Operations Research Letters 50 (2022) 129–132

Contents lists available at ScienceDirect

Operations Research Letters

www.elsevier.com/locate/orl

Approximation algorithm for the 2-stage stochastic matroid base problem ☆

Takuro Fukunaga [a], R. Ravi [b], Oleksandr Rudenko [c], Ziye Tang [b,*]

[a] Faculty of Science and Engineering, Chuo University, Japan
[b] Tepper School of Business, Carnegie Mellon University, USA
[c] Department of Mathematical Sciences, Carnegie Mellon University, USA

ARTICLE INFO

Article history:
Received 14 January 2021
Received in revised form 10 January 2022
Accepted 12 January 2022
Available online 17 January 2022

Keywords:
Two-stage stochastic matroid base problem
Greedy algorithm
Submodular set cover

ABSTRACT

We consider the 2-stage stochastic matroid base problem, where an initial set of elements is bought by paying their first-stage deterministic cost and then extended to a matroid base in the second stage after the scenario realization. The second-stage costs are unrelated to the first-stage costs and represented explicitly via a polynomial number of scenarios. The objective is to pick the initial element set so as to minimize the expected total cost incurred. For a rank-r matroid under k scenarios we present an $O(\log r + \log k)$-approximation algorithm by adapting the greedy algorithm for finding a minimum-weight matroid base.

1. Introduction

We consider the 2-stage stochastic recourse model for the matroid base problem. In this problem, we are given the deterministic first-stage cost as well as the second-stage cost distribution of matroid elements. The matroid base is formed as follows: in the first stage, we pick a subset of independent elements by paying their first-stage costs. After the second-stage scenario is realized, we take the recourse action by augmenting the initial set with additional elements in order to form a matroid base. The goal is to select the first-stage set of elements so that the expected total cost is minimized when augmented with second-stage elements. Typically the recourse action entails making rapid decisions to the revealed scenario and is thus more costly than the initial action. Thus there is a trade-off between selecting the initial elements with cheaper costs under imprecise information, and deferring the selection to the revealed second-stage with higher costs.

The 2-stage stochastic recourse model has been considered for many combinatorial optimization problems. We refer the interested reader to the survey by Swamy and Shmoys [3] and references therein for an in-depth review. We note a special case of our problem, the 2-stage stochastic minimum spanning tree problem is considered by Dhamdhere, Ravi and Singh [1], where they present a randomized $O(\log n + \log k)$-approximation algorithm for n-node graph with k scenarios. They also showed that the problem is hard to approximate within a factor $\Omega(\log n)$ unless P = NP, which means that their approximation factor is hard to improve when $k = \Theta(n^c)$ for a constant c. Their algorithm is based on LP rounding which seems difficult to generalize to the matroid case. To the best of our knowledge, no approximation algorithm is known for the matroid case.

Our contribution. For a rank-r matroid with k scenarios in the second (recourse) stage, we present a deterministic $O(\log r + \log k)$-approximation algorithm by adapting the simple greedy algorithm. This matches the performance guarantee of the randomized algorithm by Dhamdhere et al. for the special case of the 2-stage stochastic minimum spanning tree problem.

Our paper is organized as follows: in Section 2 we formally define the 2-stage stochastic matroid base problem; in Section 3 we recall and prove some matroid properties needed to analyze the algorithm performance; in Section 4 we describe and analyze the performance of our greedy algorithm.

2. Problem definition

Let $M = (E, \mathcal{I})$ be a matroid where E denotes the element set and $\mathcal{I} \subseteq 2^E$ denotes the family of independent sets. Let r denote the matroid rank and $\mathcal{B}$ denote the family of matroid bases. Let $c_0(e)$ denote the first-stage deterministic cost for each element $e \in E$. In this work, we choose the *polynomial-scenario model* (termed in [3]) to represent the scenario distribution. More specif-

☆ This material is based upon research supported in part by the U. S. Air Force Office of Scientific Research under award number FA9550-20-1-008, and JST PRESTO under grant number JPMJPR1759.

* Corresponding author.
E-mail addresses: fukunaga@ise.chuo-u.ac.jp (T. Fukunaga), ravi@andrew.cmu.edu (R. Ravi), 4rudenko@gmail.com (O. Rudenko), ziyet@andrew.cmu.edu (Z. Tang).

https://doi.org/10.1016/j.orl.2022.01.004

ically, we are given k scenarios with the corresponding probability p_i (so that $p_1 + \ldots + p_k = 1$) and the cost functions $c_i(e)$ for each element $e \in E$ for $i = 1, \ldots, k$.

