Knapsack Cover Subject to a Matroid Constraint

Venkatesan T. Chakaravarthy[1], Anamitra Roy Choudhury[2], Sivaramakrishnan R. Natarajan[3], and Sambuddha Roy[4]

1 IBM Research, New Delhi, India, vechakra@in.ibm.com
2 IBM Research, New Delhi, India, anamchou@in.ibm.com
3 University of Washington, Seattle, USA, srk2siva@gmail.com
4 IBM Research, New Delhi, India, sambuddha@in.ibm.com

Abstract

We consider the Knapsack Covering problem subject to a matroid constraint. In this problem, we are given an *universe* U of n items where item i has attributes: a *cost* $c(i)$ and a *size* $s(i)$. We also have a demand D. We are also given a matroid $\mathcal{M} = (U, \mathcal{I})$ on the set U. A feasible solution S to the problem is one such that (i) the cumulative *size* of the items chosen is at least D, and (ii) the set S is *independent* in the matroid $\mathcal{M}$ (i.e. $S \in \mathcal{I}$). The objective is to *minimize* the total cost of the items selected, $\sum_{i \in S} c(i)$. Our main result proves a 2-factor approximation for this problem.

The problem described above falls in the realm of *mixed packing covering* problems. We also consider packing extensions of certain other covering problems and prove that in such cases it is not possible to derive any constant factor approximations.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Approximation Algorithms, LP rounding, Matroid Constraints, Knapsack problems

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2013.275

1 Introduction

In this paper, we consider the problem of computing the *minimum cost* Knapsack cover subject to *matroid* constraints. The Knapsack Cover problem (shortened as KC) is a minimization problem that takes as input, an universe U of n elements. Each element i is equipped with a cost $c(i)$ and a size $s(i)$. There is also a demand D; a feasible solution S is a collection of items such that the cumulative size is at least D. The objective in this problem is to find the feasible solution of minimum total cost. Thus, the KC problem is the *covering* version of the (more usual) knapsack packing problem.

The main problem considered in this paper is the KC problem subject to certain constraints that are called *matroid* constraints. In this scenario, in addition to the input for the KC problem as mentioned above, we are given a *matroid* $\mathcal{M} = (U, \mathcal{I})$ where $\mathcal{I} \subseteq 2^U$ is the family of *independent* sets (we give the formal definition of a matroid in Section 5). A set S is considered feasible iff (i) the cumulative size of S is at least D, and (ii) S is *independent* i.e. $S \in \mathcal{I}$.

We will denote the Knapsack Cover problem subject to a Matroid constraint as the KCM problem.

The KC problem naturally arises in various applications where we have a certain demand to fulfill and a certain number of options, varying in *profit* (i.e. *size*) and cost. For instance, consider a workplace where we need a certain number of developers for a certain project; and the project manager wants to outsource this to various teams/companies where each team

33rd Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2013).
Editors: Anil Seth and Nisheeth K. Vishnoi; pp. 275–286
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

can provide a certain number of developers at a specific cost. This is precisely the knapsack cover problem discussed above.

Given this context, consider the following variants. Any single company can either provide 5 developers at a cost of 100 units or 6 developers at a cost of 115 units, etc. In the framework of knapsack cover problems, we may model these different options as distinct items in the knapsack, but with a *partition matroid* constraint on these distinct items. In this setting this means that a feasible solution may pick at most one of the options from a single company; this is precisely what we want. Another (admittedly less realistic) scenario is where different companies have mutual incompatibilities; thus two companies may not be employed for the same project at the same time. Again, this corresponds to matroid constraints in addition to the demand fulfillment constraints. Thus these are all instances of the KCM problem.

The main result of this paper is the following:

▶ **Theorem 1.** *There is a* PTAS *for the* KCM *problem; specifically, given any* $\epsilon > 0$*, there is an algorithm that runs in time* $n^{\mathcal{O}(1/\epsilon)}$ *and outputs a* $(1+\epsilon)$*-factor approximate solution.*

Given that the KCM problem is a natural generalization of the KC problem, it is instructive to compare solution approaches for the KC problem. For the knapsack cover problem, a FPTAS is known via dynamic programming with parameter scaling. The natural LP for the KC problem has an unbounded integrality gap. Despite this, authors [4, 3] have shown how to achieve constant integrality gap via augmenting the natural LP with so-called "flow-cover inequalities". Carr et al. [4] first used such LP based relaxations and LP rounding to provide a 2-factor approximation for the KC problem (among other capacitated covering problems). Carnes and Shmoys [3] showed an elegant primal dual algorithm for the same LP to also derive a 2-factor approximation.

Note that while the vanilla version of the knapsack covering problem is a *covering* problem, in the KCM problem, we have both covering *and* packing constraints. Typically, such *mixed* packing covering problems are harder to analyse as compared to pure packing or pure covering problems. This is partially because for such problems, even checking the feasibility of the constraint set can be a NP-hard problem. However, for the KCM problem, the feasibility problem is indeed in polynomial time, as we show in Section 7. To the best of our knowledge the KCM problem has not been considered earlier in literature.

