
\rho &\geq \frac{1}{a_i} \lfloor \rho a_i \rfloor \\
&\overset{(ii)}{\geq} \frac{1}{a_i} (z_i + a_i) \\
&\overset{(3.11)}{\geq} \frac{1}{a_i} \cdot \frac{1}{1-\rho^{-1}} \big((1-\rho^{-i}) a_i - i \big) + 1 \\
&= \frac{1}{1-\rho^{-1}} \left(1 - \rho^{-i} - \frac{i}{a_i} \right) + 1. \qquad (3.12)
\end{aligned}$$

Observe that $a_{j+1} > a_j$ for all $j \in \{1, ..., \ell - 1\}$ implies $a_j \geq j$ for all $j \in \{1, ..., \ell\}$. It follows that

$$\begin{aligned}
a_i &\geq \frac{1}{\rho - 1} \big(\lfloor \rho a_i \rfloor - a_i \big) \\
&\overset{(ii)}{\geq} \frac{1}{\rho - 1} z_i \\
&\overset{a_j \geq j}{\geq} \frac{1}{\rho - 1} \cdot \frac{i(i+1)}{2},
\end{aligned}$$

which implies that

$$\frac{i}{a_i} \leq \frac{2(\rho - 1)}{i + 1}. \qquad (3.13)$$

By definition of ℓ' and by Proposition 3.6, ℓ' increases when n is increased sufficiently. Thus, for every $\varepsilon > 0$, there exists some $n \in \mathbb{N}$ such that

$$\frac{\ell'}{a_{\ell'}} \overset{(3.13)}{\leq} \frac{2(\rho - 1)}{\ell' + 1} \leq \frac{\varepsilon}{2} \qquad (3.14)$$

and

$$\rho^{-\ell'} \leq \frac{\varepsilon}{2}. \tag{3.15}$$

Since we chose n sufficiently large, we can assume that this holds. By choosing $\varepsilon = 1 - \frac{\rho-1}{\rho}\varphi$ we see that $\varepsilon > 0$ since $\rho < \varphi + 1$, and we obtain

$$\begin{aligned}
\rho &\overset{(3.12)}{\geq} \frac{1}{1-\rho^{-1}}\left(1 - \rho^{-\ell'} - \frac{\ell'}{a_{\ell'}}\right) + 1 \\
&\geq \frac{1}{1-\rho^{-1}}(1-\varepsilon) + 1 \\
&= \frac{\rho}{\rho-1}\big(1 - \big(1 - \frac{\rho-1}{\rho}\varphi\big)\big) + 1 \\
&= \varphi + 1,
\end{aligned}$$

where the second inequality uses (3.14) and (3.15). This yields a contradiction to the fact that $\rho < \varphi + 1$. Thus, there is no ρ-competitive algorithm for $\rho < \varphi + 1$. ∎

This result immediately yields the desired lower bound.

COROLLARY 3.8. *For monotone,* 1*-bounded, and fractionally subadditive objectives, the knapsack problem (P) does not admit a ρ-competitive incremental solution for $\rho < \varphi + 1$.*

4. Application to flows. In this section we show that our algorithm $\textsc{Alg}_{\text{scale}}$ can be used to solve problems as given in Example 1.5. Formally, for the *incremental maximal potential-based flow problem* on parallel edges, 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. To this end, we are given a continuous and strictly increasing potential-loss function $\psi : \mathbb{R} \to \mathbb{R}$ with $\lim_{x\to\infty} \psi(x) = \infty$. Every edge e has an edge resistance $\beta(e) \in \mathbb{R}_{>0}$ and a capacity $u(e)$. Vertex potentials $p_s, p_t \in \mathbb{R}$ induce a flow of $x_e = \psi^{-1}(p/\beta(e))$ on edge e where $p = p_t - p_s$. This flow is only feasible if $x_e \leq u(e)$. The goal is to choose vertex potentials $p_s, p_t \in \mathbb{R}$ together with a subset of active edges that maximizes the total induced flow. This yields the objective

$$f(S) = \max\left\{ \sum_{e\in T} \psi^{-1}\left(\frac{p}{\beta(e)}\right) \,\middle|\, T \subseteq S, p \in \mathbb{R}_{\geq 0} \text{ with } \psi^{-1}\left(\frac{p}{\beta(e)}\right) \leq u(e) \text{ for all } e \in T \right\} \tag{4.1}$$

for all $S \subseteq E$. The function f is obviously monotone. We further obtain that f scaled by $(\min_{e\in E} u(e))^{-1}$ is M-bounded for $M := \frac{\max_{e\in E} u(e)}{\min_{e\in E} u(e)}$ because $f(\{e\}) = u(e)$. We proceed to show that the objective is fractionally subadditive.

