Mathematical Programming (2022) 191:953–979
https://doi.org/10.1007/s10107-020-01576-0

FULL LENGTH PAPER

Series A

General bounds for incremental maximization

Aaron Bernstein[1] · Yann Disser[2] · Martin Groß[3] · Sandra Himburg[2]

Received: 16 December 2019 / Accepted: 6 October 2020 / Published online: 20 October 2020
© The Author(s) 2020

Abstract

We propose a theoretical framework to capture incremental solutions to cardinality constrained maximization problems. The defining characteristic of our framework is that the cardinality/support of the solution is bounded by a value $k \in \mathbb{N}$ that grows over time, and we allow the solution to be extended one element at a time. We investigate the best-possible competitive ratio of such an incremental solution, i.e., the worst ratio over all k between the incremental solution after k steps and an optimum solution of cardinality k. We consider a large class of problems that contains many important cardinality constrained maximization problems like maximum matching, knapsack, and packing/covering problems. We provide a general 2.618-competitive incremental algorithm for this class of problems, and we show that no algorithm can have competitive ratio below 2.18 in general. In the second part of the paper, we focus on the inherently incremental greedy algorithm that increases the objective value as much as possible in each step. This algorithm is known to be 1.58-competitive for submodular objective functions, but it has unbounded competitive ratio for the class of incremental problems mentioned above. We define a relaxed submodularity condition for the objective function, capturing problems like maximum (weighted) d-dimensional matching, maximum (weighted) (b-)matching and a variant of the maximum flow problem. We show a general bound for the competitive ratio of the greedy algorithm on the class of problems that satisfy this relaxed submodularity condition. Our bound generalizes the (tight) bound of 1.58 slightly beyond sub-modular functions and yields a tight bound of 2.313 for maximum (weighted) (b-)matching. Our bound is also tight for a more general class of functions as the relevant parameter goes to infinity. Note that our upper bounds on the competitive ratios translate to approximation ratios for the underlying cardinality constrained problems, and our bounds for the greedy algorithm carry over both.

Some results of this paper appeared in preliminary form in [1].

Supported by the German Research Foundation (DFG) within Project A07 of CRC TRR 154 and Project 413095939.

Extended author information available on the last page of the article

Keywords Incremental optimization · Maximization problems · Greedy algorithm · Competitive analysis · Cardinality constraint

Mathematics Subject Classification 68W27 · 68W25 · 90C27 · 68Q25

1 Introduction

Practical solutions to optimization problems are often inherently *incremental* in the sense that they evolve historically instead of being established in a one-shot fashion. This is especially true when solutions are expensive and need time and repeated investments to be implemented, for example when optimizing the layout of logistics and other infrastructures. In this paper, we propose a theoretical framework to capture *incremental maximization* problems in some generality.

We describe an incremental problem by a set U containing the possible elements of a solution, and an objective function $f\colon 2^U \to \mathbb{R}_{\geq 0}$ that assigns to each solution $S \subseteq U$ some non-negative value $f(S)$. We consider problems of the form

$$\begin{aligned} \max\ & f(S) \\ \text{s.t. } & |S| \leq k \\ & S \subseteq U, \end{aligned} \tag{1}$$

where $k \in \mathbb{N}$ grows over time.

An incremental solution $\mathbf{S}$ is given by an order $(s_1, s_2, \dots)$ in which the elements of U are to be added to the solution over time. A good incremental solution needs to provide a good solution after k steps, for every k, compared to an optimum solution $S_k^\star$ with k elements, where we let $S_k^\star \in \arg\max_{S \subseteq U, |S|=k} f(S)$ and $f_k^\star := f(S_k^\star)$. Formally, we measure the quality of an incremental solution by its *competitive ratio*. We let $S_k := \{s_1, \dots, s_k\} \subseteq U$ denote the set of the first k elements of $\mathbf{S}$, and we say that $\mathbf{S}$ is (strictly) ρ-competitive if

$$\max_{k \in \{1,\dots,|U|\}} \frac{f_k^\star}{f(S_k)} \leq \rho.$$

An algorithm is called ρ-competitive if it always produces a ρ-competitive solution, and its *competitive ratio* is the infimum over all $\rho \geq 1$ such that it is ρ-competitive. Notice that we do not require the algorithm to run in polynomial time.

While all cardinality constrained optimization problems can be viewed in an incremental setting, clearly not all such problems admit good incremental solutions. For example, consider a cardinality constrained formulation of the classical maximum s-t-flow problem: For a given graph $G = (V, E)$, two vertices $s, t \in V$ and capacities $u\colon E \to \mathbb{R}_{\geq 0}$, we ask for a subset $E' \subseteq E$ of cardinality $k \in \mathbb{N}$ such that the maximum flow in the subgraph (V, E') is maximized. The example in Fig. 1 shows that we cannot hope for an incremental solution that is simultaneously close to optimal for cardinalities 1 and 2.

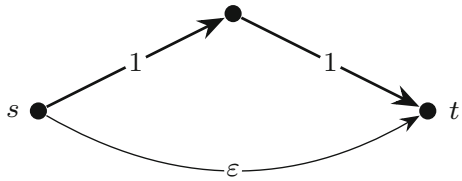

Fig. 1 Example showing that the s–t-flow problem does not always admit good incremental solutions, where $\varepsilon > 0$ is arbitrarily small

In order to derive general bounds on the competitive ratio of incremental problems, we need to restrict the class of objective functions f that we consider. Intuitively, the unbounded competitive ratio in the flow example comes from the fact that we have to invest in the s-t-path of capacity 1 as soon as possible, but this path only yields its payoff once it is completed after two steps.

In order to prevent this and similar behaviors, we can, for example, require f to be monotone (i.e., $f(S) \leq f(T)$ if $S \subseteq T$) and sub-additive (i.e., $f(S) + f(T) \geq f(S \cup T)$). Many important optimization problems satisfy these weak conditions, and we give a short list of examples below. We will see that all these (and many more) problems admit incremental solutions with a bounded competitive ratio. More specifically, we develop a general 2.618-competitive incremental algorithm that can be applied to a broad class of problems, including all problems mentioned below. We illustrate in detail how to apply our model to obtain an incremental variant of the matching problem, and then list incremental versions of other important problems that are obtained analogously.

- Maximum Weighted Matching: Consider a graph $G = (V, E)$ with edge weights $w\colon E \to \mathbb{R}_{\geq 0}$. If we think of edges as *potential* connections and edge weights as *potential* payoffs, then it is not enough to find the final matching, because we cannot construct the edges all at once: the goal is to find a sequence of edges that achieves a high pay-off in the short, the medium, and the long term. In terms of our formal framework, we add edges to a set S one at a time with $U = E$ and $f(S)$ is the maximum weight of a matching $M \subseteq S$. In order to be ρ-competitive, we need that, after k steps for every k, our solution S of cardinality k is no worse than a factor of ρ away from the optimum solution of cardinality k, i.e., $f(S) \geq f(S_k^\star)/\rho$. This model captures the setting where the infrastructure (e.g. the matching, the knapsack, the covering, or the flow) must be built up over time. The online model would be too restrictive in this setting because here we know our options in advance. Note that, as we add more edges, the set of edges S only needs to contain a large matching M, but does not have to be a matching itself. The matching M can change to an arbitrary subset of S from one cardinality to the next and does not have to stay consistent. This ensures that $f(S)$ is monotonically increasing, and is in keeping with the infrastructures setting where the potential regret present in the online model does not apply: building more infrastructure can only help, since once it is built, we can change how it is used. Accordingly, in all the problems below, the set S does not have to be a valid solution to the cardinality constrained problem at hand, but rather needs to *contain* a good solution as a subset. The objective $f(S)$ is consistently defined to be the value of the best solution that

is a subset of S. Notice that this approach can easily be generalized to MAXIMUM b-MATCHING.

- MAXIMUM WEIGHTED d-DIMENSIONAL MATCHING: Given sets $V_1, \ldots, V_d$, vectors $E \subseteq V_1 \times \ldots V_d$, and weights $w\colon E \to \mathbb{R}_{\geq 0}$, we ask for an incremental subset $S \subseteq E$ where $f(S)$ is the maximum weight of a d-dimensional matching in S, i.e., the maximum weight of a subset $M \subseteq S$ with $v_i \neq v_i'$ for all $e = (v_1, \ldots, v_d) \in M$, $e' = (v_1', \ldots, v_d') \in M \backslash \{e\}$, and all $i \in \{1, \ldots, d\}$.
- SET PACKING: Given a set of weighted sets $\mathcal{X}$ we ask for an incremental subset $S \subseteq \mathcal{X}$ where $f(S)$ is the maximum weight of mutually disjoint subsets in S. This problem captures many well-known problems such as MAXIMUM HYPERGRAPH MATCHING and MAXIMUM INDEPENDENT SET.
- MAXIMUM COVERAGE: Given a set of weighted sets $\mathcal{X} \subseteq 2^U$ over an universe of elements U, we ask for an incremental subset $S \subseteq \mathcal{X}$, where $f(S)$ is the weight of elements in $\bigcup_{X \in S} X$. This problem captures maximization versions of clustering and location problems. We can include opening costs $c\colon \mathcal{X} \to \mathbb{R}_{\geq 0}$ by letting $f(S)$ be the maximum over all subsets $S' \subseteq S$ of the number (or weight) of the sets in S' minus their opening costs.
- KNAPSACK: Given a set X of items, associated sizes $s\colon X \to \mathbb{R}_{\geq 0}$ and values $v\colon X \to \mathbb{R}_{\geq 0}$, and a knapsack of capacity 1, we ask for an incremental subset $S \subseteq X$, where $f(S)$ is the largest value $\sum_{x \in S'} v(x)$ of any subset $S' \subseteq S$ with $\sum_{x \in S'} s(x) \leq 1$. This problem can be generalized to MULTI-DIMENSIONAL KNAPSACK by letting item sizes be vectors and letting the knapsack have a capacity in every dimension.
- DISJOINT PATHS: Given a graph $G = (V, E)$, a set of pairs $\mathcal{X} \subseteq V^2$ with weights $w\colon \mathcal{X} \to \mathbb{R}_{\geq 0}$, we ask for an incremental subset $S \subseteq \mathcal{X}$, where $f(S)$ is the maximum weight of a subset $S' \subseteq S$, such that G contains mutually disjoint paths between every pair in S'.
- MAXIMUM BRIDGE-FLOW: We argued above that the maximum s-t-flow problem is not amenable to the incremental setting because it does not pay off to build paths partially.