2 Our Contribution & Techniques

We prove the following results:

- Given the knapsack cover problem with a single matroid constraint, we show a PTAS.
- Given knapsack cover with multiple matroid constraints, the feasibility problem is NP-hard. Given this, we show a *bicriteria* approximation guarantee: we exhibit an algorithm that outputs a solution of value at most that of the optimal solution OPT that is *nearly-feasible*. The formal statement is given as Theorem 5 in Section 8.

Given the mixed packing/covering nature of the KCM problem, it is difficult to apply primal dual schemas; since the dual objective function in this context have both positive and negative coefficients. Previous literature has indeed considered primal dual schemas with dual objective functions having coefficients of either sign (for instance, see [9]). However in such cases, the primal dual algorithms give only *approximately* feasible solutions. One other possibility to consider are combinatorial algorithms. For instance there is a simple minded *greedy* algorithm for the knapsack cover KC problem, based on the *cost-effectiveness* $c(i)/s(i)$ of an element i. The algorithm proceeds as follows: it guesses the *costliest* element (say, of cost C^*)in the optimal cover OPT, and removes all the elements of cost larger than C^*. The algorithm considers the rest of the elements in increasing order of their cost-effectiveness, i.e. their $c(i)/s(i)$ values, and continues selecting elements while the covering requirement is not met. This gives a 2-factor approximation to the KC problem. However, there are certain issues in adapting this greedy approach to the KCM problem as we illustrate below.

For the KCM problem, if we pick up elements according to non-decreasing $c(i)/s(i)$ values, it may happen that we cannot pick any more elements without violating the matroid constraints and yet the cumulative size of the elements picked falls short of the requisite demand D. For instance, this happens in the following scenario. We are given as input, $U = \{1, 2, 3, 4\}$, with $D = 4$ and a cardinality matroid constraint over the whole universe with $k = 2$. The items ordered by their $c(\cdot)/s(\cdot)$ values are $\{\frac{1-\epsilon}{1}, \frac{2}{2}, \frac{2}{2}, \frac{3+\epsilon}{3}\}$. In this instance, OPT would pick either $\{1, 4\}$ or $\{2, 3\}$. The greedy solution would yield $\{1, 2\}$ of total cost $(3 - \epsilon)$ but satisfying only 3 units of the demand.

There exists another ordering natural for the problem. Picking elements in decreasing order of their $s(i)$'s while being feasible for the matroid constraints is the quickest way to satisfy the knapsack covering constraint; but this may cause the cost to blow up. For the above instance, this gives the (non-optimal) solution $\{3, 4\}$ of cost $(5 + \epsilon)$.

One natural idea then, is to proceed along an *amalgam* of the two orderings. Thus, one could start out with the knapsack greedy ordering, and then when some matroid constraint becomes tight, start *swapping* or *exchanging* items, while ensuring an improvement in $\sum s(i)x_i$ towards the target demand of D. We are not able to make such ideas work as of now.

On the other hand, in order to attempt LP rounding for the KCM problem, we have to surmount the obstacle of high integrality gap. For the KC problem, [3, 4] include the exponentially many *knapsack flow-cover* inequalities to overcome the unbounded gap. In the KCM setting, since we already have exponentially many matroid constraints, it would be preferable not to add another collection of exponentially many constraints.

As mentioned above, the natural LP for the problem has an unbounded integrality gap that it inherits from the LP for the KC problem. Nevertheless, we adopt a LP based approach, where we use properties of the *basic feasible solutions* of the LP. In the case of cardinality matroids, we also show a *cycle cancelling* approach to derive the desired result.

3 Other Results

For the special case of the KCM problem where the matroid is a *partition matroid*, we are able to show a FPTAS; this uses the dynamic programming approach along with parameter scaling. For space considerations, we defer the proof to the full version.

We consider certain other covering problems and augment such problems with matroid constraints. We demonstrate that the above cases are the exception rather than the rule.

As a sample, we consider the problem of *interval covering* with a partition matroid constraint. In this problem, we are given a collection of intervals $\mathcal{I} = \{I_1, I_2, \cdots, I_n\}$ over a time range $\mathcal{T} = \{1, 2, \cdots, T\}$. We are also given a partition $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_m$ of the intervals in $\mathcal{I}$. A feasible solution S is a collection of intervals such that (1) *every* timeslot in $\mathcal{T}$ is *covered* by some interval in S and (2) there is at most one interval chosen from each $\mathcal{P}_j$ (for $1 \leqslant j \leqslant m$). The objective is to choose the minimum number of intervals. We prove that testing feasibility in this case is NP-hard (and so, no constant factor approximation algorithm is possible for this problem). This is shown via a reduction from the Vertex Cover problem; details are provided in Section 9. On the other hand, the problem considered with a *cardinality matroid* (where, a feasible solution can pick at most a certain number of intervals from $\mathcal{I}$) has a dynamic programming solution that solves it in polynomial time.

It is more likely than not that augmenting a covering problem with a matroid constraint might render the feasibility problem NP-hard. As another instance, consider *Vertex Cover*. If we were to add a cardinality constraint, then the feasibility problem is an instance of *unweighted* Vertex Cover, and is NP-hard.