PROPOSITION 4.1. *The function $f : 2^E \to \mathbb{R}_{\geq 0}$ defined in (4.1) is fractionally subadditive.*

Proof. For $e \in E$, let

$$p_e := \beta(e)\psi(u(e))$$

be the maximum potential difference between s and t such that the flow along e induced by the potential difference p_e is still feasible, i.e., does not violate the capacity constraint $u(e)$. For $e, e' \in E$, we define $x_e(p_{e'})$ to be the flow value along e induced by a potential difference of $p_{e'}$ between s and t if this flow is feasible and 0 otherwise. For $S \subseteq E$, we have

$$f(S) = \max\left\{\sum_{e\in T} \psi^{-1}\left(\frac{p}{\beta(e)}\right) \,\middle|\, T \subseteq S, p \in \mathbb{R}_{\geq 0} \text{ with } \psi^{-1}\left(\frac{p}{\beta(e)}\right) \leq u(e) \text{ for all } e \in T\right\}$$
$$= \max\left\{\sum_{e\in S} x_e(p_{e'}) \,\middle|\, e' \in E\right\},$$

i.e., f is an XOS-function and thus fractionally subadditive (see Example 1.3). ◻

As a corollary, we obtain the following result.

COROLLARY 4.2. *The best possible $\rho \geq 1$ for which the incremental maximal potential-based flow problem on parallel edges admits a ρ-competitive solution satisfies*

$$\rho \in \left[\max\{\varphi+1, M\}, \max\{3.293\sqrt{M}, 2M\}\right],$$

where $M = \frac{\max_{e\in E} u(e)}{\min_{e\in E} u(e)}$.

It is possible to define a problem instance of the potential-based maximum flow problem on parallel edges which reflects the lower bound construction in Section 3. Thus, the lower bound on the competitive ratio translates also to this special case.

COROLLARY 4.3. *The incremental maximum potential-based flow problem on parallel edges does not admit a ρ-competitive algorithm for $\rho < \varphi + 1$.*

Proof. For $i = 1, \ldots, n$ define E_i to be a set of i parallel edges from s to t with unit capacities. For $e_i \in E_i$, define its resistance to be $\beta(e_i) := \varepsilon^i$ for some $0 < \varepsilon < 1$. Let the potential loss function ψ be continuous and strictly increasing with $\psi(0) = 0$. Let $p_i := \varepsilon^i \psi(1)$ be the potential difference between s and t inducing a flow of 1 on all edges $e \in E_i$. Then, the maximum potential-based flow on a subset $S \subseteq E = \bigcup_{i=1}^n E_i$ is given by

$$f'(S) = \max\left\{\sum_{e\in T} \psi^{-1}\left(\frac{p}{\beta(e)}\right) \,\middle|\, T \subseteq S, p \in \mathbb{R}_{\geq 0} \text{ with } \psi^{-1}\left(\frac{p}{\beta(e)}\right) \leq u(e) \text{ for all } e \in T\right\}$$
$$= \max\left\{\sum_{j=1}^{i} \sum_{e\in E_j\cap S} \psi^{-1}\left(\frac{p_i}{\beta(e_j)}\right) \,\middle|\, i \in \{1, \ldots, n\}\right\}$$
$$= \max_{i\in\{1,\ldots,n\}} |S \cap E_i| + \sum_{j=1}^{i-1} \psi^{-1}(\varepsilon^{i-j}\psi(1))|S \cap E_j|.$$

The weights that represent the construction cost of the edges are defined as in the problem instance above to be $w(e_i) = b + i!$ for $b = (n + 2)!$.