  To overcome this, we consider a natural restriction of the flow problem where most edges are freely available to be used, and only the edges of a directed s-t-cut need to be built incrementally. If there are no edges over the reversely oriented cut, every s-t-path contains exactly one edge that needs to be built, and we never have to invest multiple steps to establish a single path. This problem captures logistical problems where links need to be established between two clusters, such as when bridges need to be built across a river, cables across an ocean, or when warehouses need to be opened in a supplier-warehouse-consumer network. Formally, given a directed graph $G = (V, E)$ with capacities $u\colon E \to \mathbb{R}$, vertices $s, t \in V$, and a directed s-t-cut $C \subseteq E$ induced by the partition (U, W) of V such that the directed cut induced by (W, U) is empty, we ask for an incremental subset $S \subseteq C$ where $f(S)$ is the value of a maximum flow in the subgraph $(V, E \backslash (C \backslash S))$.

It is easy to verify that the objective functions of all the problems mentioned above (and many more) have the following properties for all $S, T \subseteq U$:

1. (monotonicity): $S \subseteq T \Rightarrow f(S) \leq f(T)$,

2. (sub-additivity): $f(S) + f(T) \geq f(S \cup T)$,
3. (accountability): $\exists s \in S\colon f(S \backslash \{s\}) \geq f(S) - f(S)/|S|$.

To our knowledge, the *accountability* property has not been named before. Intuitively, it ensures that at least one element contributes not more than the average to the value of every set $S \subseteq U$. While it is easy to formulate artificial problems that have monotonicity and sub-additivity but no accountability, or vice-versa, we were not able to identify any natural problems of this kind. This justifies that we rely on accountability in our general incremental algorithm. Note that both sub-additivity and accountability are implied by submodularity.

Observe that a ρ-competitive incremental algorithm immediately yields a ρ-approximation algorithm for the underlying cardinality constrained problem, with the caveat that the resulting approximation algorithm might not be efficient, since we make no demands on the runtime of the incremental algorithm. The converse is rarely the case, since approximation algorithms usually do not construct their solution in an incremental fashion. A prominent exception are *greedy* algorithms that are inherently incremental in the sense that they pick elements one-by-one such that each pick increases the objective by the maximum amount possible. This type of a greedy algorithm has been studied as an approximation algorithm for many cardinality constrained problems, and approximation ratios translate immediately to competitive ratios for the incremental version of the corresponding problem. In particular, the greedy algorithm is known to have competitive ratio (exactly) $\frac{e}{e-1} \approx 1.58$ if the objective function f is monotone and submodular [32]. Note, however, that of all the incremental problems listed above, only Maximum Coverage (without opening costs) has a submodular objective function. It is also known that if we relax the submodularity requirement and allow f to be the minimum of two monotone (sub-)modular functions, the greedy algorithm can be arbitrarily bad [25]. We provide a different relaxation of submodularity that captures Maximum (Weighted) d-Dimensional Matching, Maximum (Weighted) (b-)Matching, and Maximum Bridge-Flow, and where the greedy algorithm has a bounded competitive/approximation ratio.

Our Results. As our first result, we show that a large class of incremental problems admits a bounded competitive ratio. We remark that our upper bound does not require sub-additivity, while our lower bound still works for sub-additive objectives.

Theorem 1 *Every incremental problem with monotone, accountable objective admits a* $(1+\varphi)$*-competitive algorithm, where* φ *is the golden ratio and* $(1+\varphi) \approx 2.618$. *No general deterministic algorithm for this class of problems has a competitive ratio of* 2.18 *or better.*

Again, note that we make no guarantees regarding the running time of our incremental algorithm. In fact, our algorithm relies on the ability to compute the optimum of the underlying cardinality constrained problem for increasing cardinalities. If we can provide an efficient approximation of this optimum, we get an efficient incremental algorithm in the following sense.

Corollary 1 *If there is a polynomial time* α*-approximation algorithm for a cardinality constrained problem with monotone, accountable objective, then we can design a polynomial time* $\alpha(1+\varphi)$*-competitive incremental algorithm.*

We also analyze the approximation/competitive ratio of the greedy algorithm. We observe that for many incremental problems like KNAPSACK, MAXIMUM INDEPENDENT SET, and DISJOINT PATHS, the greedy algorithm has an unbounded competitive ratio (Observation 4). On the other hand, we define a relaxation of submodularity called *α-augmentability* (see Definition 2 below) under which the greedy algorithm has a bounded competitive ratio. In particular, this relaxation captures MAXIMUM WEIGHTED α-DIMENSIONAL MATCHING (Proposition 4), and, for $\alpha = 2$, it captures cardinality constrained versions of MAXIMUM (WEIGHTED) (b-)MATCHING (Proposition 5) and MAXIMUM BRIDGE-FLOW (Proposition 6). We get the following result, where the tight lower bound for $\alpha = 2$ is obtained for MAXIMUM BRIDGE-FLOW, and the lower bound of α for $\alpha \in \mathbb{N}$ is obtained for MAXIMUM WEIGHTED α-DIMENSIONAL MATCHING.

Theorem 2 *For every cardinality constrained problem with a monotone and α-augmentable objective, the greedy algorithm has approximation/competitive ratio at most $\alpha \frac{e^\alpha}{e^\alpha - 1}$. This bound is tight for problems with 2-augmentable objectives, and, the greedy algorithm has competitive ratio at least α for α-augmentable objectives with $\alpha \in \mathbb{N}$.*

We point out that submodularity implies 1-augmentability (Proposition 2) which, in turn, implies accountability (Proposition 3), but not vice-versa (Observation 5). In particular, Theorem 2 generalizes the bound of $\frac{e}{e-1} \approx 1.58$ that is known for submodular functions to the larger class of 1-augmentable functions. For $\alpha = 2$, our tight bound is $\frac{2e^2}{e^2-1} \approx 2.313$. For $\alpha \to \infty$ our lower bound of α becomes tight, since $\frac{e^\alpha}{e^\alpha - 1} \to 1$. We note that the greedy algorithm is well known to yield an α-approximation for MAXIMUM WEIGHTED α-DIMENSIONAL MATCHING.

Related Work. Most work on incremental settings has focused on cardinality constrained *minimization* problems. A prominent exception is the robust matching problem, introduced by Hassin and Rubinstein [19]. This problem asks for a weighted matching M with the property that, for every value k, the total weight of the $\min(k, |M|)$ heaviest edges of M comes close to the weight of a maximum weight matching of cardinality at most k. Note that this differs from our definition of incremental matchings in that the robust matching problem demands that the "incremental" solution consists of a matching, while we allow any edge set that contains a heavy matching as a subset. Since their model is more strict, all of the following competitive ratios carry over to our setting. Note that, in contrast to our setting, the objective function of the robust matching problem is submodular, and hence the greedy algorithm has competitive ratio at most $\frac{e}{e-1} \approx 1.58$ [32]. Hassin and Rubinstein [19] gave an improved, deterministic algorithm that achieves competitive ratio $\sqrt{2} \approx 1.414$. They also give a tight example for this ratio of $\sqrt{2}$, which also works in our incremental setting. Fujita et al. [14] extended this result to matroid intersection, and Kakimura and Makino [21] showed that every independence system allows for a $\sqrt{\mu}$-competitive solution, with μ being the extendibility of the system. Matuschke at al. [28] describe a randomized algorithm for this problem that, under the assumption that the adversary does not know the outcome of the randomness, has competitive ratio $\ln(4) \approx 1.386$.

A variant of the knapsack problem with a similar notion of robustness was proposed by Kakimura et al. [22]. In this problem a knapsack solution needs to be computed, such that, for every k, the value of the k most valuable items in the knapsack compares well with the optimum solution using k items, for every k. Kakimura et al. [22] restrict themselves to polynomial time algorithms and show that under this restriction a bounded competitive ratio is possible only if the rank quotient of the knapsack system is bounded. In contrast, our results show that if we do not restrict the running time and if we only require our solution to *contain* a good packing with k items for every k, then we can be $(1+\varphi)$-competitive using our generic algorithm, even for generalizations like MULTI-DIMENSIONAL KNAPSACK. If we restrict the running time and use the well-known PTAS for the knapsack problem [20,26], we still get a $(1+\varphi)(1+\varepsilon)$-competitive algorithm. Megow and Mestre [29] and Disser et al. [9] considered another variant of the knapsack problem that asks for an order in which to pack the items that works well for every knapsack capacity. Kobayashi and Takazawa [23] study randomized strategies for cardinality robustness in the knapsack problem.

Hartline and Sharp [18] considered an incremental variant of the maximum flow problem where capacities increase over time. This is in contrast to our framework where the cardinality of the solution increases.

Incremental solutions for cardinality constrained minimization problems have been studied extensively, in particular for clustering [4,8], k-median [5,13,31], minimium spanning tree [2,15], and facility location [16]. An important result in this domain is the incremental framework given by Lin et al. [27]. This general framework allows to devise algorithms for every incremental minimization problem for which a suitable augmentation subroutine can be formulated. Lin et al. [27] used their framework to match or improve many of the known specialized bounds for the problems above and to derive new bounds for covering problems. In contrast to their result, our incremental framework allows for a general algorithm that works out-of-the-box for a broad class of incremental maximization problems and yields a constant (relatively small) competitive ratio.

Abstractly, incremental problems can be seen as optimization problems under uncertainty. Various approaches to handling uncertain input data have been proposed, ranging from robust and stochastic optimization to streaming and exploration. On this level, incremental problems can be seen as a special case of online optimization problems, i.e., problems where the input data arrives over time (see [3,12]). Whereas online optimization in general assumes adversarial input, incremental problems restrict the freedom of the adversary to deciding when to stop, i.e., the adversary may choose the cardinality k while all other data is fixed and known to the algorithm. Online problems with such a "non-adaptive" adversary have been studied in other contexts [6,11,17]. Note that online problems demand irrevocable decisions in every time step – a requirement that may be overly restrictive in many settings where solutions develop over a long time period. In contrast, our incremental model only requires a growing solution "infrastructure" and allows the actual solution to change arbitrarily over time within this infrastructure.

Regarding the greedy algorithm as an approximation algorithm for maximization problems, we mentioned above that it is well-known to achieve a competitive ratio of $\frac{e}{e-1}$ for monotone and submodular objectives [32]. Theorem 2 generalizes this result,

since 1-augmentability is a relaxation of submodularity (Proposition 2). The *submodularity ratio* was proposed by Das and Kempe [7] as a somewhat similar relaxation of submodularity, with a submodularity ratio greater zero implying an approximation guarantee for the greedy algorithm. We note that, while a submodularity ratio of 1 implies submodularity and thus 1-augmentability, otherwise, there is no relationship between a constant submodularity ratio and α-augmentability for constant α.