4 Related Work

In this paper, we consider the problem of knapsack covering subject to matroid constraints. This naturally leads us to the question of considering more general objectives; thus we might want to minimize an arbitrary *submodular* function subject to one knapsack covering constraint and a matroid constraint. However, note that in this case, strong lower bounds exist, even if we ignore the matroid constraint. Svitkina and Fleischer [14] prove the following: they show that given a submodular function $f(S)$ over a universe of size n, and a cardinality constraint $S \geqslant k$, it is NP-hard to get a factor better than $\Omega(\sqrt{\frac{n}{\log n}})$ for approximating the problem. They call this the SML (Submodular Minimization with cardinality Lower bounds) problem. Their lower bound is even stronger: it holds even for the special case of *monotone* submodular functions; and they derive lower bounds for bicriteria algorithms.

In the recent past, there has been a surge of work in the area of *submodular* maximization under various constraints. There have been several papers considering the problem of maximizing a monotone submodular function subject to matroid constraints, culminating in the breakthrough result of Vondrák [15](also see [2]). Vondrák [15] shows the optimal $(1 - 1/e)$-factor approximation for the problem via a *continuous greedy* process. Also, see the paper by Filmus and Ward [7] who give an elegant *non-oblivious* local search technique achieving the same approximation factor. In the space of non-monotone submodular functions, a recent result of Buchbinder et al. [1] (also see [6]) gives the optimal 1/2-factor approximation for the problem of unconstrained submodular maximization. The interested reader is referred to a presentation by Vondrák [16] (see slide 45) for an overview of the results in the area of submodular maximization subject to various (knapsack, matroid) constraints and their combinations.

Another line of work considers the problem of *minimizing* a submodular function subject to matroid constraints. Clearly, it does not make sense to minimize a monotone submodular function subject to such packing constraints. Thus, research has focused on the problem of minimizing a *symmetric* submodular function subject to matroid constraints. Originating from work by Shaddin [5], Goemans and Soto [8] prove a strong result that shows that symmetric submodular functions can be minimized subject to such constraints in polynomial time! In fact, their result extends to a wider class of constraints called "hereditary" constraints.

Thus, given the matroid constraints in the KCM problem, it is natural to ask about the relation between our problem and the existing literature. Here, we reiterate that most of the problems considered in literature are mostly purely packing problems: for instance, submodular maximization subject to matroid constraints or *symmetric* submodular minimization subject to matroid constraints. Our problem does not belong to the above frameworks because of the mixed packing covering nature of our constraints.

4.1 Organization

We present the relevant definitions in Section 5. We prove our result for the case of a cardinality matroid in Section 6 (see Theorem 2). We build on the case of cardinality matroids to give the proof of Theorem 1 in Section 7. We state and prove the bicriteria factor approximation for knapsack cover subject to multiple matroid constraints in Section 8. We conclude with discussions and open problems in Section 9.

5 Preliminaries

Sets: In this paper, we will use the following notation: given *disjoint* sets A and B we will use $A + B$ to serve as shorthand for $A \cup B$. Vice versa, when we write $A + B$ it will hold implicitly that the sets A and B are disjoint.

We will use the letter U for the universe; the universe will typically contain n elements. Given a set A, let $\chi(A)$ denote its *characteristic vector*: this is a vector such that $\{\chi(A)\}_i = 1$ if $i \in A$ and 0 otherwise.

Also given an element-wise function f with domain U, we will extend it to *subsets* in the natural way: $f(S) = \sum_{i \in S} f(i)$ for $S \subseteq U$.

Monotone: A set function f is called *monotone* if $f(S) \leqslant f(T)$ whenever $S \subseteq T$.

Submodular: A set function $f : 2^U \to \mathbb{R}^+$ over a universe U is called *submodular* if the following holds for any two sets $A, B \subseteq U$:

$$f(A) + f(B) \geqslant f(A \cup B) + f(A \cap B)$$

Matroid: A matroid is a pair $\mathcal{M} = (U, \mathcal{I})$ where $\mathcal{I} \subseteq 2^U$, and

1. (Hereditary Property) $\forall B \in \mathcal{I}, A \subset B \implies A \in \mathcal{I}$.
2. (Extension Property) $\forall A, B \in \mathcal{I} : |A| < |B| \implies \exists x \in B \setminus A : A + x \in \mathcal{I}$

Matroids are generalizations of vector spaces in linear algebra and are ubiquitous in combinatorial optimization because of their connection with greedy algorithms. Typically the sets in $\mathcal{I}$ are called *independent* sets, this being an abstraction of linear independence in linear algebra. The maximal independent sets in a matroid are called the *bases* (again preserving the terminology from linear algebra). An important fact for matroids is that all bases have equal cardinality – this is an outcome of the Extension Property of matroids.

Any matroid is equipped with a *rank function* $r : 2^U \to \mathbb{R}^+$. The rank of a subset S is defined to be the size of the *largest* independent set contained in the subset S. By the Extension Property, this is well-defined. The rank function of any matroid is well-known to be a *monotone* submodular function. See the excellent text by Schrijver [13] for details.