Assume there is a ρ-competitive algorithm with $\rho < \varphi + 1$ for this problem. Let $\varepsilon' > 0$ with $\rho + \varepsilon' < \varphi + 1$. By Theorem 3.7 there is an $n \in \mathbb{N}$ such that the incremental knapsack problem given above does not admit a $(\rho + \varepsilon')$-competitive solution for the objective function f defined in (3.1). Choose ε small enough such that $\rho n^2 \psi^{-1}(\varepsilon\psi(1)) < \varepsilon'$. This implies

$$f'(S) - f(S) = \sum_{j=1}^{i-1} \psi^{-1}(\varepsilon^{i-j}\psi(1))|S \cap E_j| \leq n^2\psi^{-1}(\varepsilon\psi(1)) < \frac{\varepsilon'}{\rho}.$$

Let π be a ρ-competitive solution for f' produced by the algorithm. Then, for $C \geq b+1$ we have

$$(\rho + \varepsilon')f(\pi(C)) > \rho\left(f'(\pi(C)) - \frac{\varepsilon'}{\rho}\right) + \varepsilon' \geq (f')^*(C) \geq f^*(C),$$

where in the first inequality we use $f(\pi(C)) \geq 1$ and $f'(S) - f(S) < \varepsilon'/\rho$, in the second we use ρ-competitiveness of π w.r.t. f' and in the third inequality we use the fact that $f(S) \leq f'(S)$. Thus, π would be a $(\rho+\varepsilon')$-competitive solution for the incremental knapsack problem, contradicting our assumption. Therefore, a ρ-competitive algorithm for the incremental maximum potential-based flow problem cannot exist for $\rho < \varphi + 1$. ∎

We now return to the *incremental maximum flow problem* discussed in Section 1. In this problem, we are given a directed graph $G = (V, E)$ with two designated vertices $s, t \in V$. For $r \in \mathbb{R}_{\geq 0}$, a vector $(x_e)_{e\in E}$ is an s-t-flow of value r if $x_e \leq u(e)$ for all $e \in E$ and

$$\sum_{e\in\delta^+(v)} x_e - \sum_{e\in\delta^-(v)} x_e = \begin{cases} r & \text{if } v = s,\\ -r & \text{if } v = t,\\ 0 & \text{otherwise,}\end{cases}$$

where

$$\delta^+(v) = \{e \in E \mid e = (v, w) \text{ with } w \in V\},$$
$$\delta^-(v) = \{e \in E \mid e = (w, v) \text{ with } w \in V\}$$

denote the set of outgoing edges and the set of ingoing edges of a vertex v, respectively. The incremental maximum flow problem has the objective

$$f(S) = \max\{r \mid \text{there exists an } s\text{-}t\text{-flow of value } r \text{ in } G_S = (V, S)\} \quad \text{for all } S \subseteq E.$$

It is straightforward to verify that f is modular (and, hence, also fractionally subadditive) for the case that G has only the two vertices s and t and all edges go from s to t. We here consider the case for a general graph G. For this case, it is easy to see that the objective need not to be fractionally subadditive in general. In fact, for the example of Figure 1.1a, we have

$$f(\{b\}) = 0, \qquad f(\{c\}) = 0, \qquad f(\{b, c\}) = k.$$

This contradicts fractional subadditivity for the choices $A = \{b, c\}$, $B_1 = \{b\}$, $B_2 = \{c\}$, and $\alpha_1 = \alpha_2 = 1$.

We proceed to show that despite the lack of (fractional) subadditivity, this problem has a competitive solution when $u(e) = 1$ for all $e \in E$. To solve this problem, we describe the algorithm Quickest-Increment that has been introduced by Kalinowski et al. [18] for a different incremental flow problem where the sum of the flow values for all integer capacities C is to be maximized. 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 $i \in \{0, 1, \ldots, r\}$, 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 $k \in \{1, \ldots, |E|\}$, we denote the solution of size k of the algorithm by $S^{\mathrm{A}}(k)$.

