Detailed description of the research program

1 Scientific background

Many problems in combinatorial optimization can be cast as special cases of the following general task. Given a ground set $\mathcal{N}$ of weighted elements, find a maximum weight subset of $\mathcal{N}$ obeying some constraint $\mathcal{C}$. In general, one cannot get any reasonable approximation ratio for this general task since it captures many hard problems such as maximum independent set in graphs. However, the literature includes many interesting classes of constraints for which the above task becomes more tractable. In this context, a constraint is represented as an ordered pair $(\mathcal{N}, \mathcal{I})$, where $\mathcal{N}$ is the ground set of the constraint and $\mathcal{I}$ is the collection of all subsets of $\mathcal{N}$ that are feasible. For example, if the constraint is a cardinality constraint allowing the selection of up to k elements (for some value k), then $\mathcal{I}$ will be the set $\{S \subseteq \mathcal{N} : |S| \leq k\}$. It is customary to refer to the sets of $\mathcal{I}$ as the *independent* sets of the constraint.

Perhaps the most well-known class of constraints is the class of matroids. A constraint $(\mathcal{N}, \mathcal{I})$ is a matroid if it obeys three properties.

Non-emptyness: Some subset of $\mathcal{N}$ is independent.

Down-monotonicity: If $A \subseteq \mathcal{N}$ is an independent set, then every subset of A is also independent.

Exchange-property: If $A, B \subseteq \mathcal{N}$ are two independent sets such that B is larger than A, then there exists an element $u \in B \setminus A$ such that $A \cup \{u\}$ is also independent.

Matroids were originally suggested as a generalization capturing both linear independence and combinatorial properties of interest. For example, it is not difficult to argue that if one considers a graph G and define a set of edges of G to be independent if and only if they form a forest, then this defines a matroid. Another class of constraints, known as *k-Intersection*, contains all constraints that can be represented as the intersection of k matroids. In other words, a constraint $(\mathcal{N}, \mathcal{I})$ belongs to k-Intersection if and only if there exist k matroids $(\mathcal{N}, \mathcal{I}_1), (\mathcal{N}, \mathcal{I}_2), \dots, (\mathcal{N}, \mathcal{I}_k)$ such that $\mathcal{I} = \bigcap_{j=1}^{k} \mathcal{I}_j$. For $k = 1$, the class of k-Intersection is of course equal to the class of matroids. However, for larger values of k this class captures more involved constraints that are not captured by a matroid. For example, a k-dimensional hypergraph is a hypergraph whose vertices can be partitioned into k "sides" in such a way that each edge intersects exactly one vertex of each side. The set of matchings in a k-dimensional hypergraph is a k-Intersection constraint since it can be represented as the intersection of k matroids, each one of them corresponding to one side of the hypergraph and allowing only sets of edges that do not intersect on any vertex of the given side. Note that this implies that the problem known as k-Dimensional Matching, which is a generalization of bipartitie matching asking for a maximum weight matching in a k-dimensional hypergraph, can be cast as a special case of the problem of finding a maximum weight independent set subject to a k-Intersection constraint.

The last problem was first studied by Jenkyns [26] who showed that a simple greedy algorithm achieves k-approximation for it. More than 30 years later, this was slightly improved to $(k-1+\varepsilon)$-approximation (for any positive constant ε) by Lee et al. [33], which is the state-of-the-art up to this day—except in the case of $k = 2$, for which a polynomial time algorithm is long known (see, *e.g.*, [40]). On the inapproximability side, Hazan et al. [24] showed that these approximation ratios cannot be improved by more than an $O(\log k)$ factor, even in the very special case of k-Dimensional Matching.