Two commonly encountered matroids are the (i) *Cardinality Matroid:* Given a universe U and $r \in \mathbb{N}$, the *cardinality matroid* is the matroid $\mathcal{M} = (U, \mathcal{I})$, where a set A is *independent* (i.e. belongs to $\mathcal{I}$) iff $|A| \leqslant r$. (ii) *Partition Matroid:* Given a universe U and a partition of U as $U_1, \cdots, U_r$ and non-negative integers $r_1, \cdots, r_t$, the *partition matroid* is $\mathcal{M} = (U, \mathcal{I})$, where a set A belongs to $\mathcal{I}$ iff $|A \cap U_i| \leqslant r_i$ for all $i = 1, 2, \cdots, t$.

Knapsack Cover with Matroid Constraints (KCM): We are given n items of sizes $s(1), s(2), \cdots, s(n)$, and with costs $c(1), c(2), \cdots, c(n)$. We are also given a cumulative demand D and a matroid $\mathcal{M}$. A feasible solution is a subset F such that the cumulative size of the subset F is at least D, and so that the set F is *independent* in the matroid $\mathcal{M}$. The objective is to produce a feasible solution of *minimum* cumulative cost.

Knapsack Cover with Cardinality Matroid (KCCard): In this variant of KCM, we are given a number k and a specific subset $A \subseteq \{1, 2, \cdots, n\}$. A feasible solution is a subset F such that the cumulative size of F is at least D and *no more* than k elements are chosen from the subset A.

6 KCM with Cardinality Matroids

In this section, we will consider the Knapsack Cover problem subject to a cardinality matroid constraint. Recall that in this scenario, we are given a subset A of the universe U and we may pick at most k elements from A in a feasible solution.

We use $x(A)$ as a shorthand for $\sum_{i \in A} x_i$. The LP is as follows:

$$\begin{array}{lll} & \min & \sum_i c_i \cdot x_i \\ \mathrm{LP}_1 & s.t. & \sum_i s_i \cdot x_i \geqslant D \\ & & x(A) \leqslant k \\ & \forall i & 0 \leqslant x_i \leqslant 1 \end{array}$$

We will call the constraints $x_i \geqslant 0$ and $x_i \leqslant 1$ as *trivial*; the constraints $\sum s_i \cdot x_i \geqslant D$ or $x(A) \leqslant k$ will be called the *non-trivial constraints*. In order that we may produce feasible solutions to the above LP, it is necessary to check that the feasibility problem is solvable in polynomial time. This is easy to do for the specific case of a cardinality matroid. In fact, we will prove the result in Lemma 3 for LPs corresponding to *arbitrary* matroid constraints.

▶ **Theorem 2.** *There is a* PTAS *for the* KCCard *problem.*

Proof. Let us consider a BFS solution to the above LP. It is easy to see that since there are *two* non-trivial constraints on the $x_i' s$, in a BFS solution, at most 2 variables will be set fractionally, and the other variables will be set integrally. Also note that the only way that a BFS solution may have *two* fractional variables is if *both* the constraints $\sum s_i \cdot x_i \geqslant D$ and $x(A) \leqslant k$ are *tight*.

Renaming variables, let the fractional variables be x_1 and x_2. Since the other variables are integral, the following equalities hold: $s_1 x_1 + s_2 x_2 = D'$ and $x_1 + x_2 = k'$ where k' is an integer. Clearly because of the constraints $0 \leqslant x_i \leqslant 1$ we have that $0 \leqslant k' \leqslant 2$. If $k' = 0$, then $x_1 = x_2 = 0$, contrary to their being fractional. Likewise if $k' = 2$, then $x_1 = x_2 = 1$, again a contradiction. Thus, the only case is that $k' = 1$. Thus D' is a convex combination of s_1 and s_2. Without loss of generality, let $s_1 \geqslant D' \geqslant s_2$. Since the constraint $x(A) \leqslant k$ is a packing constraint, we will not be able to pick *both* of x_1 and x_2. We will simply pick x_1 (this makes the constraint $\sum s_i x_i \geqslant D$ feasible), and set x_2 to be 0.

Thus, in this process we have raised at most 1 fractional variable.

We now use the idea (in the manner of [12]) of *pruning*. Let $c_{\max}$ be the item of *highest cost* in OPT. Although we do not know this item, we can *guess* this item by running through the n possibilities for such an item. Using this guess, we can remove items i from the LP that have $c_i > c_{\max}$. Thus, in the above process, when we raise x_1 from its current fractional value to 1, we increase the cost of our solution by at most $c_{\max}$. Thereby the total cost of the solution generated is $\mathrm{OPT} + c_{\max} \leqslant 2\mathrm{OPT}$.

In order to achieve a $(1 + \epsilon)$-factor approximation for any $\epsilon > 0$, we may guess *all* the items in OPT of cost at least $\epsilon \cdot \mathrm{OPT}$. Note that there can be at most $1/\epsilon$ such items. Let S denote this set of items, of total cost $C(S)$. Remove the items of S from the input instance to derive a *modified* instance. Thus, the modified instance has its parameters D and k appropriately reduced. Note that if S is indeed the set of items in OPT of cost at least $\epsilon \cdot \mathrm{OPT}$, then the optimal cost of the modified instance is at most $\mathrm{OPT} - C(S)$.