With $v_{\max} \in \mathbb{R}_{\geq 0}$ defined as the maximal possible s-t-flow value in the underlying graph, for $j \in \{1, \ldots, \lfloor v_{\max} \rfloor\}$, we denote by c_j the minimum number of edges required to achieve a flow value of at least j. The values λ_i and c_j are related in the following way; see Kalinowski et al. [18] (Lemma 4).

LEMMA 4.4. *When $w(e) = u(e) = 1$ for all $e \in E$, 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 for the unit weight and unit capacity case.

THEOREM 4.5. *For the incremental maximal flow problem with $w(e) = u(e) = 1$ for all $e \in E$, the algorithm QUICKEST-INCREMENT is 2-competitive.*

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

Consider some size $k \in \{1, \ldots, |E|\}$. If $k < c_1$, we have $f\big(S^*(k)\big) = 0$, i.e., every solution is competitive. If $k \geq c_1$, let $j := f\big(S^*(k)\big)$. Note that we have $f\big(S^*(c_j)\big) = j = f\big(S^*(k)\big)$ and therefore $k \geq c_j$. By Lemma 4.4, we have

$$
\begin{aligned}
\sum_{i=0}^{\lceil j/2 \rceil - 1} \lambda_i &\leq \sum_{i=0}^{\lceil j/2 \rceil - 1} \frac{c_j}{j-i} \\
&= c_j \sum_{i=0}^{\lceil j/2 \rceil - 1} \frac{1}{j-i} \\
&\leq c_j \sum_{i=0}^{\lceil j/2 \rceil - 1} \frac{1}{j - \lceil \frac{j}{2} \rceil + 1} \\
&= c_j \left\lceil \frac{j}{2} \right\rceil \frac{1}{\lfloor \frac{j}{2} \rfloor + 1} \leq c_j
\end{aligned} \tag{4.2}
$$

This implies $f(S^{\mathrm{A}}(k)) \geq f(S^{\mathrm{A}}(c_j)) \overset{(4.2)}{\geq} \lceil \frac{j}{2} \rceil \geq \frac{1}{2} j = \frac{1}{2} f(S^*(k))$. ∎

Now, we turn to the case of unit capacities and rational weights. By rescaling the weights, we can assume that, without loss of generality, the weights are integral. To transform an instance with integral weights to one where all edges have unit weight, one can simply replace every edge $e \in E$ by a path of length $w(e)$ where every edge on the new path has unit weight. Then, Theorem 4.5 can be applied and we obtain the following.

COROLLARY 4.6. *For the incremental maximal flow problem with $u(e) = 1$ and $w(e) \in \mathbb{Q}_{\geq 0}$ for all $e \in E$, the algorithm QUICKEST-INCREMENT is 2-competitive.*

If we consider capacities that are in the interval $[1, M]$, one path can bring at most M times as much flow as every other path. Combining this with the fact that the solution of QUICKEST-INCREMENT for the instance with $u(e) = 1$ for all $e \in E$ is 2-competitive yields that adding the edges in the same order is always within a factor of $2M$ of the optimum solution.

COROLLARY 4.7. *For the incremental maximal flow problem with $u(e) \in [1, M]$, $w(e) \in \mathbb{Q}_{\geq 0}$ for all $e \in E$, the solution generated by QUICKEST-INCREMENT, if it only considers capacities $u(e) = 1$, is $2M$-competitive.*

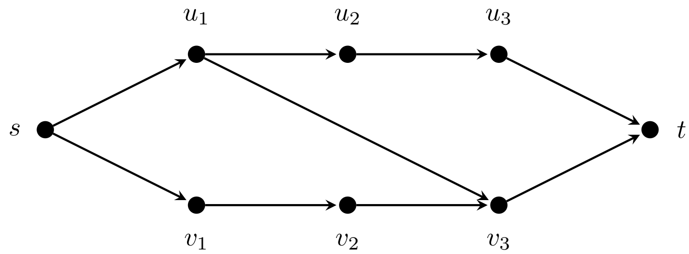


FIGURE 4.1. *A lower bound instance with best possible competitive ratio 2 for the problem* INCREMENTAL MAXIMUM *s-t*-FLOW

As it turns out, the competitiveness of QUICKEST-INCREMENT of 2 in the unit capacity case is optimal.