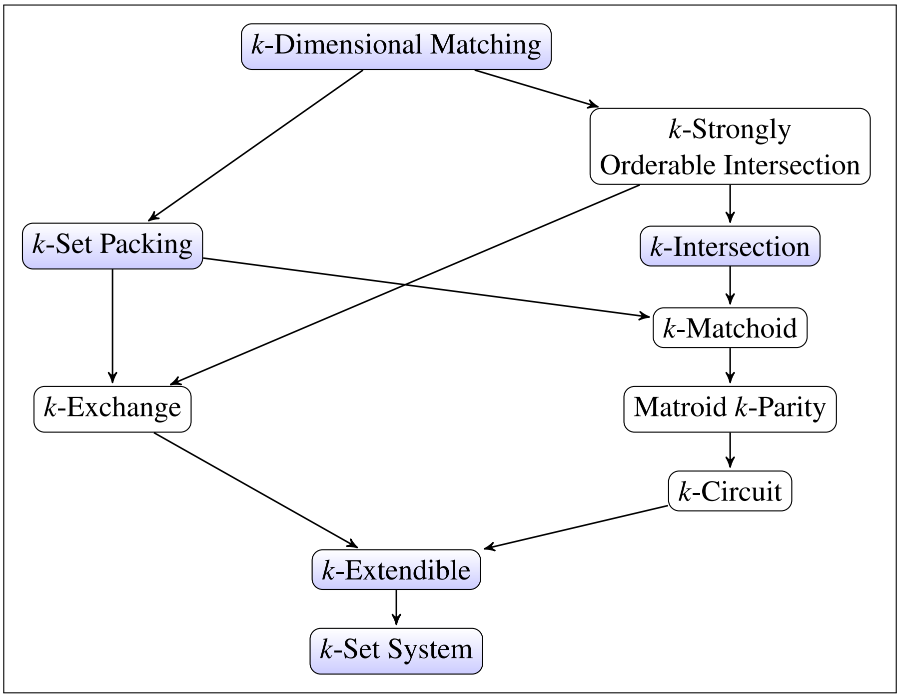

Figure 1: An hierarchy of the combinaotiral constraint classes. An arrow between two classes indicates that the upper class is a subset of the lower one. The classes marked with a blue shade are discussed in the text. For more information about the other classes, see [17, 25, 26, 34].

Recall that k-Dimensional Matching is the problem of finding a maximum weight matching in a k-dimensional hypergraph. A slight generalization of this problem is k-Set Packing which asks for finding such a matching in a general k-uniform hypergraph. Intuitively, k-Set Packing is not much more involved than k-Dimensional Matching. However, it turns out that while the constraint of k-Dimensional Matching is a k-Intersection constraint, the constraint of k-Set Packing does not belong to $f(k)$-Intersection for any function f of k. To ameliorate this issue, a vast hierarchy of classes generalizing k-Intersection and capturing k-Set Packing have been suggested in the literature, and each class in this hierarchy provides a different trade-off between generality and tractability. A general view of the hierarchy can be found in Figure 1. In general, there is no well-established common name in the literature for all the classes mentioned in Figure 1. However, since the definitions of these classes tend to be combinatorial in nature, we refer to them in this proposal as combinatorial constraint classes. We note that this name sets these classes apart from other unrelated classes of constraints studied in the literature, such as the polytope based class of unimodular constraints.

Due to a lack of space, we discuss in this proposal only three out of the classes mentioned in Figure 1. These classes include the k-Intesection class discussed above, and two classes known as k-Set Systems and k-Extendible Systems. These two classes are particularly interesting because they generalize all the other standard combinatorial classes (which implies that algorithms designed for constraints from these classes apply to all the other combinatorial constraint classes).

The class of k-Extendible System was defined by Mestre [35]. A constraint $(\mathcal{N}, \mathcal{I})$ belongs to this class if it obeys the non-emptyness and down-monotonicity properties of matroids, and in addition it obeys the following modified exchange property.

Exchange-property for k-Extendible Systems: If $A \subseteq B \subseteq \mathcal{N}$ are two independent sets, and u is an element outside B such that $A \cup \{u\}$ is independent, then there must exist a set $Y \subseteq B \setminus A$ of size at most k such that $(B \setminus Y) \cup \{u\}$ is also independent.

One can observe that this class indeed captures matching in general k-uniform hypergraphs because the set Y can be chosen as all edges of B intersecting the edge u. The class of k-Set Systems was introduced much earlier in the 1970's by Jenkyns [26] and Korte and Hausmann [31] (independently). Unlike the definitions given above for matroids and k-Extendible constrants which are based on exchange properties between pairs of independent sets, the definition k-Set System constraints is based on a structural property of maximal independent sets. This difference makes it much more difficult to work algorithmically with k-Set Systems, but makes it easy to argue that quite complex constraints fall into this class. For example, consider the constraint asking to select a subset of the edges of some graph G which form a planar subgraph of G. It is not difficult to use the fact that every planar graph can be decomposed into 3 trees to argue that this constraint is a 3-Set System.

Despite the very different definitions of k-Extendible and k-Set System, basically the same results are known for the problem of finding a maximum weight independent set subject to constraints from these two classes. On the positive side, the greedy algorithm obtains k-approximation for this problem, and on the hardness side, it was recently shown by [3, 15] that no algorithm can improve over this approximation ratio even when all the elements share an identical weight.