The matroid base is formed as follows: in the first stage, we pick a subset E_0 of independent elements by paying their first-stage costs. After the second-stage is realized as scenario i, we take the recourse action by augmenting E_0 with set E_i in order to form a base. The goal is to select the first-stage set of elements to E_0 in order to minimize the expected total cost incurred, i.e. $\sum_{e\in E_0} c^0(e) + \sum_{i=1}^k p_i \sum_{e\in E_i} c_i(e)$.

3. Properties of matroids

Roughly speaking, our algorithm employs a greedy strategy of adding elements to the first stage solution so that they improve the expected cost of completing the current solution in the second stage. In particular, we will choose greedily an element whose ratio of first-stage cost to the reduction in the expected completion cost is the smallest. To relate the cost of the elements added this way to that of the optimum we relate the ratio cost of the element to that of the optimum solution at this stage. To do this, we prove in this section that the reduction in the cost of an optimal base by contracting a subset of elements as a function of this set of elements is submodular.

Notation. We fix the cost function $c \in \mathbb{R}^E$ on a matroid $M = (E, \mathcal{I})$ throughout this section. For a subset $F \subset E$, let $c(F) := \sum_{e\in F} c(e)$. We let $c^*(M)$ denote the minimum cost of a base of M. For any $F \subseteq E$, M/F is the matroid obtained by contracting F, and T_F denotes a minimum cost base of M/F. We simply write $T_\varnothing$ as T. For ease of presentation, we assume that no two elements of E have the same cost. By this assumption, the minimum cost base of M/F is unique for any $F \subseteq E$.

The following property is well-known:

Lemma 1 *(Weak elimination property of circuits). Let C, C' be two distinct circuits of a matroid such that there exists an element $e \in C \cap C'$. Then, $C \cup C' \setminus \{e\}$ is dependent.*

The following theorem generalizes the matroid base exchange property:

Theorem 1 *([2]). Let A and B be two bases of a matroid. For any partition $A = A_1 \sqcup \cdots \sqcup A_k$, there exists a partition $B = B_1 \sqcup \cdots \sqcup B_k$ such that $(A \setminus A_i) \cup B_i$ is a base for each $i = 1, \ldots, k$.*

We now study the property of contracted matroids.

Lemma 2. *For any $X, Y \subseteq E$ with $X \subseteq Y$ where $Y \in \mathcal{I}$, we have $T_Y \subseteq T_X$*

Proof. We prove the lemma for $X = \varnothing$; if $X \neq \varnothing$, then it suffices to apply the lemma to M/X. Let $E \setminus Y = \{e_1, \ldots, e_m\}$. We suppose that they are sorted so that $c(e_1) < c(e_2) < \cdots < c(e_m)$. When the greedy algorithm computes T_Y of M/Y, it checks these elements in the increasing order of their indices, and each element is added to the solution when this addition does not violate the independence of the solution. Let T_i be the solution kept by the algorithm when e_i is going to be checked. $T_{i+1} = T_i \cup \{e_i\}$ if $T_i \cup \{e_i\} \cup Y \in \mathcal{I}$, and $T_{i+1} = T_i$ otherwise. Similarly, define T'_i as the solution kept by the algorithm applied to M when e_i is going to be checked.

We prove by induction that $T_i \subseteq T'_i \setminus Y$ holds for any i. When $i = 1$, the claim follows because $T_1 = \varnothing = T'_1 \setminus Y$. Let $i > 1$ and suppose that the condition holds for any $i' < i$. Suppose for a contradiction that $e_{i-1} \in T_i$ and $e_{i-1} \notin T'_i$. Then $T'_{i-1} \cup \{e_{i-1}\}$ has the circuit C including e_{i-1}. Because $e_{i-1} \in T_i$ implies that $T_{i-1} \cup \{e_{i-1}\}$ includes no circuit of M/Y, at least one element in $C \setminus (\{e_{i-1}\} \cup Y)$ is not included in T_{i-1}. We denote the set of such elements by K. For each element $e' \in K$, $T_{i-1} \cup \{e'\}$ has a circuit $C_{e'}$ of M/Y. Then, by the weak elimination property of circuits, $(C \setminus K) \cup (\bigcup_{e'\in K}(C_{e'} \setminus \{e'\}))$ is dependent. This contradicts the fact that $T_{i-1} \cup \{e_{i-1}\}$ is independent because $(C \setminus K) \cup (\bigcup_{e'\in K}(C_{e'} \setminus \{e'\})) \subseteq T_{i-1} \cup \{e_{i-1}\}$. □