Another important setting where the greedy algorithm yields an approximation are independence systems with bounded rank quotient [24], where elements of the ground set are weighted and the objective function value of a set is given by the maximum sum of the weights over all independent subsets. This setting can be seen as a relaxation of matroids, i.e., independence systems with rank quotient 1, for which the greedy algorithm computes an optimum solution [10,33]. In fact, an independence system that is the intersection of k matroids is has rank quotient $1/k$ [24]. In particular, this implies that the greedy algorithm yields a d-approximation for the MAXIMUM WEIGHTED d-DIMENSIONAL MATCHING problem. Another relaxation of matroids are k-extendible systems, for which greedy yields a k-approximation [30]. Again, while a rank quotient of 1 (i.e., the matroid case) implies submodularity of the objective function [32] and thus 1-augmentability, otherwise, there is no relationship between a bounded rank quotient or bounded extendibility and α-augmentability for bounded α.

2 A competitive algorithm for accountable problems

In this section, we show the first part of Theorem 1, i.e., we give an incremental algorithm that is $(1 + \varphi \approx 2.618)$-competitive for all problems with a monotone and accountable objective. For convenience, we define the *density* δ_S of a set $S \subseteq U$ via $\delta_S := f(S)/|S|$, and we let $\delta_k^\star := \delta_{S_k^\star}$ denote the optimum density for cardinality k. Our algorithm relies on the following two observations that follow from the accountability of the objective function. We note that, throughout this section, we do not require sub-additivity.

Lemma 1 *If the objective f is accountable, then, for every k, there is an ordering $(s_1^\star, s_2^\star, \ldots, s_k^\star)$ of $S_k^\star$, such that $\delta_{\{s_1^\star,\ldots,s_i^\star\}} \geq \delta_{\{s_1^\star,\ldots,s_{i+1}^\star\}}$ for all $i \in \{1, \ldots, k-1\}$. We call $(s_1^\star, s_2^\star, \ldots, s_k^\star)$ a* greedy order *of $S_k^\star$.*

Proof By accountability of f, there is an element $s_k^\star \in S_k^\star$ for which

$$\delta_{S_k^\star \setminus \{s_k^\star\}} = \frac{f(S_k^\star \setminus \{s_k^\star\})}{k-1} \geq \frac{f(S_k^\star)}{k} = \delta_{S_k^\star}.$$

We can repeat this argument for $s_{k-1}^\star \in S_k^\star \setminus \{s_k^\star\}$, $s_{k-2}^\star \in S_k^\star \setminus \{s_k^\star, s_{k-1}^\star\}$, etc. to obtain the desired ordering. □

Lemma 2 *If the objective f is monotone and accountable, we have $\delta_{k'}^\star \geq \delta_k^\star$ for all $1 \leq k' \leq k$.*

Proof Fix any cardinality $k > 1$. By accountability of the objective function f, there is an element $s^\star \in S_k^\star$ with

$$\delta_k^\star = \frac{f(S_k^\star)}{k} \leq \frac{f(S_k^\star \backslash \{s^\star\})}{k-1} \leq \frac{f(S_{k-1}^\star)}{k-1} = \delta_{k-1}^\star.$$

It follows that $\delta_k^\star$ is monotonically decreasing in k. □

Now, we define $k_0 := 1$ and $k_i := \lceil (1+\varphi)k_{i-1} \rceil$ for all positive integers i. Our algorithm operates in phases $i \in \{0, 1, \dots\}$. In each phase i, we add the elements of the optimum solution $S_{k_i}^\star$ of cardinality k_i to our incremental solution in greedy order (Lemma 1). Note that we allow the algorithm to add elements multiple times (without effect) in order to not complicate the analysis needlessly (of course we would only improve the algorithm by skipping over duplicates). In the following, we denote by t_i the number of steps (possibly without effect) until the end of phase i, i.e., we let $t_0 := k_0$ and $t_i := t_{i-1} + k_i$.

Lemma 3 *For every phase $i \in \{0, 1, \dots\}$, we have $t_i \leq \varphi k_i$.*

Proof We use induction over i, with the case $i = 0$ being trivial, since $t_0 = k_0$. Now assume that $t_{i-1} \leq \varphi k_{i-1}$ for some $i \geq 1$. Using the property $\frac{\varphi}{\varphi+1} = \varphi - 1$ of the golden ratio, we get

$$t_i = t_{i-1} + k_i \leq \varphi k_{i-1} + k_i \leq \frac{\varphi}{\varphi+1} k_i + k_i = \varphi k_i.$$

□

Finally, we show the solution $\mathbf{S}$ computed by our algorithm is $(1+\varphi)$-competitive. As before, we let S_k denote the set of the first k elements of $\mathbf{S}$.

Theorem 3 *If the objective f is monotone and accountable, then, for every cardinality k, we have $f(S_k) \geq f_k^\star/(1+\varphi)$.*

Proof We use induction over k. The claim is true for $k = t_0 = 1$, since $S_1 = S_1^\star$ by definition of the algorithm. For the inductive step, we prove that if the claim is true for $k = t_{i-1}$, then it remains true for all $k \in \{t_{i-1}+1, \dots, t_i\}$. Recall that $k_i = \lceil (1+\varphi)k_{i-1} \rceil$. By Lemma 3, we have

$$t_{i-1} \leq \varphi k_{i-1} < k_i < t_{i-1} + k_i = t_i,$$

and we can therefore distinguish the following cases.

Case 1: $t_{i-1} < k < k_i$. Since $k > t_{i-1}$, our algorithm has already completed phase $i-1$ and added all elements of $S_{k_{i-1}}^\star$, so we have $f(S_k) \geq f_{k_{i-1}}^\star$. Because k is an integer and $k < k_i = \lceil (1+\varphi)k_{i-1} \rceil$, we have that $k < (1+\varphi)k_{i-1}$. By Lemma 2, we thus have

$$f_k^\star = \delta_k^\star \cdot k < \delta_{k_{i-1}}^\star \cdot (1+\varphi)k_{i-1} = (1+\varphi) f_{k_{i-1}}^\star \leq (1+\varphi) f(S_k).$$

Case 2: $k_i \leq k \leq t_i$. At time k, our algorithm has already completed the first $k - t_{i-1}$ elements of $S^\star_{k_i}$. Since the algorithm adds the elements of $S^\star_{k_i}$ in greedy order, we have $f(S_k) \geq (k - t_{i-1})\delta^\star_{k_i}$. On the other hand, since $k \geq k_i$, by Lemma 2 we have $f^\star_k = k \cdot \delta^\star_k \leq k \cdot \delta^\star_{k_i}$. In order to complete the proof, it is thus sufficient to show that $k \leq (1+\varphi)(k - t_{i-1})$. To see this, let $k = k_i + k'$ for some non-negative integer k'. Because t_{i-1} is integral, Lemma 3 implies $t_{i-1} \leq \lfloor \varphi k_{i-1} \rfloor$. Since φ is irrational and k_{i-1} is integral, φk_{i-1} cannot be integral, thus

$$k - t_{i-1} = k' + k_i - t_{i-1} \geq k' + \lceil (1+\varphi)k_{i-1} \rceil - \lfloor \varphi k_{i-1} \rfloor = k' + k_{i-1} + 1.$$

This completes the proof, since

$$(1+\varphi)(k - t_{i-1}) \geq (1+\varphi)(k' + k_{i-1} + 1) > k' + (1+\varphi)k_{i-1} + 1 \geq k' + k_i = k.$$

□

Corollary 1 follows if we replace $S^\star_{k_i}$ by an α-approximate solution for cardinality k_i.

3 Lower bound on the best-possible competitive ratio

In this section, we show the second part of Theorem 1, i.e., we give a lower bound on the best-possible competitive ratio for the maximization of incremental problems with monotone, sub-additive, and accountable objective functions. For this purpose, we define the Region Choosing problem. In this problem, we are given N disjoint sets $R_1, \ldots, R_N$, called *regions*, with region R_i containing i elements with a value of $\delta(i)$ each. We say that $\delta(i)$ is the *density* of region R_i. The total value of all elements in the region R_i is $v(i) := i \cdot \delta(i)$ for all $i \in \{1, \ldots, N\}$.

The objective is to compute an incremental solution $S \subseteq U := \bigcup_{i=1}^N R_i$ such that the maximum value of the items from a single region in S is large. Formally, the objective function is given by $f(S) := \max_{i \in \{1,\ldots,N\}} |R_i \cap S| \cdot \delta(i)$.

Observation 1 Region Choosing *has a monotone, sub-additive, and accountable objective function.*

Proof The objective function of Region Choosing is monotone by definition.

Let S, T be two solutions to an instance of the Region Choosing problem, and consider a region $R_{S \cup T} \in \arg\max_{i \in \{1,\ldots,N\}} |R_i \cap (S \cup T)| \cdot \delta(i)$ of maximum value in the solution $S \cup T$. Let v_X denote the total value of the items from region $R_{S \cup T}$ in solution X. Then,

$$f(S \cup T) = v_{S \cup T} \leq v_S + v_T \leq f(S) + f(T),$$

which proves sub-additivity.

To show accountability, let R_S be the region of maximum value in solution S. We need to find $s \in S$ with $f(S \backslash \{s\}) \geq f(S) - f(S)/|S|$. If $S \backslash R_S = \emptyset$, then, for every $s \in S$, we have

$$f(S \backslash \{s\}) = f(S) - f(S)/|S|.$$

Otherwise, for every $s \in S \backslash R_S$, we have

$$f(S \backslash \{s\}) = f(S) \geq f(S) - f(S)/|S|.$$

□

We state the following tight lower bound without proof.

Proposition 1 *The algorithm of the previous section has competitive ratio at least* $\varphi + 1$ *for* REGION CHOOSING *for* $N \to \infty$ *and with* $\delta(i) = 1$ *for all* $i \in \{1, \ldots, N\}$.

For our general lower bound, we set $\delta(i) := i^{\beta-1}$ for some $\beta \in (0, 1)$ that we will choose later. For this choice of β, we have $\delta(i) < \delta(j)$ and $v(i) > v(j)$ for $0 \leq j < i \leq N$. Also, for $N \to \infty$ we have $\lim_{i \to \infty} v(i) = \infty$. We call instances of the REGION CHOOSING problem in this form *β-decreasing*. Observe that in every β-decreasing instance the optimum solution of cardinality $i \leq N$ is to take all i elements from region R_i. This solution has value $f_i^\star = i^\beta$.