1.1 Submodular Maximization

Up to this point, we have discussed the problem of maximizing a given weight function subject to a constraint. However, many practical problems require a more involved kind of objective function. In particular, *submodular* objective functions have found many applications in diverse fields, including machine learning [12, 20, 39], social networks [23, 28] and algorithmic game theory [13, 41]. Thus, interest has arose in studying the maximization of submodular functions subject to the combinatorial constraint classes.

A submodular function is called *monotone* if the value it gives for a set of elements cannot decrease when additional elements are added to the set. This is a natural property of many important submodular functions, and thus, many works about submodular maximization assume monotonicity. Interestingly, it turns out that the results that can be achieved for maximizing monotone submodular functions subject to various combinatorial classes of constraints are almost identical to the results that can achieved for maximizing linear functions (weights) over these classes. In particular, Fisher et al. [18] showed already in 1978 that the greedy algorithm achieves $(k+1)$-approximation for maximizing a monotone submodular function subject to a k-Set System constraint, which is almost optimal even for the more restricted class of k-Extendible constraints due to the above mentioned hardness result of Feldman et al. [16] (in fact, [16] showed also a slightly stronger inapproximability of $k + 1/2$ for monotone submodular objectives, showing a slight separation with respect to the k-approximation that can be obtained for linear objectives). For the problem of maximizing a monotone submodular function subject to a k-Intersection constraint, a slightly better approximation ratio of $k + \varepsilon$ (for any positive constant ε) is also known [33].

The situation with respect to maximizing general (not necessarily monotone) submodular functions is more interesting. In particular, the greedy algorithm does not achieve any constant approximation ratio for the maximization of such functions even subject to a simple cardinality constraint. Nevertheless, Gupta et al. [21] showed that by iteratively combining executions of the greedy algorithm with a procedure for unconstrained submodular maximization, one can get $(4k+O(1))$-approximation for maximizing a submodular function subject to a k-Set System constraint. The analysis of this approach was improved by [36], and then again by [16] who obtained the state-of-the-art $(k+O(\sqrt{k}))$-approximation for this problem. Feldman et al. [16] also showed that by running greedy on a random subset of the ground set one can get an improved $(k+2+1/k)$-approximation for the special case of a k-Extendible constraint, which almost matches the state-of-the-art $(k+1+1/(k-1)+\varepsilon)$-approximation known for k-Intersection [33].

1.2 Streaming

The increasing interest in Big Data settings over the last decade has motivated the study of maximization subject to combinatorial classes of constraints in the data stream model. In this model the elements of the ground set arrive in a stream, and the objective of the algorithm is to find a good independent set while keeping its space complexity small. Specifically, almost all works in this field study semi-streaming algorithms, which are algorithms whose space complexity is nearly linear in the maximum size of an independent set under the given constraint.

The first work, to the best out knowledge, on data stream algorithms for maximization subject to combinatorial constraint classes is the work of Chakrabarti and Kale [7]. They obtained a $[2(k+\sqrt{k(k-1)})-1]$-approximation semi-streaming algorithm for finding a maximum weight set subject to a k-Intersection constraint, and a slightly worse $4k$-approximation semi-streaming algorithm for the more general case of a monotone submodular objective. Later works studied the same problem with general (not necessarily monotone) submodular objectives, and the state-of-the-art for this problem is currently $(2k+2\sqrt{k(k+1)}+1)$-approximation due to [16], improving over a previous result by [8].

While the above results match their offline counterparts up to constant factors, handling the more general k-Extendible and k-Set System classes in the data stream model turned out to be a much more challenging task. In particular, to date, there is no non-trivial semi-streaming algorithm for maximizing submodular functions subject to these classes of constraints. Interestingly, a reduction was recently presented by [37] for these problems, showing that any semi-streaming algorithm for maximizing a monotone submodular function subject to these classes of constraints can be made to work with general (not necessarily monotone) sumodular objectives at the cost of a slight increase in its approximation ratio. This reduction demonstrates the interest of the machine learning community in these problems. However, as mentioned above, so far no non-trivial algorithm has been presented for them, and thus, there is currrently no algorithm to which this reduction can be applied.

Crouch and Stubbs [9] presented a semi-streaming k^2-approximation algorithm for finding a maximum weight independent set subject to a k-Set System constraint. This result of course extends to the special case of k-Extendible constraints; and very recently, we were able to improve over it in this special case and obtain $\tilde{O}(k)$-approximation [14], which almost matches the situation in the offline setting. No similar improvement is currently known for general k-Set System constraints, and thus, there is currently a huge gap between the state-of-the-art approximation for these constraints and the k inapproximability result due to [3].