THEOREM 4.8. *For the incremental maximal flow problem with $u(e) = w(e) = 1$ for all $e \in E$, there is no ρ-competitive algorithm with $\rho < 2$.*

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

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

with unit capacities and unit weights (cf. Figure 4.1). Let π be an arbitrary incremental solution that is ρ-competitive. If the first three elements in π are not the elements (s, u_1), (u_1, v_3), and (v_3, t) (in any order) then the solution is not competitive for $C = 3$. Thus, any competitive solution contains these three elements at the first three positions. This, however, implies that the first eight elements of π cannot contain the elements of the upper and lower paths, i.e., we have $\{(s, u_1), (u_1, u_2), (u_2, u_3), (u_3, t)\} \cup \{(s, v_1), (v_1, v_2), (v_2, v_3), (v_3, t)\}$. This implies that $f(\pi(8)) \leq 1$. Since $f^*(8) = 2$, we obtain $\rho \geq 2$, as claimed. ∎

Furthermore, similar to the incremental maximization of a fractionally subadditive function subject to a knapsack constraint, no algorithm can have a better competitive ratio than M when $u(e) \in [1, M]$ for all $e \in E$.

THEOREM 4.9. *For the incremental maximal flow problem with $u(e) \in [1, M]$ and $w(e) = 1$ for all $e \in E$, there is no ρ-competitive algorithm with $\rho < M$.*

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

$$V := \{s, t, v\},$$
$$E := \{(s, t), (s, v), (v, t)\},$$

with unit weights and capacities $u((s, t)) = 1$, $u((s, v)) = u((v, t)) = M$ (cf. Figure 1.1a).

Let π be an arbitrary incremental solution. If π does not begin with element (s, t), then it is not competitive for $C = 1$. This, however, implies that for any competitive incremental solution π, we have $\pi(2) \neq \{(s, v), (v, t)\}$. Thus, for any competitive π, we have $f(\pi(2)) = 1$ while $f^*(2) = M$. This implies $\rho \geq M$, as claimed. ∎