Lemma 3. *Let $F \in \mathcal{I}$, and define $h\colon 2^F \to \mathbb{R}_+$ by $h(X) = c^*(M) - c^*(M/X)$ for each $X \subseteq F$. Then, h is monotone (nondecreasing) submodular.*

Proof. The monotonicity of h is immediate from the definition of h. Hence we prove the submodularity of h in this proof. Let $X, Y \subseteq F$. By Lemma 2, $T_{X\cup Y} \subseteq T_X \subseteq T_{X\cap Y} \subseteq T$ and $T_{X\cup Y} \subseteq T_Y \subseteq T_{X\cap Y}$. In this proof, we assume without loss of generality that the elements in $X \cup Y$ are not included in T; if $e \in X \cup Y$ is included in T, then we make a new element e' parallel to e, and replace e in X and in Y by e'. In a similar vein, if $X \cap Y \neq \varnothing$, then it suffices to consider $M/(X \cap Y)$ instead of M. Therefore, we assume that $X \cap Y = \varnothing$, and hence $h(X \cap Y) = 0$. In the following, we prove $h(X) + h(Y) \geq h(X \cup Y)$.

Each of $X \cup Y$ and $T \setminus T_{X\cup Y}$ includes a base of $M/T_{X\cup Y}$. Then, by applying Theorem 1 to $X \cup Y$ and $T \setminus T_{X\cup Y}$, there exists a partition $T \setminus T_{X\cup Y} = X' \sqcup Y'$ such that each of $X' \cup Y$ and $X \cup Y'$ includes a base of $M/T_{X\cup Y}$. By the definition of $T_{X\cup Y}$, each of $T_{X\cup Y} \cup X' \cup Y$ and $T_{X\cup Y} \cup X \cup Y'$ includes a base of M.

Observe that $h(X \cup Y) = c(T) - c(T_{X\cup Y}) = c(X') + c(Y')$. Since $T_{X\cup Y} \cup Y' = T \setminus X'$ includes a base of M/X, we have $c(T_X) \leq c(T) - c(X')$. Thus, $h(X) = c(T) - c(T_X) \geq c(X')$. Similarly, since $T_{X\cup Y} \cup X' = T \setminus Y'$ includes a base of M/Y, we have $c(T_Y) \leq c(T) - c(Y')$. This implies $h(Y) = c(T) - c(T_Y) \geq c(Y')$. Combining these inequalities gives $h(X) + h(Y) \geq h(X \cup Y)$. □

Based on the submodularity of h, we can prove the following lemma, which is the key to analyzing the performance of our greedy algorithm in Section 4.

Lemma 4. *For any independent set $F \in \mathcal{I}$ of $M = (E, \mathcal{I})$, there exists an element $e \in F$ such that*

$$\frac{c_0(e)}{\sum_{i=1}^k p_i(c_i^*(M) - c_i^*(M/e))} \leq \frac{c_0(F)}{\sum_{i=1}^k p_i(c_i^*(M) - c_i^*(M/F))}. \quad (1)$$

Proof. Let $F = \{e_1, e_2, \ldots, e_l\}$, and let $e \in \arg\min_{e'\in F} c_0(e')/\sum_{i=1}^k p_i(c_i^*(M) - c_i^*(M/e'))$. We prove that this e satisfies (1).