2 Research objectives and expected significance

This research proposal has three main objectives. The first objective is to advance our theoretical understanding of maximization subject to combinatorial classes of constraints. This class of problems is very mathematically clean and also includes some very fundamental and well known problems such as k-Set Packing. Thus, it is important to improve our theoretical understanding of the approximability of these problems.

The second objective of this proposal is to advance the algorithms used in practice for linear and submodular maximization. The offline variants of these problems are usually solved via the greedy algorithm in practice. As discussed above, the greedy algorithm is optimal from a theoretical point of view for many of these problems. However, its popularity in practice is more due to its great practical performance than due to its optimal but weak theoretical guarantee. Nevertheless, one can view the theoretical optimality of the greedy algorithm as an indication that this algorithm should be of good value also in practice. We believe the same kind of logic can be applied also to the data stream model and other models of interest. In other words, if we can find algorithms with close to optimal theoretical guarantees for these models, then these algorithms are likely to be of great value also in practice (as long as they are simple enough, which is usually the case in this field). It should be noted that many of the above mentioned works about theoretical algorithms also included an empirical study of the performance of the algorithms they suggested, showing that they are superior to various natural benchmarks (see, *e.g.*, [15, 16, 36, 37]).

The last objective of this proposal is the more abstract objective of studying combinatorial constraint classes in the context of additional models and problems. Naturally this objective is much more vague than the two previous ones. Thus, to clarify it, we list below two examples of research questions that fall under this objective.

- Up until recently, the combinatorial constraint classes have been studied only in the traditional offline computation model. Recent works, however, began the study of these classes in the context of two modern computational models motivated by Big Data applications: the data stream model, and the Map-Reduce model (see, *e.g.*, [10, 11]). We would like to extend this route of research, and study combinatorial constraint classes also in additional modern computation models, such as the property testing model.

- All the above works study maximization problems. However, it might be interesting to study also minimization problems involving combinatorial constraint classes. This has not been done much in the past since it is easy to convert maximization to minimization results in the offline case when the objective is linear, and minimization of submodular objectives is very hard even subject to a simple cardinality constraint [43] (which is a very special case of all the constraint classes we are interested in). Nevertheless, we believe that minimization of linear objectives subject to combinatorial constraint classes in non-offline models, such as the streaming model, is an interesting question for which it might be possible to obtain non-trivial results.

3 Detailed description of the proposed research

In the following pages we describe various open problems that we suggest studying in this proposal. For clarity, we have one sub-section for the offline setting and a different sub-section for the data stream model. Within each sub-section we present and discuss open problems corresponding to the two more concrete objectives described in Section 2. Namely, open problems that are clean and important, but nevertheless, are mostly of theoretical value; and open problems that we believe are likely to have also a significant impact on practical applications. We do not discuss explicit open problems related to the more vague objective presented in Section 2 because this objective is about new problems that have not been studied in the past.

3.1 Research Directions in the Offline Setting

Asymptotic Behavior for Large k Values. If one is only interested in the asymptotic behavior for large values of k, then maximization subject to combinatorial constraint classes is mostly well-understood in the offline setting. There are algorithms achieving $(k+o(k))$-approximation even for the general problem of maximizing a general submodular function subject to a k-Set System constraint, and this is optimal even for the much more specific problem of finding a maximum size independent set in a k-Extendible system [15]. Nevertheless, for k-Intersection constraints there is a gap between the above approximation ratio and the lower bound of $\Omega(k/\log k)$ due to [24]. Some works have striven to close this gap at the context of the very well known k-Set Packing problem, which is technically not a special case of finding a maximum weight independent set subject to a k-Intersection constraint, but is close to be such a special case. However, so far, these works have only managed to reduce the gap by a constant factor. Specifically, Berman [4] obtained $(k/2+O(1))$-approximation for k-Set Packing, and Sviridenko and Ward [42] further improved the approximation ratio for the unweighted version of this problem to $k/3+O(1)$.

This suggests the natural research avenue of improving over the results of [4] and [42] for k-Set Packing. However, this is probably a very difficult objective since all previous results for this problem, except for the basic k-approximation via the greedy algorithm, are based on local search approaches, and [42] provides a very strong impossibility result suggesting that the local search approach cannot be used to improve over their result. A more tractable open question is whether the technique of [4] can be applied to k-Intersection constraints, and lead to $(k/2+O(1))$-approximation for maximizing linear and submodular functions over such constraints. A few years ago, a similar question was answered in the affirmative by Ward [45] for a different class of constraints known as k-Exchange. However, doing the same for k-Intersection is more challenging, and is likely to require also ideas from the work of [33]—which is the only work to date improving over the simple k-approximation of the greedy algorithm in the context of k-Intersection constraints.

