A note on set union knapsack problem

Ashwin Arulselvan

Institut für Mathematik, Technische Universität Berlin, Germany, arulsel@math.tu-berlin.de.

Abstract

Recently, Khuller, Moss and Naor presented a greedy algorithm for the budgeted maximum coverage problem. In this note, we observe that this algorithm also approximates a special case of set-union knapsack problem within a constant factor. In the special case, an element is a member of less than a constant number of subsets. This guarantee naturally extends to densest k-subgraph problem on graphs of bounded degree.

Keywords: approximation, densest k-subgraph, set-union knapsack, greedy

1. Introduction

The Set Union Knapsack Problem (SUKP) comprises of a set of elements $U = \{1, \ldots, n\}$ and a set of items $\mathcal{S} = \{1, \ldots, m\}$. Each item, $i = 1, \ldots, m$, correspond to a subset of elements and we denote it by S_i. Each item has a nonnegative profit given by $p : \mathcal{S} \rightarrow \mathbb{R}_+$ and each element has a nonnegative weight given by $w : U \rightarrow \mathbb{R}_+$. For a subset $A \subseteq \mathcal{S}$, we define the weighted union of set A as $W(A) = \sum_{e \in \cup_{i \in A} S_i} w_e$ and $P(A) = \sum_{i \in A} p_i$. We want to find a subset of items $\mathcal{S}^* \subseteq \mathcal{S}$ such that $P(\mathcal{S}^*)$ is maximized and $W(\mathcal{S}^*) \leq B$. Goldschmidt et. al. [2] studied the problem and presented a dynamic program running in exponential time to solve the problem exactly. In the densest k-subhypergraph problem [3], we are given a hypergraph $H(V, E)$ and we have to determine a set of k nodes such that the subhypergraph induced by this set has maximum number of hyperedges. SUKP reduces to the densest k-subhypergraph problem (DkH), when we have unit weights and unit profits, with the elements and items corresponding to the nodes and hyperedges respectively and the budget being k. Recently [3] it has been shown that densest k-subhypergraph problem cannot be approximated within factor of $2^{(\log n)^\delta}$, for some $\delta > 0$, unless $3SAT \in DTIME(2^{n^{\frac{3}{4}+\epsilon}})$. For the special case, where we have the item size equal to exactly 2, we have the densest k-subgraph problem (DkH on graphs). The best known algorithm provides an approximation factor of $O(\min\{n^\delta, n/k\})$, for any $\delta < 1/3$ [1].

We present a greedy algorithm for the SUKP with the additional restriction that the number of items in which an element is present is bounded by a constant d. We will show that the algorithm provides a $(1 - e^{-\frac{1}{d}})$ approximation. This factor naturally extends to densest k-subgraph problem where the input graph has a bounded degree. To the best of our knowledge, the only known result about this case is that it is NP-hard, even with the maximum degree $d \leq 3$ [1]. The algorithm and the analysis directly follows from the work of Khuller, Moss and Naor [4] for the budgeted maximum coverage problem (BMCP). Hence, the novelty of the note lies in the new observations made about some existing open problems and not on

the algorithm or its analysis. The BMCP and SUKP, despite the similarities, are significantly different problems. SUKP in the general case is hard to approximate.

2. Algorithm and Analysis

We need a few more notations before we present the algorithm. We define d_e as the frequency of an element e, i.e, the number of items in which element e is present. So, we have $\max_{e \in U} d_e \leq d$. For an item i, we denote the profit of item i by p_i and define $W_i' = \sum_{e \in S_i} \frac{w_e}{d_e}$, where w_e is the weight of element e.

We consider all possible subsets of items of cardinality 2, whose weighted union is within the budget B. We augment each of these subsets with items (not in the subset) one by one in the decreasing order of the ratio $\frac{p_i}{W_i'}$, if its inclusion does not violate the budget B. We then choose of the best of these augmented sets as A. Afterwards, we pick item with the highest profit as $S_{\max}$. The best of A and $S_{\max}$, in terms of their profits, is returned as the solution. As a side note, we point out that the items could be considered in the increasing order of the the ratio of sum of weights of elements in the item that are yet to be picked to its profit and this will ensure the same guarantee on the approximation factor, but it is easier to follow the analysis of Khuller et.al. with the one presented.

We write the greedy augmentation as a subroutine GREEDY for the sake of presenting the analysis.