Since we remove all the items of S from the LP, for every item i still remaining in the LP, it holds that $c_i \leqslant \epsilon \cdot \mathrm{OPT}$. As before, the total cost of our solution is at most $(\mathrm{OPT} - C(S)) + \epsilon \cdot \mathrm{OPT} + C(S) = (1 + \epsilon)\mathrm{OPT}$.

Thus the algorithm considers all possible sets S of size $1/\epsilon$, and for each choice of S, solves the LP for the modified instance. The total run-time of the algorithm is $n^{\mathcal{O}(1/\epsilon)}$. ◀

Alternative Proof

The proof given above considers the BFS of the LP and proves certain properties of the BFS solution; in this sense, it is akin to *iterative rounding* (see [11]). However, one can provide another proof of the fact that any optimal solution to LP_1 can be modified to an optimal solution that contains at most *two* fractional variables. This proof goes via *cycle cancelling*.

In the normal usage of cycle cancelling, the "weight"(i.e. the LP values) is *shifted* between two variables. However, in the current scenario, we will need to shift the weight between *three* variables. This is the primary reason why we have *two* fractional variables.

In fact, the (more combinatorial styled) cycle cancelling technique applied to LP_1 shows that *any* feasible solution can be modified in this manner.

Suppose we are given a solution x to LP_1, and wlog rename the variables so that the fractional variables are $x_1, x_2, \cdots, x_m$. Given these fractional variables, we will attempt to change only the first 3 variables, by amounts δ_1, δ_2, and δ_3 (and leave the other variables - fractional or integral - unchanged). Thus the values of the variables x_1, x_2, x_3 will become $(x_1 + \delta_1), (x_2 + \delta_2), (x_3 + \delta_3)$. It is required that this operation does not violate the knapsack cover constraint or the cardinality constraint. We also want that in this process, the objective function does not degrade. These conditions translate to the following inequalities for $\delta_1, \delta_2, \delta_3$:

$$\begin{aligned}
&\delta_1 + \delta_2 + \delta_3 = 0\\
&s_1\delta_1 + s_2\delta_2 + s_3\delta_3 \geqslant 0\\
&c_1\delta_1 + c_2\delta_2 + c_3\delta_3 \leqslant 0
\end{aligned}$$

Since the system is homogeneous, the $(0, 0, 0)$ solution is always feasible. However, in order to perform cycle cancelling, we want a *non-zero* δ_i vector.

We may eliminate the variable δ_3 from the above system of inequations to get:

$$\begin{aligned}
&(s_1 - s_3)\delta_1 + (s_2 - s_3)\delta_2 \geqslant 0\\
&(c_1 - c_3)\delta_1 + (c_2 - c_3)\delta_2 \leqslant 0
\end{aligned}$$

The pertinent question is whether this system of inequalities always has a non-zero solution in (δ_1, δ_2). The two inequalities correspond to two halfplanes in $\mathbb{R}^2$. The halfplanes are intersecting - for instance, the point $(0, 0)$ lies on both of them. But the intersection of two halfplanes is an unbounded region, so has a non-zero vector in it.

For instance, if $s_1 = s_3$ and $c_1 = c_3$, we would need to set $\delta_2 = 0$, but we may set $\delta_1 = 1$ (and consequently, δ_3 is set to -1).

While this method works for the cardinality matroid (where we have a single constraint corresponding to the matroid constraint), we do not know how to make this work for an arbitrary matroid.

7 KCM with arbitrary Matroids

In this section, we extend the result in Theorem 2 to arbitrary matroids. Let us recall the problem: the universe U consists of n items, each item with attributes costs $c(i)$ and sizes $s(i)$. There is a minimum coverage demand D. We are also given a matroid $\mathcal{M} = (U, \mathcal{I})$. A *feasible* solution S to the KCM is one that satisfies the knapsack covering constraints and

Guess $c_{\max}$, the costliest element in OPT.
for each guess of $c = c_{\max}$ **do**
  $A \leftarrow \{i : c(i) > c_{\max}\}$
  Augment LP_2 with constraints $x_i = 0$, $\forall i \in A$
  Solve LP_2; let the two fractional variables be x_1 and x_2
  If $s_1 \geqslant s_2$, raise x_1 to 1, $x_2 = 0$;
  else $x_1 = 0$, $x_2 = 1$.
  Let this solution be denoted by S_c.
end for
Output the subset S_c with the minimum cost

Figure 1 Main Algorithm.

such that $S \in \mathcal{I}$. Let $r(\cdot)$ denote the *rank* function of the matroid M. Then the LP for the KCM problem stands as follows:

$$\begin{array}{lll} & \min & \sum_i c(i) \cdot x_i \\ \mathrm{LP}_2: & s.t. & \sum_i s(i) \cdot x_i \geqslant D \\ & \forall S & x(S) \leqslant r(S) \\ & \forall i & 0 \leqslant x_i \leqslant 1 \end{array}$$

Firstly, we show that we can solve the feasibility problem in polynomial time.