An opposite research avenue is to try to extend the inapproximability results of [3] and [15] to subsets of the k-Extendible class, with the hope of eventually getting some inapproximability for k-Intersection. In a nutshell, the inapproximability results of [3] and [15] are based on planting a large independent set inside a very symmetric base constraint, which makes it very difficult for an algorithm to identify the elements of the planted large independent set. The constraint obtained by this planting process can be easily shown to be a k-Set System, but Feldman et al. [15] had to "smooth" it in order to make it also a k-

Extendible constraint. The main obstacle for using the same technique for classes of constraints strictly included within the k-Extendible class is the need to develop more involved smoothing techniques that can make the constraint produced by the planting process fall within these classes.

Submodular Maximization subject to a Matching Constraint. As mentioned above, so far we have discussed questions that are interesting in the regime of large k values, which is mostly of theoretical interest since the approximation ratios that can be obtained for such large k values are quite poor. From a more practical point of view, the regime of small k values is more important. A particularly interesting special case that falls within this regime is the problem of maximizing a monotone submodular function subject to a matching constraint. In this problem one is given a graph and a submodular function over its edges, and the objective is to find a matching in the graph maximizing this submodular function. If the graph is bipartite, then this is a special case of maximizing a monotone submodular function subject to a 2-Intersection constraint, and if the graph is general, then this is still a special case of maximizing a monotone submodular function subject to a 2-Extendible constraint (in fact, even subject to a 2-Matchoid or a 2-Exchange constraint).

A $(2+\varepsilon)$-approximation is implied for the bipartite and general cases of this problem by the general results of Lee et al. [33] and Feldman et al. [17], respectively. One could expect that for the special case of a matching constraint one could improve over the results implied by these general works, but no such improvement was found despite the wide interest of the research community in this basic problem. On the inapproximability side, this problem generalizes the maximization of a monotone submodular function subject to a (simplified) partition matroid, and thus, cannot be approximated up to a ratio of $1-1/e+\varepsilon$ for any positive constant ε [44].

Recently, another long standing 2-approximation was improved over by new submodular optimization works. Specifically, the greedy algorithm obtains a 2-approximation for maximizing a monotone submodular function subject to a matroid constraint, and up until very recently the only known way to improve over that was via continuous extensions. These continuous extensions have large integrability gaps in the context matching constraints, and thus, cannot be used to break the 2-approximation for maximizing a monotone submodular function subject to a matching constraint. A series of recent works has suggested a more combinatorial approach for obtaining better than 2-approximation for matroid constraints [5, 6, 32], and it might be possible to use the ideas from this approach to get better algorithms also for matching constraints.

3.2 Research Directions in the Data Stream Model

Tightening the Approximation Ratio for Maximization subject to k-Intersection Constraints. Recall that Chakrabarti et al. [7] described a semi-streaming algorithm achieving roughly $4k$-approximation for the problem of finding a maximum weight independent set subject to a k-Intersection constraint. This result is not known to be tight even asymptotically, however, it will be very difficult to improve over its asymptotic behavior since the state-of-the-art offline algorithm for the problem achieves roughly $(k-1)$-approximation [33]. A more realistic objective is to try to improve over the constant in the guarantee of Chakrabarti et al. [7]. Given the offline results, it is natural to hope for an approximation ratio of $k+O(1)$. An evidence that it might be possible to achieve this goal can obtained from a recent result by Paz and Schwartzman [38]. A straightforward generalization of their result is a semi-streaming algorithm achieving k-approximation for the special case of the above problem in which the constraint can be represented as the intersection of k simplified partition matroids (as opposed to k general matroids as in the definition of the k-Intersection class).

In addition to improving over the constant in the asymptotic guarantee of Chakrabarti et al. [7], it is also interesting to improve over this guarantee in the regime of small k values, which is the regime that tends to be of more practical usefulness. For $k = 1$, a trivial local search semi-streaming algorithm can find a maximum weight independent set subject to a single matroid constraint (1-Intersection constraint). The exact approximation ratio given by Chakrabarti et al. [7] reproduces this guarantee for $k = 1$, however, its approximation guarantee becomes as large as $3+2\sqrt{2} \approx 5.828$ already for $k = 2$. Unfortunately, it seems that the analysis of the algorithm of Chakrabarti et al. [7] is tight, and thus, it will be necessary to introduce new ideas into the algorithm in order to improve over its approximation guarantee. One possible such idea is to use the local ratio approach, which is at the heart of the above mentioned result of Paz and Schwartzman [38] for partition matroids. A very different approach is to borrow ideas from the work of Korula et al. [32] who dealt with the maximization of a submodular function subject to a single matroid constraint, which is somewhat related to the maximization of a linear function subject to two matroid constraints due to the tight relationship between matroids and submodular functions.