For any $E' \subseteq F$, let $h(E') = c_i^*(M) - c_i^*(M/E')$. Then, for any $i = 1, 2, \ldots, k$,

$$\begin{aligned} & c_i^*(M) - c_i^*(M/F) \\ &= \sum_{j=1}^{l} \left(c_i^*\left(M/\{e_1, e_2, \ldots, e_{j-1}\}\right) - c_i^*\left(M/\{e_1, e_2, \ldots, e_j\}\right)\right) \\ &= \sum_{j=1}^{l} \left(h\left(\{e_1, e_2, \ldots, e_j\}\right) - h\left(\{e_1, e_2, \ldots, e_{j-1}\}\right)\right) \end{aligned}$$

holds. By the submodularity of h proven in Lemma 3, the right-hand side is at most

$$\sum_{j=1}^{l} \left(h(\{e_j\}) - h(\varnothing)\right) = \sum_{j=1}^{l} \left(c_i^*(M) - c_i^*(M/e_j)\right).$$

Therefore,

$$\frac{c_0(e)}{\sum_{i=1}^{k} p_i(c_i^*(M) - c_i^*(M/e))}$$
$$\leq \frac{\sum_{j=1}^{l} c_0(e_j)}{\sum_{j=1}^{l}\sum_{i=1}^{k} p_i\left(c_i^*(M) - c_i^*(M/e_j)\right)}$$
$$\leq \frac{c_0(T)}{\sum_{i=1}^{k} p_i(c_i^*(M) - c_i^*(M/F))}. \quad \square$$

4. Algorithm

4.1. Overview

Recall r is the matroid rank and k is the number of scenarios. Our algorithm consists of two steps. First we perform a preprocessing step so that the second stage cost in the worst case is at most rk times the optimal second stage cost. This step is achieved by choosing certain elements in the first stage so that we can forbid the use of costly elements in each scenario in the second stage while still keeping the problem feasible. We employ the greedy algorithm for the submodular set cover from [4] to show that the elements chosen in this step have a bounded cost. In the second step, we run a simple greedy algorithm to pick elements until the second stage cost becomes comparable to the optimal second stage cost. The restriction that the worst-case second stage cost is at most rk times the optimal allows us to bound the cost of the first stage; the first stage aims to reduce the second stage costs from up to rk times optimal to a constant factor of optimal, by paying roughly the cost of an optimal solution for reducing the expected second-stage costs by a constant factor.

4.2. Preprocessing

In the preprocessing step we aim to solve the following problem.

Definition 1 *(Matroid base extension).* Let $M = (E, \mathcal{I})$ be a matroid, and let $F_1, \ldots, F_k \subseteq E$ be given subsets of E. We are also given a weight $w(e)$ for each $e \in E$. Then, we want to find a minimum weight subset $F \subseteq E$ such that $F \cup F_i$ includes a base of M for each $i = 1, 2, \ldots, k$.

In the following, we present an $O(\log k)$-approximation algorithm based on a reduction to the submodular set cover problem.

Definition 2 *(Submodular set cover).* In the submodular set cover problem, we are given a monotone (nondecreasing) submodular function f on a ground set I and a nonnegative weight of each element in I. Then the problem requires finding a minimum weight subset I' of I such that $f(I') = f(I)$.

Wolsey [4] shows that there exists a greedy algorithm which achieves an approximation ratio of $O(\log \max_{i \in I} f(\{i\}))$ for the submodular set cover problem.

Theorem 2. *There exists an $O(\log k)$-approximation algorithm for the matroid base extension problem.*

Proof. We claim that the matroid base extension problem can be reduced to an instance of the submodular set cover problem. Define a function $f\colon 2^E \to \mathbb{R}_+$ by $f(F) = \sum_{i=1}^{k} r_{M/F_i}(F)$ for each $F \subseteq E$, where r_{M/F_i} is the rank function of the matroid M/F_i. Then F is feasible to the matroid base extension problem if and only if $f(F) = f(E)$. Note that $\max_{e \in E} f(\{e\}) \leq k$, and thus this reduction gives an $O(\log k)$-approximation for the matroid base extension problem. $\square$

Algorithm 1 Preprocess$(M, \{c_i\}_{i=1}^k, b)$.

Input: M: matroid; c_i: cost function for scenario i; b: estimate of expected second-stage cost
Output: A subset of elements A
1: Let $F_i := \{e \in E : p_i c_i(e) \leq b\}$ for $i = 1, \ldots, k$
2: Let A be the output of the $O(\log k)$-approximation algorithm for the matroid base extension problem with above F_i and weight function c_0 ▷ Theorem 2

Algorithm 2 $O(\log r + \log k)$-approximation algorithm.