REFERENCES

[1] G. Amanatidis, G. Birmpas, and E. Markakis, *Coverage, matching, and beyond: New results on budgeted mechanism design*, in Proceedings of the 12th International Conference on Web and Internet Economics (WINE), 2016, pp. 414–428.
[2] N. Anari, N. Haghtalab, S. Naor, S. Pokutta, M. Singh, and A. Torrico, *Structured robust submodular maximization: Offline and online algorithms*, in Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS), 2019, pp. 3128–3137.
[3] A. Badanidiyuru, S. Dobzinski, and S. Oren, *Optimization with demand oracles*, in Proceedings of the 13th ACM Conference on Electronic Commerce (EC), 2012, pp. 110–127.
[4] A. Bernstein, Y. Disser, M. Gross, and S. Himburg, *General bounds for incremental maximization*, Math. Program., 191 (2020), pp. 953–979.
[5] Y. Disser, M. Klimm, N. Megow, and S. Stiller, *Packing a knapsack of unknown capacity*, SIAM J. Discrete Math., 31 (2017), pp. 1477–1497.
[6] Y. Disser, M. Klimm, and D. Weckbecker, *Fractionally subadditive maximization under an incremental knapsack constraint*, in Proceedings of the 19th Workshop on Approximation and Online Algorithms (WAOA), 2021, pp. 206–223.
[7] Y. Disser and D. Weckbecker, *Unified greedy approximability beyond submodular maximization.* arXiv:2011.00962, 2020.
[8] S. Dobzinski, N. Nisan, and M. Schapira, *Approximation algorithms for combinatorial auctions with complement-free bidders*, Mathematics of Operations Research, 35 (2010), pp. 1–13.
[9] S. Dobzinski and M. Schapira, *An improved approximation algorithm for combinatorial auctions with submodular bidders*, in Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA), 2006, pp. 1064–1073.
[10] U. Feige, *A threshold of ln* n *for approximating set cover*, J. ACM, 45 (1998), pp. 634–652.
[11] U. Feige, *On maximizing welfare when utility functions are subadditive*, SIAM J. Comput., 39 (2009), pp. 122–142.
[12] R. Fujita, Y. Kobayashi, and K. Makino, *Robust matchings and matroid intersections*, SIAM J. Discrete Math., 27 (2013), pp. 1234–1256.
[13] M. Gross, M. E. Pfetsch, L. Schewe, M. Schmidt, and M. Skutella, *Algorithmic results for potential-based flows: Easy and hard cases*, Networks, 73 (2019), pp. 306–324.
[14] J. Hartline and A. Sharp, *Incremental flow*, Networks, 50 (2007), pp. 77–85.
[15] R. Hassin and S. Rubinstein, *Robust matchings*, SIAM J. Discrete Math., 15 (2002), pp. 530–537.
[16] N. Kakimura and K. Makino, *Robust independence systems*, SIAM J. Discrete Math., 27 (2013), pp. 1257–1273.
[17] N. Kakimura, K. Makino, and K. Seimi, *Computing knapsack solutions with cardinality robustness*, Jpn. J. Ind. Appl. Math., 29 (2012), pp. 469–483.
[18] T. Kalinowski, D. Matsypura, and M. W. Savelsbergh, *Incremental network design with maximum flows*, Eur. J. Oper. Res., 242 (2015), pp. 51–62.
[19] Y. Kawase, H. Sumita, and T. Fukunaga, *Submodular maximization with uncertain knapsack capacity*, SIAM J. Discrete Math., 33 (2019), pp. 1121–1145.
[20] M. Klimm and M. Knaack, *Maximizing a submodular function with bounded curvature under an unknown knapsack constraint*, in Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), 2022, pp. 49:1–49:19.
[21] Y. Kobayashi and K. Takazawa, *Randomized strategies for cardinality robustness in the knapsack problem*, Theoret. Comput. Sci., 699 (2017), pp. 53–62.
[22] B. Lehmann, D. Lehmann, and N. Nisan, *Combinatorial auctions with decreasing marginal utilities*, Games Econom. Behav., 55 (2006), pp. 270–296.
[23] G. Lin, C. Nagarajan, R. Rajaraman, and D. P. Williamson, *A general approach for incremental approximation and hierarchical clustering*, SIAM J. Comput., 39 (2010), pp. 3633–3669.
[24] A. Marchetti-Spaccamela and C. Vercellis, *Stochastic on-line knapsack problems*, Math. Program., 68 (1995), pp. 73–104.
[25] J. Matuschke, M. Skutella, and J. A. Soto, *Robust randomized matchings*, Math. Oper. Res., 43 (2018), pp. 675–692.
[26] N. Megow and J. 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), 2013, pp. 495–504.
[27] V. Mirrokni, M. Schapira, and J. VondrÃ¡k, *Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions*, 07 2008, pp. 70–77.

[28] A. Navarra and C. M. Pinotti, *Online knapsack of unknown capacity: How to optimize energy consumption in smartphones*, Theoret. Comput. Sci., 697 (2017), pp. 98–109.
[29] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, *An analysis of approximations for maximizing submodular set functions - I*, Math. Program., 14 (1978), pp. 265–294.
[30] N. Nisan, *Bidding and allocation in combinatorial auctions*, in Proceedings of the 2nd ACM Conference on Electronic Commerce (EC), 2000, pp. 1–12.
[31] J. B. Orlin, A. S. Schulz, and R. Udwani, *Robust monotone submodular function maximization*, Math. Program., 172 (2018), pp. 505–537.
[32] Y. Singer, *Budget feasible mechanisms*, in 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, 2010, pp. 765–774.
[33] M. Sviridenko, *A note on maximizing a submodular set function subject to a knapsack constraint*, Oper. Res. Lett., 32 (2004), pp. 41–43.
[34] C. Thielen, M. Tiedemann, and S. Westphal, *The online knapsack problem with incremental capacity*, Math. Methods Oper. Res., 83 (2016), pp. 207–242.
[35] Y. Yoshida, *Maximizing a monotone submodular function with a bounded curvature under a knapsack constraint*, SIAM J. Discrete Math., 33 (2019), pp. 1452–1471.