Unweighted Matching in the Streaming Setting. As mentioned above, an easy generalization of the result of Paz and Schwartzman [38] yields a semi-streaming algorithm achieving k-approximation for the problem of finding a maximum weight set subject to the intersection of k simplified partition matroids. The original result of [38] applies only to the $k = 2$ case, and shows a 2-approximation semi-streaming algorithm for the problem of finding a maximum weight matching in a graph. Interestingly, this is the best known approximation ratio by a semi-streaming algorithm even for the unweighted variant of this problem. For the unweighted variant the same approximation ratio of 2 can also be obtained by simply outputting a maximal matching, and improving over this easy 2 has been open for a long time. No algorithm has managed so far to achieve this goal, but the interest of the community in it has led to many related results. Kapralov [27] showed that that no semi-streaming algorithm can obtain $e/(e-1) \approx 1.582$-approximation for this problem, and this was recently complemented by a result of Assadi et al. [2] showing a $(3/2)$-approximation data stream algorithm using $O(n^{3/2})$ space (note that the last algorithm is not a semi-streaming algorithm, and thus, does not contradict the previous inapproximability result). Another line of work relaxes the problem we consider by assuming the edges of the graph arrive at a uniformly random (rather than adversarial) order [19, 29, 30]. The best of these works describe semi-streaming algorithms achieving 1.976-approximation for finding a maximum size matching in a general graph [19] and an improved approximation ratio of 1.855 for the special case of a bipartite graph [29].

We have two goals in the context of these problems. One natural goal is of course to try to improve over the above results. Another goal is to see whether the ideas that allowed [19, 29, 30] to improve over the easy 2-approximation can be generalized to the problem of finding a maximum *size* independent set subject to a k-Intersection constraint. A k approximation can be obtained for this problem by simply outputting a base, and we hope the ideas from these works can be used to improve over it. We note that this was already done for the case of $k = 2$ in the random order model [22], but we are not aware of similar works for general k values or the adverserial order model.

Linear Maximization subject to k-Extendible Constraints. Up until a few months ago, the best semi-streaming algorithm for finding a maximum weight independent set subject to either a k-Extendible or a k-Set System constraint was a k^2-approximation algorithm by [9]. As mentioned above, very recently we managed to come up with a new algorithm improving over the algorithm of [9] and obtaining $O(k \log k)$-approximation for k-Extendible constraints [14]. Our algorithm consists of two main steps. The first step is a reduction to a special case of the problem consisting only of weights which are powers of k, and the second step is simply executing the algorithm of [9] on the output of the reduction. The crux of the analysis is using the properties of k-Extendible constraints to show that for the instances produced by the reduction (*i.e.*, instances containing only weights that are powers of k) the algorithm of [9] actually provides $O(k)$-approximation. Our result then follows by combining this with a loss of $O(\log k)$ due to the reduction itself.

The $O(k \log k)$-approximation we get is still larger by a factor of $O(\log k)$ compared to the $O(k)$-approximation known for k-Intersection constraints. From a theoretical point of view it is very interesting to close this gap, and ideas developed for this purpose might find applications also in improving the empirical performance of algorithms for this problem. One can of course try to close the gap by coming up with a completely new algorithm, however, we believe that it should also be possible to so by making some relatively minor modifications to our algorithm. Here we describe two modification ideas that have a potential to achieve this goal. The first idea is to multiply all the weights of the elements by a random constant factor. This will have no impact on the quality of the various solutions (since the value of every solution will change by exactly this factor), but it will make the partition of the elements into weight classes by the algorithm random. We hope that introducing randomness into the algorithm in this way will help it avoid carefully constructed bad instances, and thus, will allow us to prove for it an improved approximation guarantee (see [1] for an example of a very different setting in which this trick was used to obtain an optimal result). The second modification idea is to observe that the reduction losses an $O(\log k)$ factor only when the optimal solution has a specific structure. If this structure is exploitable by a different algorithm, then one should be able to get an improved approximation ratio by balancing the two algorithms.