In order to impose a lower bound on the best-possible competitive ratio for β-decreasing instances, we need some insights into the structure of incremental solutions with an optimal competitive ratio. First, consider a solution that picks only $i' < i$ elements from region R_i. In this case, we could have picked i' elements from region $R_{i'}$ instead – this would only improve the solution, since densities are decreasing. Secondly, if we take i elements from region R_i, it is always beneficial to take them in an uninterrupted sequence before taking any elements from a region R_j with $j > i$: Our objective depends only on the region with the most value, therefore it never helps to take elements from different regions in an alternating fashion. This leads us the following observation.

Observation 2 *For every β-decreasing instance of* REGION CHOOSING *there is an incremental solution with optimal competitive ratio of the following structure: For $k_0 < k_1 < \cdots < k_m \in \mathbb{N}$ with $m \in \mathbb{N}$, it takes k_0 elements from region R_{k_0}, followed by k_1 elements from R_{k_1}, and so on, until finally k_m elements from region R_{k_m} are chosen.*

Thus, we can describe an algorithm for the REGION CHOOSING problem by an increasing sequence of region indices $k_0, \ldots, k_m$. Note that, in order to have a bounded competitive ratio if $N \to \infty$, we must have $m \to \infty$, since $\lim_{i \to \infty} v(i) \to \infty$. We are interested in a cardinality for which an incremental solution given by $k_0, \ldots, k_m$ has a bad competitive ratio. We define

$$\alpha_i := \frac{1}{k_i} \sum_{j=0}^{i} k_j \qquad \forall i \in \{0, \ldots, m\}.$$

Observe that $\alpha_i > 1$ for all $i \in \{1, \ldots, m\}$. We know that the value of the optimum solution for cardinality $\alpha_i k_i$ is $v(\alpha_i k_i) = (\alpha_i k_i)^\beta$, whereas the incremental solution only achieves a value of $v(k_i) = (k_i)^\beta$. This allows us to derive the following necessary condition on the α_i-values of ρ-competitive solutions.

Observation 3 *If an incremental solution defined by a sequence $k_0, \ldots, k_m$ is ρ-competitive for some $\rho \geq 1$, then, for all $i \in \{0, \ldots, m\}$, we must have*

$$\rho \geq \frac{v(\alpha_i k_i)}{v(k_i)} = \left(\frac{\alpha_i k_i}{k_i}\right)^{\beta} = \alpha_i^{\beta} \iff \alpha_i \leq \rho^{\frac{1}{\beta}}. \tag{2}$$

We will exclude a certain range of values of ρ by showing that we can find a $\beta \in (0, 1)$ such that, for a sufficiently large number of regions N, necessary condition (2) is violated. We do this by showing that, for some $i^\star \in \mathbb{N}$ and some fixed $\varepsilon > 0$, we have $\alpha_{i+1} - \alpha_i > \varepsilon$ for all $i \geq i^\star$, i.e., as i goes to ∞, condition (2) must eventually be violated. The following definition relates a value of $\beta \in (0, 1)$ to a lower bound on the competitive ratio ρ for β-decreasing instances.

Definition 1 A pair (ρ, β) with $\rho \geq 1$ and $\beta \in (0, 1)$ is *problematic* if there is $\varepsilon > 0$ such that for all $x \in (1, \rho^{1/\beta}]$ it holds that $h_{\rho,\beta}(x) < 0$, where

$$h_{\rho,\beta}(x) := (\rho^{\frac{1}{\beta}} + \varepsilon - x)^{\frac{1}{1-\beta}} - \frac{x}{x - 1 + \varepsilon}.$$

We show that problematic pairs indeed have the intended property.

Lemma 4 *If (ρ, β) is a problematic pair, then ρ is a strict lower bound on the competitive ratio of incremental solutions for β-decreasing instances of* Region Choosing.

Proof We fix a problematic pair (ρ, β) and let ε be as in Definition 1. Consider a β-decreasing instance of sufficiently large size N and assume that there is a ρ-competitive incremental solution for this instance, given by the sequence $k_0, k_1, \ldots, k_m$. Consider a cardinality $k = \sum_{j=0}^{i} k_j$ for any $i \in \{1, \ldots, N\}$ for which the incremental solution takes all elements from regions $R_{k_0}, \ldots, R_{k_i}$. Assume that we do not take any additional elements for larger cardinalities. We are interested in the first cardinality for which ρ-competitiveness would be violated, i.e., where $v(k_i)$ is not enough to be ρ-competitive. This is the minimal cardinality t_i with $f^\star_{t_i} = t_i^{\beta} > \rho k_i^{\beta} = \rho v(k_i)$, i.e.,

$$t_i := \min\left\{t_i \in \mathbb{N} \;\middle|\; t_i > \rho^{\frac{1}{\beta}} k_i\right\} \leq \rho^{\frac{1}{\beta}} k_i + 1.$$

Then, for cardinality t_i, the incremental solution must have taken enough value from a later region to be ρ-competitive. Without loss of generality, we can assume this region to be $R_{k_{i+1}}$, otherwise the incremental solution that skips region $R_{k_{i+1}}$ is also ρ-competitive, and we can consider this solution instead. It follows that the incremental solution must satisfy

$$\left(t_i - \sum_{j=0}^{i} k_j\right) \delta(k_{i+1}) \geq \frac{1}{\rho} v(t_i) > k_i^{\beta},$$

which, by definition of α_i, implies

$$\left(\rho^{\frac{1}{\beta}} k_i + 1 - \alpha_i k_i\right) \delta(k_{i+1}) > k_i^{\beta}. \tag{3}$$

Defining $q_i := k_i/k_{i-1}$ for all $i \in \{1, \ldots, N\}$ gives $\delta(k_{i+1}) = \delta(q_{i+1}k_i) = \delta(k_i) \cdot q_{i+1}^{\beta-1}$. With this, Eq. (3) can be written as

$$v(k_i) = k_i^{\beta} < \left(\rho^{\frac{1}{\beta}} + \frac{1}{k_i} - \alpha_i\right) q_{i+1}^{\beta-1} \delta(k_i) k_i = \left(\rho^{\frac{1}{\beta}} + \frac{1}{k_i} - \alpha_i\right) q_{i+1}^{\beta-1} v(k_i).$$

Dividing by $v(k_i)$ yields

$$q_{i+1} < \left(\rho^{\frac{1}{\beta}} + \frac{1}{k_i} - \alpha_i\right)^{\frac{1}{1-\beta}}. \tag{4}$$

Since the incremental solution is ρ-competitive, we have $\alpha_i \in (1, \rho^{\frac{1}{\beta}}]$ by condition (2). Because (ρ, β) is a problematic pair, we have

$$h_{\rho,\beta}(\alpha_i) = (\rho^{\frac{1}{\beta}} + \varepsilon - \alpha_i)^{\frac{1}{1-\beta}} - \frac{\alpha_i}{\alpha_i - 1 + \varepsilon} < 0. \tag{5}$$

Observe that since $\lim_{i\to\infty} k_i = \infty$, we can find an $i^\star$ such that $\frac{1}{k_i} < \varepsilon$ for all $i \geq i^\star$. Thus, for $i \geq i^\star$ Eqs. (5) and (4) imply

$$q_{i+1} < (\rho^{\frac{1}{\beta}} + \varepsilon - \alpha_i)^{\frac{1}{1-\beta}} < \frac{\alpha_i}{\alpha_i - 1 + \varepsilon}.$$

For $i \geq i^\star$, we therefore get

$$\begin{aligned} \alpha_{i+1} - \alpha_i &= \frac{1}{k_{i+1}} \sum_{j=0}^{i+1} k_j - \alpha_i \\ &= 1 + \frac{\alpha_i}{q_{i+1}} - \alpha_i \\ &> \varepsilon. \end{aligned}$$

But this implies that, if N is sufficiently large, condition (2) eventually gets violated, which contradicts the fact that the incremental solution has competitive ratio ρ. □

All that remains is to specify a problematic pair in order to obtain a lower bound via Lemma 4. It is easy to verify that (2.18, 0.86) is a problematic pair. Note that the resulting bound of 2.18 can slightly be increased to larger values below 2.19.

Theorem 4 *There is no* 2.18*-competitive incremental* REGION CHOOSING *algorithm.*

4 The greedy algorithm for augmentable problems

In this section, we analyze the greedy algorithm that computes an incremental solution $\mathbf{S}$ with $S_k = S_{k-1} \cup \{s_k\}$, where $s_k \in \arg\max_{s \in U \setminus S_{k-1}} f(S_{k-1} \cup \{s\})$ and $S_0 = \emptyset$. This algorithm is well-known to have a competitive ratio of $\frac{e}{e-1} \approx 1.58$ if the objective function f is monotone and submodular [32]. Note that every (non-negative) monotone and submodular function is sub-additive and accountable. On the other hand, in general, the greedy algorithm does not have a bounded competitive ratio for incremental problems.

Observation 4 *The greedy algorithm has an unbounded competitive ratio for many incremental problems with monotone, sub-additive, and accountable objectives, e.g., for the* KNAPSACK*, the* WEIGHTED INDEPENDENT SET*, and the* DISJOINT PATHS *problem.*

Proof We construct a knapsack instance with three types of items for small $\varepsilon > 0$ and any $k \in \mathbb{N}$: One item of size and value both $1-\varepsilon$, k items of size 2ε and value $1-2\varepsilon$, and k items of size and value both ε^2. Obviously, the greedy algorithm first takes the largest item and then continues with the smallest one, since each of them further increases the maximum value that can be packed in the knapsack of capacity 1. Consequently, the greedy solution has value below 1 for cardinality k, while the optimum value approaches k.

We can reproduce the same behavior for the WEIGHTED INDEPENDENT SET problem by choosing a star of degree k plus k isolated vertices as our input graph, where the center of the star has weight $1 - \varepsilon$, the leaves of the star have weight $1 - 2\varepsilon$, and the isolated vertices have weight ε^2.

Similarly, for DISJOINT PATHS, we can choose a long path and many isolated edges as input. The endpoints of the path form a pair of weight $1 - \varepsilon$, each edge along the path is a pair of weight $1 - 2\varepsilon$, and each isolated edge has weight ε^2. □

We will now define a subclass of incremental problems where the competitive ratio of greedy can be bounded. Observe that submodularity of a function $f : 2^U \to \mathbb{R}_{\geq 0}$ implies that, for every $S \neq T \subseteq U$, there exists an element $t \in T \setminus S$ with $f(S \cup \{t\}) - f(S) \geq (f(S \cup T) - f(S))/|T \setminus S|$. Accordingly, we can define the following relaxation of submodularity.