▶ **Lemma 3.** *The feasibility problem for* LP_2 *is solvable in polynomial time.*

Proof. Note that the feasibility problem may be converted into the following problem:

$$\begin{array}{lll} \mathrm{LP}_3: & \max & \sum_i s(i) \cdot x_i \\ & \forall S & x(S) \leqslant r(S) \\ & \forall i & 0 \leqslant x_i \leqslant 1 \end{array}$$

But this is precisely the maximum independent set question in a matroid and is well known to be solvable in polynomial time [13]. ◀

We are now ready to prove Theorem 1.

Proof (of Theorem 1). Let x^* denote a BFS solution to LP_2. Let there be ℓ fractional variables in the solution x^*. We will call any constraint that is not of the form $x_i \geqslant 0$ or $x_i \leqslant 1$ as "nontrivial". We will consider the non-trivial constraints that are satisfied with equality by the solution x^*. Given that there are ℓ fractional (and hence $n - \ell$ integral) variables, we have that precisely ℓ non-trivial constraints are tight. Moreover, this collection of tight constraints are *linearly independent*. We will also assume the following *normal* form for the tight constraints provided by a BFS. If a variable x_i is integral (either 0 or 1) in the solution x^*, we will consider the corresponding equation $x_i = 0$ or $x_i = 1$ as being tight. Thus, by virtue of linear independence of the tight constraints in a BFS, any non-trivial *tight* constraint has to contain at least *one* fractional variable.

There are two cases: either the constraint $\sum_i s(i)x_i \geqslant D$ is tight or is not. Let us consider the situation where this constraint is *not* tight. Thus the ℓ tight constraints are all of the form $x(S) \leqslant r(S)$ for some subset S. It is a well known property (for instance, see [11]) that the sets corresponding to these tight constraints may be assumed to form a *chain*.

We record this as a claim and for completeness, we prove this here. This proposition uses the fact that the rank function of a matroid is *submodular*.

▶ **Claim 4.** The *linearly independent* tight constraints $x(S) = r(S)$ can be assumed to form a *chain*. Thus, if there are ℓ linearly independent tight constraints, then the corresponding sets may be relabeled $S_1, S_2, \cdots, S_\ell$ such that $S_i \subset S_{i+1}$ for all $1 \leqslant i \leqslant (\ell - 1)$.

Proof. Consider any two tight constraints corresponding to sets S_1 and S_2. Thus, $x(S_1) = r(S_1)$ and $x(S_2) = r(S_2)$. Consider the following chain of inequalities:

$$\begin{aligned} r(S_1) + r(S_2) &\overset{tight}{=} x(S_1) + x(S_2) = x(S_1 \cap S_2) + x(S_1 \cup S_2) \\ &\leqslant\ r(S_1 \cap S_2) + r(S_1 \cup S_2) \overset{submodular}{\leqslant} r(S_1) + r(S_2) \end{aligned}$$

Thus equalities hold throughout the chain, and so $x(S_1 \cap S_2) = r(S_1 \cap S_2)$ and $x(S_1 \cup S_2) = r(S_1 \cup S_2)$. This means that $S_1 \cup S_2$ and $S_1 \cap S_2$ also are tight sets. Note that $\chi(S_1) + \chi(S_2) = \chi(S_1 \cap S_2) + \chi(S_1 \cup S_2)$. So, if S_1 and S_2 are such that $S_1 \setminus S_2 \neq \emptyset$ and $S_2 \setminus S_1 \neq \emptyset$, then we can replace one of the sets S_1 or S_2 in our system of linearly independent equations by $S_1 \cap S_2$ and $S_1 \cup S_2$. Repeating this process ensures that a maximal collection of *linearly independent* tight constraints form a *chain*. ◀

The sets S_i may not be equal since that would violate the linear independence of the corresponding constraints. Also, an earlier observation implies that $S_{i+1} \setminus S_i$ has to contain at least *one* fractional variable (and, bottoming out, this holds true for $S_1 \setminus \emptyset = S_1$ too). However since $x(S_{i+1}) - x(S_i)$ is an integer, there has to be at least 2 fractional variables in $S_{i+1} \setminus S_i$. But since the sets $S_{i+1} \setminus S_i$ are disjoint (for $1 \leqslant i \leqslant (\ell - 1)$ we thereby collect at least 2ℓ fractional variables. Since we started the argument with ℓ fractional variables, this implies that $\ell \geqslant 2\ell$, which is impossible since $\ell \geqslant 1$.

Thus, the constraint $\sum s(i)x_i \geqslant D$ has to be tight. But we can again decompose the tight constraints of the form $x(S) \leqslant r(S)$ as a chain of $(\ell - 1)$ constraints. A similar argument like the one above gives that there are at least $2(\ell - 1)$ fractional variables. Thus $\ell \geqslant 2(\ell - 1)$, and we get that $\ell \leqslant 2$. Since $\sum s(i)x_i \geqslant D$ is tight, this implies that there is precisely one tight constraint of the form $x(S) \leqslant r(S)$. Now we are back to the cardinality case, and we can mimic the proof of Theorem 2, and we thereby prove that LP_3 has at most 2 fractional variables.