Linear Maximization subject to k-Set System Constraints. Since our improvement in [14] applies only to k-Extendible constraints, the k^2-approximation algorithm by [9] is still the state-of-the-art for finding a maximum weight independent set subject to a general k-Set System constraint. We believe that this result cannot be improved by much, and thus, our intention is to try and prove a matching inapproximability result for this problem. One should note, however, a significant difference between the inapproximability that we want to prove here and the inapproximability results of [3, 15] mentioned above for related problems. The last two results apply even to the unweighted variants of the problem they consider. In contrast, the inapproximability result we would like to prove cannot apply to the unweighted version of our problem since the greedy algorithm can be used to get a k-approximation for this unweighted version (in the unweighted setting the greedy algorithm can be implemented as a semi-streaming algorithm). Thus, to prove such an inapproximability result we will have to come up with some new technical ideas. Since the results of [3, 15] apply to offline problems, the most natural source for such ideas are known inapproximability results for streaming problems. In particular, communication complexity lower bounds are often used in such inapproximation results, and we would like to look into the possibility of combining such a lower bound with ideas from the results of [3, 15].

Submodular Maximization subject to k-Extendible and k-Set System Constraints. As mentioned above, the machine learning community is very interested in finding semi-streaming algorithms for maximizing a submodular function subject to k-Extendible and k-Set System constraints, and this interest has already yielded a reduction between the monotone and non-monotone versions of this problem [37]. However, currently no algorithm with a non-trivial guarantee is known for either version of the problem, and thus, there is no algorithm on which the above reduction can be applied.

In most cases, results for maximization of linear objectives subject to combinatorial classes of constraints can be easily traslated into (slightly weaker) results for the maximization of submodular objectives subject to such constraints. However, making the algorithm of [9] (and our variant of it given by [14]) work for submodular objectives turned out to be non-trivial. The basic obstacle for that is that this algorithm maintains a set of solutions, and the decision of what solution to add an element to depends on the weight of the element. In the submodular context, the weight should be replaced with the marginal contribution of the element. However, since there are multiple solutions, it not clear with respect to which of them should we calculate this marginal contribution. One natural approach to answer this question is to make the decision whether to add the element to any given solution using the marginal with respect to that particular solution. Unfortunately, however, this does not seem to make the analysis of [9] go through.

We recently made an observation that the approximation ratios of [9] and [14] hold also with respect to the total weights of the maintained solutions, and not only with respect to the weight of the optimal solution. Thus, we believe that the above difficulty can be solved by considering the marginal contribution of the elements with respect to the *union* of all the maintained solutions. We plan to verify that this is indeed the case and also check the empirical performance of algorithms obtained via this observation in various machine learning applications.

References

[1] Noga Alon, Moran Feldman, and Moshe Tennenholtz. Revenue and reserve prices in a probabilistic single item auction. *Algorithmica*, 77(1):1–15, 2017.

[2] Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. In *SODA*, pages 1616–1635, 2019.

[3] Ashwinkumar Badanidiyuru and Jan Vondrák. Fast algorithms for maximizing submodular functions. In *SODA*, pages 1497–1514, 2014.

[4] Piotr Berman. A d/2 approximation for maximum weight independent set in d-claw free graphs. *Nord. J. Comput.*, 7(3):178–184, 2000.

[5] Niv Buchbinder, Moran Feldman, Yuval Filmus, and Mohit Garg. Online submodular maximization: Beating 1/2 made simple. In *IPCO*, pages 101–114, 2019.

[6] Niv Buchbinder, Moran Feldman, and Mohit Garg. Deterministic ($1/2 + \varepsilon$)-approximation for submodular maximization over a matroid. In *SODA*, pages 241–254, 2019.

[7] Amit Chakrabarti and Sagar Kale. Submodular maximization meets streaming: matchings, matroids, and more. *Math. Program.*, 154(1-2):225–247, 2015.

[8] Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. Streaming algorithms for submodular function maximization. In *ICALP*, pages 318–330, 2015.

[9] Michael Crouch and Daniel M. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *APPROX*, pages 96–104, 2014.

[10] Rafael da Ponte Barbosa, Alina Ene, Huy L. Nguyen, and Justin Ward. The power of randomization: Distributed submodular maximization on massive datasets. In *ICML*, pages 1236–1244, 2015.

[11] Rafael da Ponte Barbosa, Alina Ene, Huy L. Nguyen, and Justin Ward. A new framework for distributed submodular maximization. In *FOCS*, pages 645–654, 2016.

[12] Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *ICML*, pages 1057–1064, 2011.

[13] Shaddin Dughmi, Tim Roughgarden, and Mukund Sundararajan. Revenue submodularity. *Theory of Computing*, 8(1):95–119, 2012.