Definition 2 We say that $f : 2^U \to \mathbb{R}_{\geq 0}$ is *α-augmentable* for an $\alpha > 0$, if for every $S, T \subseteq U$ with $T \setminus S \neq \emptyset$ there exists an element $t \in T \setminus S$ with

$$f(S \cup \{t\}) - f(S) \geq \frac{f(S \cup T) - \alpha f(S)}{|T|}. \tag{6}$$

4.1 Characterization of augmentability

We first observe that augmentability is a relaxation of submodularity in the following sense. Note also that α-augmentability implies α'-augmentability for $\alpha' > \alpha$.

Proposition 2 *Every submodular set function is* 1*-augmentable. Not every* 1*-augmentable set function is submodular.*

Proof Assume that f is submodular, i.e., for all sets X, Y, we have

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y).$$

For every pair of sets S, T with $T \backslash S = \{e_1, \ldots, e_k\} \neq \emptyset$ it then follows that

$$\begin{aligned}\sum_{i=1}^{k} f(S \cup \{e_i\}) &\geq f(S) + f(S \cup \{e_1, e_2\}) + \sum_{i=3}^{k} f(S \cup \{e_i\}) \\ &\geq 2f(S) + f(S \cup \{e_1, e_2, e_3\}) + \sum_{i=4}^{k} f(S \cup \{e_i\}) \\ &\ldots \\ &\geq (k-1)f(S) + f(S \cup T).\end{aligned}$$

This implies that there exists $t \in T \backslash S$ with

$$f(S \cup \{t\}) \geq \frac{f(S \cup T) + (k-1)f(S)}{k} = \frac{f(S \cup T) - f(S)}{k} + f(S).$$

We rewrite this to obtain

$$f(S \cup \{t\}) - f(S) \geq \frac{f(S \cup T) - f(S)}{|T \backslash S|} \geq \frac{f(S \cup T) - f(S)}{|T|},$$

i.e., f is 1-augmentable.

It remains to provide a 1-augmentable function that is not submodular. One such function $f\colon 2^U \to \mathbb{R}_{\geq 0}$ with $U = \{e_1, e_2, e_3\}$ is given by

$$f(S) := \begin{cases} 7, & \text{if } S = U, \\ 6, & \text{if } S = \{e_1, e_2\}, \\ 5, & \text{if } e_1 \in S \text{ and } e_2 \notin S, \\ |S|, & \text{otherwise.} \end{cases}$$

It is easy to verify that this function is 1-augmentable, but it is not submodular (not even sub-additive), since $f(\{e_1, e_3\}) + f(\{e_2\}) < f(\{e_1, e_2, e_3\})$. □

We show the following relationship between 1-augmentability and accountability. We will see later (Observation 5) that no such relationship holds for α-augmentability with $\alpha \geq 2$.

Proposition 3 *Every* 1*-augmentable function* $f\colon 2^U \to \mathbb{R}_{\geq 0}$ *is accountable.*

Proof Consider an arbitrary subset $T \subseteq U$ with $n := |T|$ and let $S_0 := \emptyset, T_0 := T$. For $i = 1, \ldots, n-1$, we incrementally construct sets $S_i := S_{i-1} \cup \{t_i\}$ and $T_i := T_{i-1} \setminus \{t_i\}$, where we can choose $t_i \in T_{i-1} \setminus S_{i-1} = T_{i-1}$ such that

$$f(S_{i-1} \cup \{t_i\}) - f(S_{i-1}) \geq \frac{f(S_{i-1} \cup T_{i-1}) - f(S_{i-1})}{|T_{i-1}|},$$

because f is 1-augmentable. With $T_{i-1} \cup S_{i-1} = T$ and $|T_{i-1}| = n - i + 1$, this yields

$$f(S_i) \geq \frac{f(T) - f(S_{i-1})}{n - i + 1} + f(S_{i-1}) = \frac{f(T) + (n-i) f(S_{i-1})}{n - i + 1},$$

or

$$f(T) - f(S_i) \leq \left(\frac{n-i}{n-i+1}\right)(f(T) - f(S_{i-1})).$$

Letting $\{t_n\} := T_{n-1}$ and applying this inequality repeatedly yields

$$f(T) - f(T \setminus \{t_n\}) \leq \big(f(T) - f(S_0)\big) \cdot \prod_{i=1}^{n-1} \frac{n-i}{n-i+1} \leq \frac{f(T)}{|T|},$$

where we used that f is non-negative.

We conclude that t_n satisfies the requirement for accountability. □

Our definition of α-augmentability is meaningful in the sense that it induces an interesting subclass of incremental problems. In particular, we show that it captures the objective function of MAXIMUM (WEIGHTED) d-DIMENSIONAL MATCHING for $d = \alpha$.

Proposition 4 *The objective function of* MAXIMUM (WEIGHTED) d-DIMENSIONAL MATCHING *is always d-augmentable, but, for every $\alpha < d$, there exists an instance where it is not α-augmentable.*

Proof Let an instance of MAXIMUM WEIGHTED d-DIMENSIONAL MATCHING be given by $V_1, \ldots, V_d$ and $E \subseteq V_1 \times \cdots \times V_d w\colon E \to \mathbb{R}_{\geq 0}$. Consider $S, T \subseteq E$ with $T \setminus S \neq \emptyset$, and let M_S, $M_{S \cup T}$ be of maximum weight among all matchings that are subsets of S, and $S \cup T$, respectively. In particular, $f(S) = w(M_S)$ and $f(S \cup T) = w(M_{S \cup T})$.

For $e = (v_1, \ldots, v_d) \in M_{S \cup T} \setminus M_S$, let $M_{S,e} := \{e' = (v'_1, \ldots, v'_d) \in M_S \mid v_i = v'_i \text{ for some } i \in \{1, \ldots, d\}\}$ denote the set of elements of M_S that intersect e. Observe that, since $M_{S \cup T}$ is a d-dimensional matching, we have $\bigcup_{e \in M_{S \cup T} \setminus M_S} M_{S,e} \subseteq M_S \setminus M_{S \cup T}$, and that every $e' \in M_S$ is contained in at most d such intersection sets. We can lower bound $f(S \cup \{e\})$ by replacing in M_S the elements $M_{S,e}$ by the single

element e. This yields

$$\begin{aligned}\sum_{e\in M_{S\cup T}\setminus M_S}(f(S\cup\{e\})-f(S)) &\geq \sum_{e\in M_{S\cup T}\setminus M_S}\big(w(e)-w(M_{S,e})\big)\\ &\geq w(M_{S\cup T}\setminus M_S)-d\cdot w\Big(\textstyle\bigcup_{e\in M_{S\cup T}\setminus M_S}M_{S,e}\Big)\\ &\geq w(M_{S\cup T}\setminus M_S)-d\cdot w(M_S\setminus M_{S\cup T})\\ &= w(M_{S\cup T})-w(M_S\cap M_{S\cup T})-d\cdot w(M_S\setminus M_{S\cup T})\\ &= w(M_{S\cup T})-w(M_S)-(d-1)\cdot w(M_S\setminus M_{S\cup T})\\ &\geq w(M_{S\cup T})-d\cdot w(M_S)\\ &= f(S\cup T)-d\cdot f(S).\end{aligned}$$

Note that every summand on the left-hand side of this expression is non-negative, and summands for $e\in S$ are zero. It follows that, if $M_{S\cup T}\setminus S\neq\emptyset$, there must exist $t\in(M_{S\cup T}\setminus S)\subseteq T\setminus S$ with

$$\begin{aligned}f(S\cup\{t\})-f(S) &\geq \frac{f(S\cup T)-d\cdot f(S)}{|M_{S\cup T}\setminus S|}\\ &\geq \frac{f(S\cup T)-d\cdot f(S)}{|T\setminus S|}\\ &\geq \frac{f(S\cup T)-d\cdot f(S)}{|T|}.\end{aligned}$$

If $M_{S\cup T}\setminus S=\emptyset$, then $f(S\cup T)=f(S)$, and we can choose any $t\in T\setminus S\neq\emptyset$. Thus, f is d-augmentable.

To see that f is not α-augmentable for $\alpha\in(0,d)$, consider an (unweighted) instance of Maximum d-Dimensional Matching with $V_i=\{v_{i,1},\ldots,v_{i,d}\}$ for $i\in\{1,\ldots,d\}$ and $E=S\cup T$ with $S:=\{(v_{1,1},v_{2,2},\ldots,v_{d,d})\}$ and $T:=\bigcup_{i=1}^{d}\{(v_{1,i},\ldots,v_{d,i})\}$. For every $e\in T\setminus S$, we have $f(S\cup\{e\})=f(S)=1$, and thus

$$f(S\cup\{e\})-f(S)=0<\frac{d-\alpha}{d}=\frac{f(S\cup T)-\alpha f(S)}{|T|}.$$

Hence, f is not α-augmentable. □

We show next that 2-augmentability specifically captures the problems Maximum (Weighted) (b-)Matching and Maximum Bridge-Flow. Recall that, for $b\colon 2^E\to\mathbb{N}$, a b-matching in an unweighted graph $G=(V,E)$ is a set of edges $M\subseteq E$, such that the degree of every vertex v in (V,M) is upper bounded by $b(v)$. The problem Maximum Weighted b-Matching is defined anologously to Maximum Weighted Matching.