Let the fractional variables be x_1 and x_2 (modulo renaming of variables). The tight constraint $\sum s(i)x_i \geqslant D$ simplifies to $s_1x_1 + s_2x_2 = D'$. Also we have that $x_1 + x_2 = k'$ for some *integral* k'. Since $0 < x_i < 1$ for all $i = 1, 2$, the only possibility is that $k' = 1$. This implies that D' is a convex combination of s_1 and s_2, and thus, one of these quantities is at least as large as D'; wlog, let $s_1 \geqslant D'$. Thus, we can raise the variable x_1 to 1 and *reduce* the variable x_2 to 0. This change in the variables x_1 and x_2 may potentially make the solution infeasible for the LP. To this end, let us consider the constraints in which the variables x_1 *or* x_2 appear. First, note that the constraint $\sum s(i)x_i \geqslant D$ is kept feasible, because of the choice of the fractional variable to raise. Now consider any matroid constraint $x(S) \leqslant r(S)$. If S contains both of items x_1 and x_2, then feasibility for this constraint is maintained (since $x_1 + x_2$ is not changed). Suppose that S contains only the item x_1. Then the constraint $x(S) \leqslant r(S)$ could not have been tight in the BFS solution x^*. This is because $r(S)$ is an integer whereas $x(S)$ contains just a single fractional variable and cannot be an integer. Thus, raising x_1 to 1 does not violate feasibility for this constraint. If S contains only the item x_2, the argument is simpler: the value of x_2 is lowered, and so feasibility is maintained for the packing constraint $x(S) \leqslant r(S)$.

Finally, given an $\epsilon > 0$, we prune items of the input by guessing the elements of *high-cost* (i.e. elements of cost $> \epsilon \cdot \mathrm{OPT}$) and modifying the input instance as in Theorem 2; this gives us the $(1+\epsilon)$-factor guarantee. ◀

8 Multiple Matroids

In this section, we consider the knapsack cover problem subject to *multiple matroid* constraints; call this the KCMM problem. In this problem, in addition to the knapsack cover constraint $\sum s(i)x_i \geqslant D$, we have $t \in \mathbb{N}$ matroid constraints. Given the matroids $\mathcal{M}_i$ for $1 \leqslant i \leqslant t$, let r_i denote the rank function for matroid $\mathcal{M}_i$. The IP for this problem is as follows:

$$\mathrm{IP}_4: \quad \begin{array}{ll} \min & \sum_i c(i) \cdot x_i \\ s.t. & \sum_i s(i) \cdot x_i \geqslant D \\ \forall S, t & x(S) \leqslant r_t(S) \\ \forall i & x_i \in \{0,1\} \end{array}$$

The feasibility problem for this IP is the *Matroid Intersection* problem. For $t \geqslant 3$, this is NP-hard: for instance, the Hamiltonian Circuit problem is a special case of the intersection of 3 matroids.

This leads to the hardness of approximation of the KCMM problem: for $t \geqslant 3$, it is impossible to achieve any factor approximation for the KCMM problem. We show that this obstacle can be overcome by considering *bicriteria* approximation algorithms. The main result of this section is that the KCMM problem allows good bicriteria approximations.

▶ **Theorem 5.** *There is a bicriteria approximation algorithm for the* KCMM *problem that outputs a solution S that satisfies the following properties: (i) $c(S) \leqslant c(\mathrm{OPT})$, (ii) S satisfies all the matroid constraints and (iii) $s(S) \geqslant \theta(\frac{D}{t})$.*

Proof. Consider the auxiliary IP parametrized by a quantity β. This IP is obtained by interchanging the roles of the objective and the knapsack cover constraint in IP_4.

$$\mathrm{IP}_5: \quad \begin{array}{ll} \max & \sum_i s(i) \cdot x_i \\ s.t. & \sum_i c(i) \cdot x_i \leqslant \beta \\ \forall S, t & x(S) \leqslant r_t(S) \\ \forall i & x_i \in \{0,1\} \end{array}$$

Thus we have converted the mixed packing covering problem IP_4 into a purely packing problem IP_5. The objective in the problem IP_5 is a submodular function (in fact a linear function) and the problem is to maximize this function subject to 1 knapsack (packing) constraint and t matroid constraints. There are efficient algorithms [10] that show $\mathcal{O}(t)$-factor approximations for this problem; denote this procedure by P.

We run the procedure P for various values of β in decreasing order; for each value of β, P gives an approximate solution to IP_5. We stop when the objective value of the solution returned by P is $D/\theta(t)$. Let this solution be S. We return S as the solution to IP_4 and the corresponding value of β is the objective value.

If there is a feasible solution F to IP_4 of objective value β', then F is a feasible solution to IP_5 (with $\beta = \beta'$) of objective value at least D. This is because F is a *feasible* solution to IP_4; so $\sum_{i \in F} s(i) \geqslant D$. So, procedure P on IP_5 with $\beta = \beta'$ produces a solution F' that has value $\sum_{i \in F} s(i) \geqslant D/\theta(t)$.

This proves the result.

◀

9 Discussion & Open Problems

In this paper we considered a mixed packing-covering problem called the KCM problem. We showed that it admits a PTAS. We note that the same result applies if we replace the matroid constraints by so-called *polymatroid* constraints (where the constraints are $x(S) \leqslant f(S)$ for an arbitrary *submodular* function f instead of the rank function).