[14] Moran Feldman and Ran Haba. Almost optimal semi-streaming maximization for k-extendible systems. *CoRR*, abs/1906.04449, 2019.

[15] Moran Feldman, Christopher Harshaw, and Amin Karbasi. Greed is good: Near-optimal submodular maximization via greedy optimization. In *COLT*, pages 758–784, 2017.

[16] Moran Feldman, Amin Karbasi, and Ehsan Kazemi. Do less, get more: Streaming submodular maximization with subsampling. In *NeurIPS*, pages 730–740, 2018.

[17] Moran Feldman, Joseph Naor, Roy Schwartz, and Justin Ward. Improved approximations for k-exchange systems - (extended abstract). In *ESA*, pages 784–798, 2011.

[18] Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. *An analysis of approximations for maximizing submodular set functions—II*, pages 73–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.

[19] Buddhima Gamlath, Sagar Kale, Slobodan Mitrovic, and Ola Svensson. Weighted matchings via unweighted augmentations. In *PODC*, pages 491–500, 2019.

[20] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artif. Intell. Res.*, 42:427–486, 2011.

[21] Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. In *WINE*, pages 246–257, 2010.

[22] Guru Prashanth Guruganesh and Sahil Singla. Online matroid intersection: Beating half for random arrival. In *IPCO*, pages 241–253, 2017.

[23] Jason D. Hartline, Vahab S. Mirrokni, and Mukund Sundararajan. Optimal marketing strategies over social networks. In *WWW*, pages 189–198, 2008.

[24] Elad Hazan, Shmuel Safra, and Oded Schwartz. On the complexity of approximating k-set packing. *Computational Complexity*, 15(1):20–39, 2006.

[25] Tom A. Jenkyns. *Matchoids: A generalization of matchings and matroids*. PhD thesis, University of Waterloo, Waterloo, Ontario, 1974.

[26] Tom A. Jenkyns. The efficacy of the "greedy" algorithm. In *South Eastern Conference on Combinatorics, Graph Theory and Computing*, pages 341–350, 1976.

[27] Michael Kapralov. Better bounds for matchings in the streaming model. In *SODA*, pages 1679–1697, 2013.

[28] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.

[29] Christian Konrad. A simple augmentation method for matchings with applications to streaming algorithms. In *MFCS*, pages 74:1–74:16, 2018.

[30] Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *APPROX*, pages 231–242, 2012.

[31] Bernhard Korte and Dirk Hausmann. An analysis of the greedy heuristic for independence systems. *Annals of Discrete Math.*, 2:65–74, 1978.

[32] Nitish Korula, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. *SIAM J. Comput.*, 47(3):1056–1086, 2018.

[33] Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Math. Oper. Res.*, 35(4):795–806, 2010.

[34] Jon Lee, Maxim Sviridenko, and Jan Vondrák. Matroid matching: The power of local search. *SIAM J. Comput.*, 42(1):357–379, 2013.

[35] Julián Mestre. Greedy in approximation algorithms. In *ESA*, pages 528–539, 2006.

[36] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, and Amin Karbasi. Fast constrained submodular maximization: Personalized data summarization. In *ICML*, pages 1358–1367, 2016.

[37] Baharan Mirzasoleiman, Stefanie Jegelka, and Andreas Krause. Streaming non-monotone submodular maximization: Personalized video summarization on the fly. In *AAAI*, pages 1379–1386, 2018.

[38] Ami Paz and Gregory Schwartzman. A $(2+\varepsilon)$-approximation for maximum weight matching in the semi-streaming model. *ACM Trans. Algorithms*, 15(2):18:1–18:15, 2019.

[39] Mehraveh Salehi, Amin Karbasi, Dustin Scheinost, and R. Todd Constable. A submodular approach to create individualized parcellations of the human brain. In *MICCAI*, pages 478–485, 2017.

[40] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[41] Andreas S. Schulz and Nelson A. Uhan. Approximating the least core value and least core of cooperative games with supermodular costs. *Discrete Optimization*, 10(2):163–180, 2013.

[42] Maxim Sviridenko and Justin Ward. Large neighborhood local search for the maximum set packing problem. In *ICALP*, pages 792–803, 2013.

[43] Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM J. Comput.*, 40(6):1715–1737, 2011.

[44] Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM J. Comput.*, 42(1):265–304, 2013.

[45] Justin Ward. A $(k+3)/2$-approximation algorithm for monotone submodular k-set packing and general k-exchange systems. In *STACS*, pages 42–53, 2012.