Proposition 5 *The objective function of* Maximum (Weighted) b-Matching *is always 2-augmentable, but, for every $\alpha<2$, there exists an instance where it is not α-augmentable.*

Proof Let $G = (V, E)$ be a graph, let $w\colon E \to \mathbb{R}_{\geq 0}$ be edge weights, let $b\colon V \to \mathbb{N}$ be vertex capacities, and let $f\colon 2^E \to \mathbb{N}$ such that $f(S)$ denotes the maximum weight of a b-matching in the subgraph (V, S) of G. Consider two edge sets $S, T \subseteq E$ with $S \neq T$ and let $M_S \subseteq S$ be a maximum weight b-matching in the graph (V, S), i.e., no vertex $v \in V$ is incident to more than $b(v)$ edges in M_S and $w(M_S) = f(S)$, where $w(X) := \sum_{e \in X} w(e)$. We let $d_{M_S}\colon V \to \mathbb{N}$ denote the vertex degrees of the subgraph (V, M_S), and define $l_{M_S}(v)$ to be the weight of the b-matching M_S that we lose if we need to reduce $b(v)$ by one, i.e., for every $v \in V$ we define

$$l_{M_S}(v) := \begin{cases} 0, & \text{if } d_{M_S}(v) < b(v), \\ \min_{e=\{v,w\} \in M_S} w(e), & \text{if } d_{M_S}(v) = b(v). \end{cases}$$

Now take any edge $e = \{u, v\}$ of the maximum weight b-matching $M_{S \cup T}$ in the graph $(V, S \cup T)$, and assume that we need to add e to M_S without violating vertex capacities, i.e., we may first need to remove edges from M_S to make room for e. If e is already part of M_S, we do not need to change the matching, and, in particular, the weight of the matching remains unchanged. Otherwise, by definition of l_{M_S}, we can ensure that the change in weight of the b-matching is at least

$$w(e) - l_{M_S}(u) - l_{M_S}(v).$$

Observe that, since $e \in M_{S \cup T}$, the weight values yielded by $l_{M_S}(u)$ and $l_{M_S}(v)$ must originate from edges in $M_S \backslash M_{S \cup T}$. Therefore, if we sum the above change over all edges in $M_{S \cup T} \backslash M_S$, and let $b'(v)$ denote the degree of v in the subgraph $(V, M_{S \cup T} \backslash M_S)$, we obtain

$$\begin{aligned} & w(M_{S \cup T} \backslash M_S) - \sum_{v \in V} b'(v) l_{M_S}(v) \\ & \quad \geq w(M_{S \cup T} \backslash M_S) - 2w(M_S \backslash M_{S \cup T}) \\ & \quad = w(M_{S \cup T}) - w(M_{S \cup T} \cap M_S) - 2w(M_S \backslash M_{S \cup T}) \\ & \quad = w(M_{S \cup T}) - w(M_S) - w(M_S \backslash M_{S \cup T}) \\ & \quad \geq w(M_{S \cup T}) - 2w(M_S) \\ & \quad = f(S \cup T) - 2f(S). \end{aligned}$$

Since M_S is a maximum matching in (V, S), no edge in S can have a positive contribution to this sum. If this expression is still positive, there must be an edge $e \in M_{S \cup T} \backslash S$ that increases the weight of M_S by at least

$$(f(S \cup T) - 2f(S))/|M_{S \cup T} \backslash S| \geq (f(S \cup T) - 2f(S))/|T \backslash S|,$$

and we get

$$f(S \cup \{e\}) - f(S) \geq (f(S \cup T) - 2f(S))/|T \backslash S|,$$

as claimed. Otherwise, the right-hand side is not positive, and the inequality is trivially satisfied by monotonicity of f. Hence, f is 2-augmentable.

To see that the objective function need not be α-augmentable for $\alpha < 2$, consider an unweighted path $P = (V, E)$ of length three with edges $e_1, e_2, e_3 \in E$ in this order along the path. Recall that $f(X)$ is the cardinality of a maximum matching in the subgraph (V, X) for every $X \subseteq E$. With this, for $S := \{e_2\}$ and $T := \{e_1, e_3\}$ we have

$$f(S \cup \{t\}) - f(S) = 0 < \frac{2-\alpha}{2} = \frac{f(S \cup T) - \alpha f(S)}{|T|},$$

for all $t \in T \backslash S$. Thus f is not α-augmentable for $\alpha < 2$. □

Proposition 6 *The objective function of* MAXIMUM BRIDGE-FLOW *is 2-augmentable but not α-augmentable for $\alpha < 2$.*

Proof Recall that for any subset $S \subset C$, $f(S)$ is the value of the maximum flow using edges in $E \backslash (C \backslash S)$. For any set $X \subseteq C$, let G_X be the graph that contains all the edges of X, plus all the edges in G that are not in C: Let f_S be some maximum flow in G_S, and let $\text{VAL}(f_S)$ be its value. By definition of the MAXIMUM BRIDGE-FLOW objective function we have $\text{VAL}(f_S) = f(S)$.

Now, f_S can also be viewed as a flow in $G_{S \cup T}$. Let $G^S_{S \cup T}$ be the residual graph in $G_{S \cup T}$ formed by flow f_S. Let f_r be the maximum flow in $G^S_{S \cup T}$ (r for residual), and let $\text{VAL}(f_r)$ be its value.

By the properties of residual graphs, we know that $f(S \cup T) = f(S) + \text{VAL}(f_r)$. Rearranging we get that $\text{VAL}(f_r) = f(S \cup T) - f(S)$. Now, by the property of flows, the flow f_r can be decomposed into source-sink paths and cycles, and we can define f_r to be a max flow in $G^S_{S \cup T}$ that contains only paths (no cycles), because such a max flow must always exist. There are two types of paths to consider in the decomposition of f_r: some use backwards edges in S, while others only forward edges in $S \cup T$. Note that the total capacity of all backwards residual edges in $G^S_{S \cup T}$ is $\text{VAL}(f_S) = f(S)$. Thus, if we let f_r^{nb} be the sub-flow of f_r that uses no backwards edges (nb for no backwards edges), and we let $\text{VAL}(f_r^{nb})$ be its value, then we have

$$\text{VAL}(f_r^{nb}) \geq \text{VAL}(f_r) - f(S) = f(S \cup T) - 2f(S). \tag{7}$$

Now, note that the flow f_r^{nb} is decomposed into paths which each cross the cut C exactly once, because by the problem definition the cut C is directed one way, and f_r^{nb} contains no backwards edges with which to go back to the source side of the cut. Moreover, none of these paths use any edges in S; if such a path existed, then it would use no edges in $C \backslash S$ (because it crosses the cut exactly once), so it could have been added to f_s in G_S, which contradicts f_S being the maximum flow in G_S. Thus, every flow-path in f_r^{nb} uses a single edge in $T \backslash S$ and no other edges in C. Now, for any edge $e \in T$, let $f_r^{nb}(e)$ be the flow on e in f_r^{nb}. We now argue that $f(S \cup \{e\}) - f(S) \geq f_r^{nb}(e)$. To see this, let $G^S_{S \cup \{e\}}$ be the residual graph of $G_{S \cup \{e\}}$ defined by flow f_S. By the properties of residual graphs we have that $f(S \cup \{e\}) - f(S)$ is precisely the value of the maximum flow in $G^S_{S \cup \{e\}}$. But note that all the flow-paths

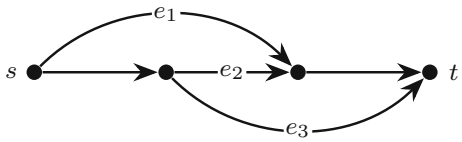

Fig. 2 Example of a MAXIMUM BRIDGE-FLOW instance with unit capacities that is not 1-augmentable

in f_r^{nb} that go through edge e do not go through any other edge in T, and so they are also valid flow-paths in $G^S_{S\cup\{e\}}$. Thus, the value of the maximum flow in $G^S_{S\cup\{e\}}$ is at least $f_r^{nb}(e)$, as desired.

Since every flow-path in f_r^{nb} only goes through a single edge in T we have $\text{VAL}(f_r^{nb}) = \sum_{e\in T} f_r^{nb}(e)$. Also, since f_S is a maximum flow in G_S, we have $f_r^{nb}(e) = 0$ for $e \in S$, and thus $\text{VAL}(f_r^{nb}) = \sum_{e\in T\setminus S} f_r^{nb}(e)$. Thus, there is some edge $e \in T$ with $f_r^{nb}(e) \geq \text{VAL}(f_r^{nb})/|T\setminus S|$, so by the argument in the paragraph above, $f(S\cup\{e\}) - f(S) \geq \text{VAL}(f_r^{nb})/|T\setminus S|$. Equation (7) then completes the proof.

To see that the objective function need not be α-augmentable for $\alpha < 2$, consider the graph $G = (V, E)$ in Fig. 2, where all arcs have capacity 1, and the directed cut is $C = \{e_1, e_2, e_3\}$. Recall that the objective function of the BRIDGE-FLOW problem is defined such that $f(X)$ is the value of a maximum flow in the graph $(V, (E\setminus C)\cup X)$ for $X \subseteq C$. With this, for $S := \{e_2\}$ and $T := \{e_1, e_3\}$ we have

$$f(S\cup\{t\}) - f(S) = 0 < \frac{2-\alpha}{2} = \frac{f(S\cup T) - \alpha f(S)}{|T|},$$

for all $t \in T\setminus S$. Thus f is not α-augmentable for $\alpha < 2$. □

4.2 Upper bound for the greedy algorithm

We now show the first part of Theorem 2, i.e., we show an upper bound on the competitive ratio of the greedy algorithm for incremental problems with monotone and α-augmentable objective functions. Note that, by Proposition 2, our result strengthens the known upper bound of $\frac{e}{e-1}$ for submodular objectives, by establishing the same bound for the slightly larger class of 1-augmentable set functions.

Theorem 5 *For every maximization problem with monotone and α-augmentable objective, the greedy algorithm is $\alpha\frac{e^\alpha}{e^\alpha-1}$-competitive.*

Proof Fix any $k \in \{1, \ldots, |U|\}$ and, for $i \in \{0, \ldots, k\}$, let S_i be our greedy incremental solution after i elements have been added. Then, for $i \in \{1, \ldots, k\}$, α-augmentability applied with $S = S_{i-1}$ and $T = S_k^\star$ guarantees the existence of an element $t \in S_k^\star\setminus S_{i-1}$ such that

$$f(S_{i-1}\cup\{t\}) - f(S_{i-1}) \geq \frac{f(S_{i-1}\cup S_k^\star) - \alpha f(S_{i-1})}{k} \geq \frac{f_k^\star - \alpha f(S_{i-1})}{k}, \quad (8)$$

where we used monotonicity of f.

Since we construct S_i greedily, we have $f(S_i) \geq f(S_{i-1} \cup \{t\})$. With (8) this yields

$$f(S_i) \geq \left(\frac{\alpha}{k} - 1\right)\left(\frac{1}{\alpha} f_k^\star - f(S_{i-1})\right) + \frac{1}{\alpha} f_k^\star,$$

or, equivalently,

$$\frac{1}{\alpha} f_k^\star - f(S_i) \leq \left(1 - \frac{\alpha}{k}\right)\left(\frac{1}{\alpha} f_k^\star - f(S_{i-1})\right).$$

Applying this inequality repeatedly, we obtain

$$\frac{1}{\alpha} f_k^\star - f(S_i) \leq \left(1 - \frac{\alpha}{k}\right)^i \left(\frac{1}{\alpha} f_k^\star - f(S_0)\right) \leq e^{-\frac{\alpha i}{k}} \frac{1}{\alpha} f_k^\star, \tag{9}$$

where we used that $f(S_0) \geq 0$ and $1 + x \leq e^x$ for $x \in \mathbb{R}$.