GREEDY(G,U)

1: **while** $U \neq \emptyset$ **do**
2: Choose $i \in U$ with the highest value $\frac{p_i}{W_i'}$
3: **if** $W(G \cup \{i\}) \leq B$ **then**
4: $G = G \cup \{i\}$
5: **end if**
6: $U = U \backslash \{i\}$
7: **end while**

A-SUKP(G)

1: Let $S_{\max}$ be the set with the highest profit
2: $A = \emptyset$
3: **for all** $G \subset \mathcal{S}$ such that $|G| = 2, W(G) \leq B$ **do**
4: G = GREEDY(G, $\mathcal{S} \backslash G$)
5: $A = \arg\max\{P(G), P(A)\}$
6: **end for**
7: Return $\arg\max\{P(A), P(S_{\max})\}$

We present the analysis as in [4] for the BMCP. Let OPT be the set of items picked in the optimal solution. Let us order the items in OPT in the non-increasing order of their profits. Let Y be the first two items in this order. Let Y' be the set of items added by the greedy subroutine to Y. Let r be the first iteration in the greedy subroutine, where we consider an item in OPT but we do not add it to G as it exceeds the budget (the IF condition fails in Step 3). Let us assume that a total of ℓ items are added to the solution set G until this point, i.e., the set Y' has ℓ items. We will consider the items in the order, $i = 1, \ldots, \ell, \ell+1$, in which they are added by the greedy subroutine with $\ell+1$ being the first item in OPT that was considered

and rejected. Let k_i be the iteration in the greedy subroutine where we have added item i (or rejected item $i = \ell + 1$) and Y'_i be the set of items added to Y so far.

The proofs of the following lemmas are only sketched as the arguments are straight from [4]

Lemma 2.1 ([4]). *For each $i = 1, \ldots, \ell + 1$, we have*

$$p_i \geq \frac{W'_i}{B} \left(P(OPT \backslash Y) - \sum_{j=1}^{i-1} p_i \right).$$

Proof Sketch:. For all $j \in OPT \backslash \{Y \cup A_{i-1}\}$ we have that

$$\frac{p_j}{W'_j} \leq \frac{p_i}{W'_i}$$

and we also have

$$W(OPT \backslash \{Y \cup A_{i-1}\}) \leq B.$$

Hence,

$$\sum_{j \in OPT \backslash \{Y \cup A_{i-1}\}} p_j \leq \sum_{j \in OPT \backslash \{Y \cup A_{i-1}\}} W'_j \left(\frac{p_i}{W'_i} \right) \leq W(OPT \backslash \{Y \cup A_{i-1}\}) \left(\frac{p_i}{W'_i} \right) \leq B \left(\frac{p_i}{W'_i} \right).$$

□

Lemma 2.2 ([4]). *For each $i = 1, \ldots, \ell + 1$, we have*

$$\sum_{j=1}^{i} p_i \geq \left(1 - \prod_{j=1}^{i} \left(1 - \frac{W'_j}{B} \right) \right) P(OPT \backslash Y).$$

Proof Sketch:. We will prove it through induction. Clearly, it is true for $i = 1$. Let us assume it is true for some $i - 1$ and we will prove it for the case i. Then,

$$\begin{aligned}
\sum_{j=1}^{i} p_j &= \sum_{j=1}^{i-1} p_j + p_i \\
&\geq \sum_{j=1}^{i-1} p_j + \frac{W'_i}{B} \left(P(OPT \backslash Y) - \sum_{j=1}^{i-1} p_j \right) \\
&= \left(1 - \frac{W'_i}{B} \right) \sum_{j=1}^{i-1} p_j + \frac{W'}{B} P(OPT \backslash Y) \\
&\geq \left(1 - \prod_{j=1}^{i} \left(1 - \frac{W'_j}{B} \right) \right) P(OPT \backslash Y).
\end{aligned}$$

The first inequality follows from Lemma 2.1 and the second from the induction hypothesis. □

Claim 2.3 ([4]). *If $a_1, \ldots, a_n \in \mathbb{R}_+$ and $\sum_{i=1}^{n} a_i = \alpha A$, then the function*

$$\left(1 - \prod_{i=1}^{n}\left(1 - \frac{a_i}{A}\right)\right)$$

attains a minimum value of $(1 - (1 - \frac{\alpha}{A})^n)$.

Theorem 2.4 ([4]). *The greedy algorithm is within a factor of $(1 - e^{-\frac{1}{d}})$ from the optimal.*

Proof. We define $\widehat{W}$ as follows:

$$\widehat{W} = \sum_{i=1}^{\ell+1} W_i' = \sum_{i=1}^{\ell+1} \sum_{e \in S_i} \frac{w_e}{d_e}.$$

Multiplying by d throughout, we get

$$d\widehat{W} = \sum_{i=1}^{\ell+1} dW_i' = \sum_{i=1}^{\ell+1} \sum_{e \in S_i} d\frac{w_e}{d_e} \geq \sum_{i=1}^{\ell+1} \sum_{e \in S_i} w_e \geq W(\bigcup_{i=1}^{\ell+1} S_i) \geq B.$$

From Lemma 2.2, we have the following,

$$\begin{aligned}
\sum_{j=1}^{\ell+1} p_i &\geq \left(1 - \prod_{j=1}^{\ell+1}\left(1 - \frac{W_j'}{B}\right)\right) P(OPT \backslash Y) \\
&\geq \left(1 - \prod_{j=1}^{\ell+1}\left(1 - \frac{W_j'}{d\widehat{W}}\right)\right) P(OPT \backslash Y) \\
&\geq \left(1 - \left(1 - \frac{1}{d(\ell+1)}\right)^{\ell+1}\right) P(OPT \backslash Y) \\
&\geq (1 - e^{-\frac{1}{d}}) P(OPT \backslash Y).
\end{aligned}$$