However, we do not expect (even) constant factor approximations for most covering problems with matroid constraints. This is because the *feasibility problem* for such a covering problem with a matroid constraint may be NP-hard.

As an instance, let us consider the interval cover problem mentioned in Section 3. We will reduce the Vertex Cover problem to an interval cover instance, with a partition matroid. Let the input instance be $G = (V, E)$ and a number k; the decision problem is to find if G has a vertex cover of size at most k. The interval cover instance is constructed as follows. For every vertex $v \in V$, there is a *long* interval I_v; for different vertices v and v' the corresponding intervals are disjoint. Given an edge $e = (u, v)$, there are *two short* intervals corresponding to the edge $I_{e,u}$ and $I_{e,v}$. The span of $I_{e,u}$ for an edge e incident on the vertex u is contained within the interval I_u. The various intervals $I_{e,u}$ for different e's incident on u are all disjoint (and contained in I_u). The partition matroid constraints are as follows: out of all the intervals I_u ($u \in V$), we are allowed to pick at most k; and out of the intervals $I_{e,u}$ and $I_{e,v}$ (for $e = (u, v)$) we are allowed to pick at most 1. Suppose there is a vertex cover $S \subset V$ of size at k in G. Then our feasible solution for the interval cover problem will be constructed as follows: pick interval I_u for $u \in S$. Since S is a vertex cover, every edge e has one of its endpoints in S. Thus at least one of the short intervals $I_{e,u}$ or $I_{e,v}$ is not required, since the overlapping long interval I_u (or I_v) is picked. Thus the other short interval corresponding to e may be picked, while preserving the partition matroid constraint for $I_{e,u}$ and $I_{e,v}$. In the other direction, consider a feasible solution F for the interval cover instance. For any edge e, at most one of the short intervals $I_{e,u}$, $I_{e,v}$ may be in the feasible solution. Thus for the short interval that is absent, say, $I_{e,u}$, the long interval I_u has to be present in the solution. This means that the set of u's such that the interval I_u belongs to the solution F forms a vertex cover. Since feasibility stipulates at most k of the long intervals be selected, this means that we are able to extract a vertex cover of size at most k.

Despite feasibility being the principal obstacle for covering problems with matroid constraints, some open problems do remain. The principal open question is the following. In Section 8, we considered the case of t matroid constraints along with a knapsack cover constraint. However, note that for $t = 2$, the feasibility problem is in polynomial time (this is the Matroid Intersection problem for two matroids). So, a constant factor approximation algorithm is not ruled out for $t = 2$.

Our solution to the KCM problem involves solving multiple LP's. The knapsack cover (KC) problem has efficient greedy algorithms. Therefore, a natural question is whether the KCM problem has efficient greedy algorithms.

Our results do not rule out a FPTAS for the KCM problem. In fact, we are able to show FPTASes for the case of the KCM problem with a *partition matroid*. Is there a FPTAS for the KCM problem for arbitrary matroids?

Acknowledgements. We gratefully acknowledge discussions with Amit Kumar, Rajiv Gandhi, Guy Kortsarz and Yogish Sabharwal. We also gratefully acknowledge modifications suggested by the anonymous reviewers. The submission version contained only a 2-factor approximation for the KCM problem; a reviewer noted that a slight modification

of the same technique also yields a PTAS (that is now included as Theorem 1). The same reviewer also observed that the results essentially carry over to polymatroid constraints.

References

1 Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *FOCS*, pages 649–658, 2012.
2 Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.
3 Tim Carnes and David B. Shmoys. Primal-dual schema for capacitated covering problems. In *IPCO*, pages 288–302, 2008.
4 Robert D. Carr, Lisa Fleischer, Vitus J. Leung, and Cynthia A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *SODA*, pages 106–115, 2000.
5 Shaddin Dughmi. Submodular functions: Extensions, distributions, and algorithms. a survey. *CoRR*, abs/0912.0322, 2009.
6 Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, 2011.
7 Yuval Filmus and Justin Ward. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *FOCS*, pages 659–668, 2012.
8 Michel X. Goemans and José A. Soto. Symmetric submodular function minimization under hereditary family constraints. *CoRR*, abs/1007.2140, 2010.
9 Fabrizio Grandoni, Jochen Könemann, Alessandro Panconesi, and Mauro Sozio. A primal-dual bicriteria distributed algorithm for capacitated vertex cover. *SIAM J. Comput.*, 38(3):825–840, 2008.
10 Anupam Gupta, Viswanath Nagarajan, and R. Ravi. Thresholded covering algorithms for robust and max-min optimization. In *ICALP (1)*, pages 262–274, 2010.
11 Lap-Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization.* Cambridge University Press, New York, NY, USA, 1st edition, 2011.
12 Jan Karel Lenstra, David B. Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. In *FOCS*, pages 217–224, 1987.
13 A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency.* Springer, 2003.
14 Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM J. Comput.*, 40(6):1715–1737, 2011.
15 Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pages 67–74, 2008.
16 Jan Vondrák. Submodular functions and their applications, Plenary talk at SODA. 2013.