Finally, setting $i = k$ in (9) yields

$$f(S_k) \geq \frac{1}{\alpha}\left(1 - e^{-\alpha}\right) f_k^\star,$$

or $\frac{f_k^\star}{f(S_k)} \leq \alpha \frac{e^\alpha}{e^\alpha - 1}$. □

With this, we can complement Proposition 3 by showing that there is no relationship between accountability and α-augmentability for $\alpha \geq 2$.

Observation 5 *A monotone and accountable set function need not be α-augmentable for any α. Conversely, for every $\alpha \geq 2$, a monotone and α-augmentable set function need not be accountable.*

Proof The first part of the statement follows directly from Observation 4 and Theorem 5.

For the second part of the statement, consider ground set $U = \{e_1, e_2, e_3\}$ and the monotone function $f\colon 2^U \to \mathbb{R}_{\geq 0}$ given by

$$f(S) := \begin{cases} 4, & \text{if } S = U, \\ 2, & \text{if } S = \{e_1\}, \\ |S|, & \text{otherwise.} \end{cases}$$

It is easy to verify that f is 2-augmentable and thus α-augmentable for $\alpha \geq 2$. However, f is not accountable, since $f(U \backslash \{e\}) = 2 < 4 - \frac{4}{3} = f(U) - \frac{f(U)}{|U|}$ for all $e \in U$. □

4.3 Lower bounds for the greedy algorithm

We now show the second part of Theorem 2, i.e., we show lower bounds on the competitive ratio of the greedy algorithm. Note that the upper bound of $\alpha \frac{e^\alpha}{e^\alpha - 1}$ shown

above converges to α in the limit $\alpha \to \infty$. We first establish an asymptotically tight lower bound of α for α-augmentable objectives.

Theorem 6 *For $\alpha \in \mathbb{N}$, $\alpha \geq 2$, the greedy algorithm has a competitive ratio of at least α for the α-augmentable problem* MAXIMUM WEIGHTED α-DIMENSIONAL MATCHING.

Proof Consider an instance of MAXIMUM WEIGHTED α-DIMENSIONAL MATCHING with $V_i = \{v_{i,1}, \ldots, v_{i,2\alpha-1}\}$ for $i \in \{1, \ldots, \alpha\}$ and $E = E_1 \cup E_2 \cup E_3$, where we set $E_1 := \bigcup_{i=1}^{\alpha}\{(v_{1,i}, \ldots, v_{\alpha,i})\}$, $E_2 := \{(v_{1,1}, v_{2,2}, \ldots, v_{\alpha,\alpha})\}$, and $E_3 := \bigcup_{i=\alpha+1}^{2d-1}\{(v_{1,i}, \ldots, v_{\alpha,i})\}$. Let $w(e) = 1$ for $e \in E_1$, let $w(e) = 1 + \varepsilon$ for $e \in E_2$, and let $w(e) = \varepsilon$ for $e \in E_3$, where $\varepsilon > 0$ is arbitrarily small. The greedy algorithm computes an incremental solution $\mathbf{S}$ that first selects the element in E_2, then the elements in E_3, and finally the elements in E_1. In particular, we have $f(S_\alpha) = 1 + \alpha\varepsilon$ and $f_\alpha^\star = \alpha$. The competitive ratio of the greedy algorithm is thus bounded by

$$\rho \geq \frac{\alpha}{1 + \alpha\varepsilon} \to_{\varepsilon \to 0} \alpha.$$

□

For submodular objectives, it is well-known that the greedy algorithm has competitive ratio *exactly* $\frac{e}{e-1}$ [32]. By Proposition 2, the corresponding lower bound carries over to 1-augmentable objectives, which implies that our upper bound is tight beyond the class of all submodular functions. We now show that our upper bound is also tight for 2-augmentable objectives, which may be an indication that the bound is tight in general.

To show this lower bound, we construct the following family of instances for the MAXIMUM BRIDGE-FLOW problem. For $k \in \mathbb{N}$, we define a graph $G_k = (V_k, E_k)$ with designated nodes s and t by (see Fig. 3)

$$\begin{aligned}
V_k &:= \{s, t\} \cup \left\{v_i^1, v_i^4 \;\middle|\; i = 1, \ldots, 2k\right\} \cup \left\{v_i^2, v_i^3 \;\middle|\; i \in \{1, \ldots, 4k\}\right\}, \\
E_k &:= E_k^1 \cup E_k^\infty \cup \bigcup_{i=1}^{2k} E_{k,i} \cup \bigcup_{i=1}^{2k} E'_{k,i}, \\
E_k^1 &:= \left\{(s, v_i^2), (v_{3k+i}^3, t) \;\middle|\; i \in \{1, \ldots, k\}\right\}, \\
E_k^\infty &:= \left\{(s, v_{3k+i}^2), (v_i^2, v_i^3), (v_{3k+i}^2, v_{3k+i}^3), (v_i^3, t) \;\middle|\; i \in \{1, \ldots, k\}\right\}, \\
E_{k,i} &:= \left\{(s, v_i^1), (v_i^1, v_{k+i}^2), (v_{k+i}^2, v_{k+i}^3), (v_{k+i}^3, v_i^4), (v_i^4, t)\right\} \quad \forall i \in \{1, \ldots, 2k\}, \\
E'_{k,i} &:= \left\{(v_i^1, v_j^2), (v_{3k+j}^3, v_i^4) \;\middle|\; j = 1, \ldots, k\right\} \quad \forall i \in \{1, \ldots, 2k\}.
\end{aligned}$$

The edge capacities $u_k \colon E_k \to \mathbb{R}_{\geq 0}$ are given by $u_k(e) = (\frac{k}{k-1})^{2k+1-i}$ for $e \in E_{k,i}$, by $u_k(e) = \frac{1}{k}(\frac{k}{k-1})^{2k+1-i}$ for $e \in E'_{k,i}$, by $u_k(e) = 1$ for $e \in E_k^1$, and by $u_k(e) = \infty$ for $e \in E_k^\infty$.

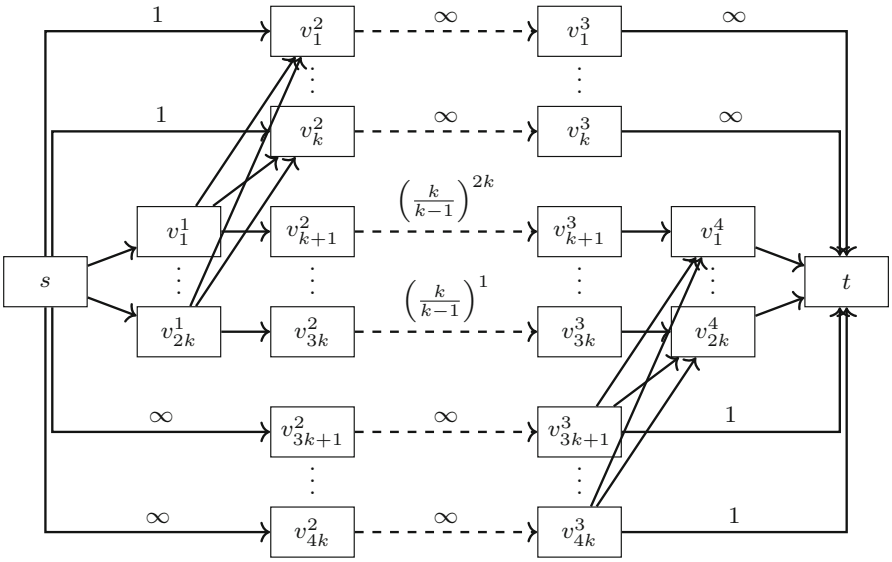

Fig. 3 The lower bound construction G_k for the greedy algorithm

For every G_k, we choose a directed s-t-cut $C_k := \left\{(v_i^2, v_i^3) \;\middle|\; i = 1, \ldots, 4k\right\}$.

Without loss of generality, we will assume in the following that we can resolve all ties in the greedy algorithm to our preference. This can be done formally by adding some very small offsets to the edge weights, but we omit this for clarity. Now consider how the greedy algorithm operates on graph G_k.

Lemma 5 *In step $j \in \{1, \ldots, 2k\}$, the greedy algorithm picks edge (v_{k+j}^2, v_{k+j}^3).*

Proof We prove the lemma by induction on j, starting with step $j = 1$, together with the fact that all picked edges can fully be saturated. Choosing $(v_{k+1}^2, v_{k+1}^3) \in C_k$ in the first step results in a possible s-t-flow of $(\frac{k}{k-1})^{2k}$ along the s-t-path in $E_{k,1}$, thus fully saturating the edge. By construction, selecting $(v_{k+i}^2, v_{k+i}^3) \in C_k$ with $i \in \{2, \ldots, 2k\}$ yields less s-t-flow, since these edges have lower capacity. Picking edge $(v_i^2, v_i^3) \in C_k$ with $i \in \{1, \ldots, k, 3k+1, \ldots, 4k\}$ results in a flow value bounded by the sum of the incoming edge capacities at v_i^2 or the outgoing edge capacities at v_{3k+i}^3. In either case, we get a flow of

$$
\begin{aligned}
1 + \frac{1}{k}\sum_{i=1}^{2k}\left(\frac{k}{k-1}\right)^i &= 1 + \frac{1}{k}\left(\frac{\left(\frac{k}{k-1}\right)^{2k+1} - 1}{\frac{k}{k-1} - 1} - 1\right) \\
&= 1 + \frac{1}{k}\left((k-1)\left(\frac{k}{k-1}\right)^{2k+1} - (k-1) - 1\right) \\
&= \left(\frac{k}{k-1}\right)^{2k}.
\end{aligned}
$$

Thus, no other edge results in more s-t-flow than edge (v^2_{k+1}, v^3_{k+1}), and with suitable tie-breaking we can ensure that the greedy algorithm picks this edge first, as claimed.