The second inequality follows from claim 2.3. Now we have

$$\sum_{i=1}^{\ell} p_i + P(S_{\max}) \geq \sum_{i=0}^{\ell+1} p_i \geq (1 - e^{-\frac{1}{d}}) P(OPT \backslash Y).$$

The profit of the $(\ell+1)$-st item that was rejected must be less than the profit of the two items in Y. So we have $p_{\ell+1} \leq \frac{1}{2} P(Y)$.

Now we have

$$\begin{aligned}
P(A) &\geq P(Y) + P(Y') \\
&= P(Y) + \sum_{i=1}^{\ell} p_i \\
&\geq P(Y) + (1 - e^{-\frac{1}{d}}) P(OPT \backslash Y) - p_{l+1} \\
&\geq \frac{1}{2} P(Y) + (1 - e^{-\frac{1}{d}}) P(OPT \backslash Y) \\
&\geq (1 - e^{-\frac{1}{d}})(P(Y) + P(OPT \backslash Y) \\
&= (1 - e^{-\frac{1}{d}}) P(OPT).
\end{aligned}$$

The last inequality is true for all $d \geq 2$. For $d = 1$, the SUKP is just the regular knapsack problem.

□

We would like to make a few remarks before we conclude. If we set up the natural independence system for the problem (or its dual) with respect to subsets of items whose weighted union is within the budget (whose complement exceeds some profit), the special case under consideration does not correspond to a bounded rank quotient, making the case studied non-trivial.

We also would like to point out the fact that

$$\lim_{d\to\infty} \left| (1 - e^{-\frac{1}{d}}) - \frac{1}{d} \right| = 0.$$

This gives evidence to the claim that the true approximation factor is actually the constant $\frac{1}{d}$. On the negative side, consider a star S_{d-1}, with d edges. Let us replace each leaf node by a complete graph, K_d. If we have to pick d nodes, the greedy algorithm would pick the internal node and the leaf nodes, which induces a subgraph with exactly d edges. However, the optimal solution is $\frac{d(d-1)}{2}$. So, we cannot achieve a factor better than $\frac{2}{(d-1)}$, but the tightness of the analysis is still open.

We could easily note that the greedy algorithm is mimicking a primal-dual type algorithm, so we could hope to obtain such a factor by using a linear program for the analysis. For instance, let us consider the following simple algorithm to obtain a $\frac{1}{2d}$ approximation factor. $\mathcal{P}_1$ is the natural LP relaxation for SUKP and $\mathcal{P}_2$ is a relaxed version of it.

$$\begin{aligned}
\mathcal{P}_1: \quad & \max \sum_{i=1}^{m} p_i x_i \\
& \text{s.t } \sum_{j=1}^{n} w_j y_j \leq B \\
& y_j \geq x_i, \forall j \in S_i, \forall i = 1 \ldots m \\
& x_i, y_j \geq 0
\end{aligned}
\qquad\qquad
\begin{aligned}
\mathcal{P}_2: \quad & \max \sum_{i=1}^{m} p_i x_i \\
& \text{s.t } \sum_{i=1}^{m} W_i' x_i \leq \frac{B}{d} \quad (\text{or } \sum_{i=1}^{m} W(S_i) x_i \leq B) \\
& x_i \geq 0
\end{aligned}$$

x_i and y_j are the relaxed binary variables corresponding to item i and element j, indicating whether they are picked in the solution or not. We can readily observe that every integral feasible solution to $\mathcal{P}_2$ is a feasible solution to SUKP. We also have that every feasible solution to the $\mathcal{P}_1$ is feasible to $\mathcal{P}_2$, when we scale it down by d. This is true because we are overestimating (implicitly) the weights of the elements in $\mathcal{P}_2$ by a factor of at most d. We also know that the solution from the greedy algorithm for the maximum value knapsack problem can be bounded by its dual LP value within a factor of 2. These two facts give us a $\frac{1}{2d}$-approximation for the special case of SUKP by just using the LP.

References

[1] U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, 29:2001, 1999.

[2] O. Goldschmidt, D. Nehme, and G. Yu. Note: On the set-union knapsack problem. *Naval Research Logistics (NRL)*, 41(6):833–842, 1994.

[3] M. T. Hajiaghayi, K. Jain, K. Konwar, L. C. Lau, I. I. Măndoiu, A. Russell, A. Shvartsman, and V. V. Vazirani. The minimum k-colored subgraph problem in haplotyping and DNA primer selection. In *Proc. Int. Workshop on Bioinformatics Research and Applications (IWBRA)*, 2006.

[4] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70:39–45, 1999.