Input: M: matroid; c_i: cost function for scenario $i = 0, 1, \ldots k$
Output: $B \subset E$
1: Guess a and b such that $a/2 \leq c_0(E_0^*) \leq a$ and $b/2 < \sum_{i=1}^{k} p_i c_i^*(M/E_0^*) \leq b$ for some optimal solution E_0^*.
2: $A \leftarrow$ Preprocess$(M, \{c_i\}_{i=1}^k, b)$ ▷ Algorithm 1
3: Set $M' := M/A$, $j := 0$ and $T_0 := \varnothing$. For any $F \subset E \backslash A$, let $g(F) := \sum_{i=1}^{k} p_i c_i^*(M'/F)$.
4: **while** $g(T_j) - b > \frac{g(T_0)-b}{rk}$ and $\exists e$ with $c_0(e) \leq a$ in M'/T_j **do**
5: In M'/T_j, find an element e_j such that $c_0(e_j) \leq a$ and it minimizes $c_0(e_j)/(g(T_j) - g(T_j \cup \{e_j\}))$.
6: Let $T_{j+1} = T_j \cup \{e_j\}$. Increase j by 1
7: Let B be an arbitrary maximal independent set of $A \cup T_j$

Algorithm 1 outlines the preprocessing step. Let E_0^* be the set of elements chosen by an optimal solution in the first stage. Then the second stage cost of the optimal solution is $\sum_{i=1}^{k} p_i c_i^*(M/E_0^*)$. In the preprocessing, we guess this value within a constant factor. We let b be the guessed value, and we assume that $b/2 < \sum_{i=1}^{k} p_i c_i^*(M/E_0^*) \leq b$. For each $i \in [k]$, we define $F_i = \{e \in E\colon p_i c_i(e) \leq b\}$. Then, we solve the matroid base extension problem with weights $w := c_0$ by the $O(\log k)$-approximation algorithm. Let A be the obtained solution. This solution satisfies the following property.

Lemma 5. *$c_0(A) = O(\log k) \cdot c_0(E_0^*)$ and $\sum_{i=1}^{k} p_i c_i^*(M/A) \leq rkb$.*

Proof. When the ith scenario is realized, no element $e \in E \setminus F_i$ is chosen by the optimal solution in the second stage since otherwise, the second stage cost of the solution is at least $p_i c_i(e) > b$. Hence $E_0^* \cup F_i$ includes a base of M for each $i \in [k]$. This means that E_0^* is a feasible solution for the matroid base extension problem, and thus $c_0(A) = O(\log k) \cdot c_0(E_0^*)$.

Since A is feasible for the matroid base extension problem, F_i includes a base of M/A for each $i \in [k]$. Since each element of F_i has cost of at most b/p_i, the cost of the base is at most rb/p_i. Hence $\sum_{i=1}^{k} p_i c_i^*(M/A) \leq \sum_{i=1}^{k} p_i \cdot (rb/p_i) = rkb$. $\square$

4.3. Main part

Our algorithm is described in Algorithm 2. This algorithm outputs the set of elements chosen in the first stage. The guesses of a and b in line 1 can be enumerated in polynomial time by using binary search over the range of values of the optimal solution cost.

Let $l + 1$ be the number of iterations in Algorithm 2 (thus the output of the algorithm is a maximal independent set of $A \cup T_{l+1}$).

Lemma 6. *$c_0(A \cup T_{l+1}) = O(\log r + \log k) \cdot c_0(E_0^*)$.*

Proof. By applying Lemma 4 where we set M to be M'/T_j and F to be E_0^*, we have that $\forall j = 0, \ldots, l$,

$$\frac{c_0(e_j)}{g(T_j) - g(T_{j+1})} \leq \frac{c_0(E_0^*)}{g(T_j) - g(T_j \cup E_0^*)}, \tag{2}$$

where g is defined in line 3 of Algorithm 2. As a result, we can relate $g(T_{j+1})$ to $g(T_j)$ as follows:

$$\begin{aligned} g(T_{j+1}) - b &= g(T_j) - b - (g(T_j) - g(T_{j+1})) \\ &\le g(T_j) - b - \frac{c_0(e_j)}{c_0(E_0^*)} \cdot (g(T_j) - g(T_j \cup E_0^*)) \\ &\le \left(1 - \frac{c_0(e_j)}{c_0(E_0^*)}\right)\left(g(T_j) - b\right) \\ &\le e^{-c_0(e_j)/c_0(E_0^*)}\left(g(T_j) - b\right), \end{aligned}$$

where the second inequality follows from $g(T_j \cup E_0^*) \le g(E_0^*) = \sum_{i=1}^k p_i c_i^*(M'/E_0^*) \le \sum_{i=1}^k p_i c_i^*(M/E_0^*) \le b$. Therefore,

$$g(T_l) - b \le e^{-\sum_{j=0}^{l-1} c_0(e_j)/c_0(E_0^*)}\left(g(T_0) - b\right).$$