Now assume that greedy picked edges $(v^2_{k+1}, v^3_{k+1}), \ldots, (v^2_{k+j-1}, v^3_{k+j-1})$ before step j, and these edges can fully be saturated. Then, in step j, it can pick edge (v^2_{k+j}, v^3_{k+j}) to increase the s-t-flow value by $(\frac{k}{k-1})^{2k+1-j}$ along the s-t-path in $E_{k,j}$, thus saturating the edge. This is again better than selecting an edge $(v^2_{k+i}, v^3_{k+i}) \in C_k$ for $i \in \{j+1, \ldots, 2k\}$, since these edges have lower capacity. For the remaining edges, we need to account for the fact that the edges (s, v^1_i) and (v^4_i, t) for $i \in \{1, \ldots, j-1\}$ are already saturated, by induction. Therefore, the gain in flow value if we add any edge $(v^2_i, v^3_i) \in C_k$ with $i \in \{1, \ldots, k, 3k+1, \ldots, 4k\}$ in step j is

$$\begin{aligned}
1 + \frac{1}{k} \sum_{i=1}^{2k+1-j} \left(\frac{k}{k-1}\right)^i &= 1 + \frac{1}{k}\left(\frac{\left(\frac{k}{k-1}\right)^{2k+2-j} - 1}{\frac{k}{k-1} - 1} - 1\right) \\
&= 1 + \frac{1}{k}\left((k-1)\left(\frac{k}{k-1}\right)^{2k+2-j} - (k-1) - 1\right) \\
&= \left(\frac{k}{k-1}\right)^{2k+1-j}.
\end{aligned}$$

This is again not better than picking edge (v^2_{k+j}, v^3_{k+j}), and with suitable tie-breaking we can ensure that the greedy algorithm picks (v^2_{k+j}, v^3_{k+j}), as claimed. □

With this, we are ready to show the following result, which, together with Proposition 6, implies the second part of Theorem 2.

Theorem 7 *The greedy algorithm has competitive ratio at least* $\frac{2e^2}{e^2-1} \approx 2.313$ *for* MAXIMUM BRIDGE-FLOW.

Proof By Lemma 5, in the first $2k$ steps, the greedy algorithm picks the edges $(v^2_{k+1}, v^3_{k+1}), \ldots, (v^2_{3k}, v^3_{3k})$. Thus, after step $2k$, greedy can send an s-t-flow of value

$$\sum_{i=1}^{2k} \left(\frac{k}{k-1}\right)^i = \left(\frac{\left(\frac{k}{k-1}\right)^{2k+1} - 1}{\frac{k}{k-1} - 1} - 1\right) = (k-1)\left(\frac{k}{k-1}\right)^{2k+1} - k.$$

On the other hand, the solution of size $2k$ consisting of the edges $(v^2_1, v^3_1), \ldots, (v^2_k, v^3_k)$, and $(v^2_{3k+1}, v^3_{3k+1}), \ldots, (v^2_{4k}, v^3_{4k})$ results in an (optimal) flow value of

$$2k + 2\sum_{i=1}^{2k} \left(\frac{k}{k-1}\right)^i = 2(k-1)\left(\frac{k}{k-1}\right)^{2k+1}.$$

This corresponds to a competitive ratio of

$$\frac{2(k-1)\left(\frac{k}{k-1}\right)^{2k+1}}{(k-1)\left(\frac{k}{k-1}\right)^{2k+1}-k}=\frac{2\left(\frac{k}{k-1}\right)^{2k}}{\left(\frac{k}{k-1}\right)^{2k}-1}=\frac{2\left(\frac{k}{k-1}\right)^{2(k-1)+2}}{\left(\frac{k}{k-1}\right)^{2(k-1)+2}-1}.$$

Substituting $x := k-1$ and using the identity $\lim_{x\to\infty}(1+1/x)^x = e$, we get the lower bound on the competitive ratio of the greedy algorithm claimed in Theorem 2 in the limit:

$$\lim_{x\to\infty}\frac{2\left(\frac{x+1}{x}\right)^{2x}\left(\frac{x+1}{x}\right)^{2}}{\left(\frac{x+1}{x}\right)^{2x}\left(\frac{x+1}{x}\right)^{2}-1}=\lim_{x\to\infty}\frac{2e^2\left(\frac{x+1}{x}\right)^{2}}{e^2\left(\frac{x+1}{x}\right)^{2}-1}=\frac{2e^2}{e^2-1}.$$

□

5 Conclusion

We have defined a formal framework that captures a large class of incremental problems and allows for incremental solutions with bounded competitive ratios. We also defined a new and meaningful subclass consisting of problems with α-augmentable objective functions for which the greedy algorithm has a bounded competitive ratio. Hopefully our results can inspire future work on incremental problems from a perspective of competitive analysis.

The following open problems are left for future research:

1. Close the gap between our bounds of 2.618 and 2.18 for the best-possible competitive ratio of (deterministic) incremental algorithms.
2. Extend our lower bound of 2.18 to randomized algorithms and/or show that randomized algorithms can perform strictly better than deterministic algorithms in terms of competitive ratios.
3. For α-augmentable objectives, we showed that our bound of $\alpha\frac{e^\alpha}{e^\alpha-1}$ for the competitive ratio of the greedy algorithm is tight when $\alpha \in \{1, 2\}$ and when $\alpha \to \infty$. Prove or disprove that the bound is tight for all $\alpha \in \mathbb{N}$.
4. Prove whether or not the greedy algorithm is a best-possible incremental algorithm for α-augmentable objectives.

Acknowledgements We wish to thank Andreas Bärtschi and Daniel Graf for initial discussions and the the general idea that lead to the algorithm of Sect. 2, and David Weckbecker for valuable feedback for the manuscript. We are also very grateful for the insightful comments provided by anonymous reviewers.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If

material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Bernstein, A., Disser, Y., Groß, M.: General bounds for incremental maximization. In: Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP), pp. 43:1–43:14 (2017)
2. Blum, A., Chalasani, P., Coppersmith, D., Pulleyblank, B., Raghavan, P., Sudan, M.: The minimum latency problem. In: Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC), pp. 163–171 (1994)
3. Borodin, A., El-Yaniv, R.: Online Computation and Competitive Analysis. Cambridge University Press, Cambridge (1998)
4. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental clustering and dynamic information retrieval. SIAM J. Comput. **33**(6), 1417–1440 (2004)
5. Chrobak, M., Kenyon, C., Noga, J., Young, N.E.: Incremental medians via online bidding. Algorithmica **50**(4), 455–478 (2007)
6. Dani, V., Hayes, T.P.: Robbing the bandit: less regret in online geometric optimization against an adaptive adversary. In: Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 937–943 (2006)
7. Das, A., Kempe, D.: Approximate submodularity and its applications: subset selection, sparse approximation and dictionary selection. J. Mach. Learn. Res. **19**(3), 1–34 (2018)
8. Dasgupta, S., Long, P.M.: Performance guarantees for hierarchical clustering. J. Comput. Syst. Sci. **70**(4), 555–569 (2005)
9. Disser, Y., Klimm, M., Megow, N., Stiller, S.: Packing a knapsack of unknown capacity. SIAM J. Discrete Math. **31**(3), 1477–1497 (2017)
10. Edmonds, J.: Matroids and the greedy algorithm. Math. Program. **1**, 127–136 (1971)
11. Faigle, U., Kern, W., Turán, G.: On the performance of on-line algorithms for partition problems. Acta Cybern. **9**(2), 107–119 (1989)
12. Fiat, A., Woeginger, G.J. (eds.): Online Algorithms: The State of the Art. Springer, Berlin (1998)
13. Fotakis, D.: Incremental algorithms for facility location and k-median. Theor. Comput. Sci. **361**(2–3), 275–313 (2006)
14. Fujita, R., Kobayashi, Y., Makino, K.: Robust matchings and matroid intersections. SIAM J. Discrete Math. **27**(3), 1234–1256 (2013)
15. Goemans, M., Kleinberg, J.: An improved approximation ratio for the minimum latency problem. Math. Program. **82**(1–2), 111–124 (1998)
16. Greg Plaxton, C.: Approximation algorithms for hierarchical location problems. J. Comput. Syst. Sci. **72**(3), 425–443 (2006)
17. Halldórsson, M.M., Shachnai, H.: Return of the boss problem: competing online against a non-adaptive adversary. In: Proceedings of the 5th International Conference on Fun with Algorithms (FUN), pp. 237–248 (2010)
18. Hartline, J., Sharp, A.: Incremental flow. Networks **50**(1), 77–85 (2007)
19. Hassin, R., Rubinstein, S.: Robust matchings. SIAM J. Discrete Math. **15**(4), 530–537 (2002)
20. Ibarra, O.H., Kim, C.E.: Fast approximation algorithms for the knapsack and sum of subset problems. J. ACM **22**(4), 463–468 (1975)
21. Kakimura, N., Makino, K.: Robust independence systems. SIAM J. Discrete Math. **27**(3), 1257–1273 (2013)
22. Kakimura, N., Makino, K., Seimi, K.: Computing knapsack solutions with cardinality robustness. Jpn. J. Ind. Appl. Math. **29**(3), 469–483 (2012). https://doi.org/10.1007/s13160-012-0075-z
23. Kobayashi, Y., Takazawa, K.: Randomized strategies for cardinality robustness in the knapsack problem. Theor. Comput. Sci. **699**, 53–62 (2017)
24. Korte, B., Hausmann, D.: An analysis of the greedy heuristic for independence systems. Ann. Discrete Math. **2**, 65–74 (1978)

25. Krause, A., McMahan, H.B., Guestrin, C., Gupta, A.: Robust submodular observation selection. J. Mach. Learn. Res. **9**, 2761–2801 (2008)
26. Lawler, E.L.: Fast approximation algorithms for knapsack problems. Math. Oper. Res. **4**(4), 339–356 (1979)
27. Lin, G., Nagarajan, C., Rajaraman, R., Williamson, D.P.: A general approach for incremental approximation and hierarchical clustering. SIAM J. Comput. **39**(8), 3633–3669 (2010)
28. Matuschke, J., Skutella, M., Soto, J.A.: Robust randomized matchings. Math. Oper. Res. **43**(2), 675–692 (2018)
29. Megow, N., Mestre, J.: 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), pp. 495–504 (2013)
30. Mestre, J.: Greedy in approximation algorithms, pp. 528–539 (2006)
31. Mettu, R.R., Plaxton, C.G.: The online median problem. SIAM J. Comput. **32**(3), 816–832 (2003)
32. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—I. Math. Program. **14**(1), 265–294 (1978)
33. Rado, R.: Note on independence functions. Proc. Lond. Math. Soc. **7**, 300–320 (1957)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Aaron Bernstein[1] · Yann Disser[2] · Martin Groß[3] · Sandra Himburg[2]

✉ Yann Disser
disser@mathematik.tu-darmstadt.de

Aaron Bernstein
bernstei@gmail.com

Martin Groß
mgrob@uwaterloo.ca

Sandra Himburg
sandra.himburg94@gmail.com

1 Rutgers University New Brunswick, New Brunswick, USA

2 TU Darmstadt, Darmstadt, Germany

3 University of Waterloo, Waterloo, Canada