Notice that $g(T_l) - b > (g(T_0) - b)/(rk)$ holds since the algorithm did not terminate in the l-th iteration. Hence, by taking the log and arranging the terms in the above inequality, we have $\sum_{j=0}^{l-1} c_0(e_j) < (\log r + \log k) \cdot c_0(E_0^*)$.

Notice that $T_{l+1} = \{e_0, e_1, \ldots, e_l\}$. We have $c_0(e_l) \le a \le c_0(E_0^*)$ by the definition of e_l and a. Hence, $c_0(T_{l+1}) \le (1 + \log r + \log k) \cdot c_0(E_0^*)$. Moreover, $c_0(A) = O(\log k) \cdot c_0(E_0^*)$ by Lemma 5. Therefore, the lemma is proved. □

Lemma 7. $\sum_{i=1}^k p_i c_i^*(M/(A \cup T_{l+1})) \le 4 \sum_{i=1}^k p_i c_i^*(M/E_0^*)$.

Proof. If line 4 becomes true in Algorithm 2, since a is an upper bound on $c_0(E_0^*)$, we have $E_0^* \subset T_{l+1} \cup A$. Therefore the lemma is trivially true.

Otherwise, by the termination condition of the while loop in Algorithm 2,

$$\sum_{i=1}^k p_i c_i^*(M'/T_{l+1}) \le \left(1 - \frac{1}{rk}\right) b + \frac{1}{rk} \sum_{i=1}^k p_i c_i^*(M'/T_0).$$

Moreover, $\sum_{i=1}^k p_i c_i^*(M'/T_0) = \sum_{i=1}^k p_i c_i^*(M') = \sum_{i=1}^k p_i c_i^*(M/A) \le rkb$ holds, where the last inequality follows from Lemma 5. Therefore,

$$\sum_{i=1}^k p_i c_i^*(M'/T_{l+1}) \le \left(1 - \frac{1}{rk}\right) b + b \le 4 \sum_{i=1}^k p_i c_i^*(M/E_0^*). \quad \square$$

Theorem 3. *Algorithm 2 achieves an approximation factor within* $O(\log r + \log k)$.

Proof. The cost of the solution output by Algorithm 2 is $c_0(B) + \sum_{i=1}^k p_i c_i^*(M/B)$. Since B is a maximal independent set of $A \cup T_{l+1}$, $c_0(B) \le c_0(A \cup T_{l+1})$ and $M/B = M/(A \cup T_{l+1})$. Thus the cost of the solution is at most $c_0(A \cup T_{l+1}) + \sum_{i=1}^k p_i c_i^*(M/(A \cup T_{l+1}))$. Moreover, this is at most $O(\log r + \log k) \cdot c_0(E_0^*) + 4 \sum_{i=1}^k p_i c_i^*(M/E_0^*)$ by Lemmas 6 and 7. Since the optimal objective value is $c_0(E_0^*) + \sum_{i=1}^k p_i c_i^*(M/E_0^*)$, this means that the approximation factor of Algorithm 2 is $O(\log r + \log k)$. □

References

[1] K. Dhamdhere, R. Ravi, M. Singh, On two-stage stochastic minimum spanning trees, in: International Conference on Integer Programming and Combinatorial Optimization, Springer, 2005, pp. 321–334.

[2] M. Lasoń, List coloring of matroids and base exchange properties, Eur. J. Comb. 49 (2015) 265–268.

[3] C. Swamy, D.B. Shmoys, Approximation algorithms for 2-stage stochastic optimization problems, ACM SIGACT News 37 (1) (2006) 33–46.

[4] L.A. Wolsey, An analysis of the greedy algorithm for the submodular set covering problem, Combinatorica 2 (4) (1982) 385–393.