Approximating Minimum Spanning Sets in Hypergraphs and Polymatroids

Gregor Baudis, Clemens Gröpl, Stefan Hougardy, Till Nierhoff*, and Hans Jürgen Prömel **

Institut für Informatik, Humboldt-Universität zu Berlin, 10099 Berlin, Germany

Abstract. We present a new analysis of the greedy algorithm for the problem of finding a minimum spanning subset in k-polymatroids. This algorithm has a performance ratio of approximately $\ln k$, which is best possible for large k. A consequence of this algorithm is a polynomial time approximation algorithm with approximation ratio $\ln k$ for finding minimum weight spanning subhypergraphs in $(k+1)$-restricted hypergraphs. This generalization of the well-known set cover problem naturally arises when computing Steiner minimum trees. Other applications of the algorithm include the rigidity problem in statics.

1 Introduction

Given a set S with n elements and a collection of weighted subsets of S, the set cover problem asks for a minimum weight subcollection of these subsets that cover all elements of S. This problem is known to be NP-hard and a result of Feige [5] indicates that set cover cannot be approximated in polynomial time better than $\ln n$. Chvátal [3] showed that a certain variant of the greedy algorithm yields an $H(k)$-approximation algorithm for k-set cover, i.e., the set cover problem where the size of all subsets is bounded by k. Here, $H(k)$ denotes the k-th harmonic number, i.e., $H(k) = \sum_{i=1}^{k} 1/i$. The result of Feige and the fact that $H(n) = \ln n + O(1)$ imply that the greedy algorithm is the best possible polynomial time approximation algorithm for set cover. A straightforward reduction from the minimum vertex cover problem with bounded degrees [14] shows that k-set cover is APX-complete for every constant $k \geq 3$, while 2-set cover (also known as the edge cover problem) can be solved exactly in polynomial time.

The set cover problem can also be viewed as a problem in hypergraphs: Given a hypergraph $\mathcal{H}$ with weights on its edges, find a minimum weight subhypergraph of $\mathcal{H}$ that covers all vertices of $\mathcal{H}$. The *minimum spanning sub(hyper)graph* problem (MSS) is a variant of this problem, where the subhypergraph is required to be connected. A hypergraph is called *k-restricted* if the size of all hyperedges is bounded by k. In this

* Corresponding author, email: nierhoff@informatik.hu-berlin.de

** The project "Approximative, randomisierte und probabilistische Algorithmen für Kombinatorische Optimierungsprobleme" is supported by the Deutsche Forschungsgemeinschaft, no. Pr 296/6-1. This work was also supported in part by project no. Pr 296/4-2.

paper, we present a new analysis for the greedy $H(k-1)$-approximation algorithm for the minimum spanning subhypergraph problem in k-restricted hypergraphs (k-MSS). Note that our analysis naturally generalizes Chvátal's analysis for the k-set cover problem, which is a special case of $(k+1)$-MSS: An optimum solution to k-set cover for a hypergraph $\mathcal{H}$ is an optimum solution to $(k+1)$-MSS in the hypergraph that is obtained from $\mathcal{H}$ by adding the same new vertex to all hyperedges in $\mathcal{H}$.

There is a close connection between the k-MSS problem and the problem of computing Steiner minimum trees: Given a graph $G = (V, E)$ with terminal set $S \subset V$ a k-restricted hypergraph on the vertex set S can be generated by taking as hyperedges all subsets of S of size at most k and weighting such a subset with the length of a Steiner minimum tree for this subset. Note that for constant k, these weights can be computed in polynomial time. A minimum spanning subgraph in that k-restricted hypergraph yields a Steiner tree in G. It has been shown [4, 2] that for sufficiently large k the length of this Steiner tree comes arbitrarily close to the length of a Steiner minimum tree. Therefore, good approximation algorithms to the k-MSS problem yield also good approximation algorithms for the Steiner tree problem. All recent approximation algorithms for solving the Steiner tree problem [17, 19, 1, 20, 15, 9, 8, 16] are based on this approach.

The approximation algorithm for k-MSS uses a similar greedy strategy as Chvátal's algorithm for k-set cover. However, the analysis needs some new idea. The main reason for this is that the connectedness of the subhypergraph – as required in a solution to k-MSS – is a global property that cannot be decided locally. By starting with an empty subhypergraph and greedily adding hyperedges, the number of components decreases by a certain amount. The reduction of the number of components is a map defined for each set of hyperedges. This map has a well known combinatorial property: It is the dimension function of a *polymatroid*. This shows that the k-MSS problem can be formulated as the problem of finding a minimum spanning set in a certain polymatroid. The general concept of polymatroids provides a natural way to shape the arguments needed for the analysis of the greedy algorithm. An analysis based on linear programming duality was given by Wolsey [18]. Our new analysis relies on the fact that each polymatroid can be represented by a system of subsets of an appropriately chosen matroid. The above mentioned result of Feige [5] implies that the approximation ratio of the greedy algorithm for spanning sets in k-polymatroids and for k-MSS is best possible for large k. See also Fujito's survey article [6] for a related LP-based dual greedy algorithm with an incomparable performance ratio.

In the next section, polymatroids and the spanning set problem for them are introduced. We also describe more formally how the spanning subgraph and set cover problems reduce to it. As another application of polymatroid theory for which an approximation algorithm is useful, we sketch the problem of finding a minimum cost set of joints in a bar and joint structure that make it rigid. The remaining sections contain our new analysis of the greedy algorithm.

2 Polymatroids and Applications

A *polymatroid* $P = (H, d)$ consists of a finite set H and a *dimension function* d. The dimension function maps each subset of H to an integer. It is nonnegative, monotone increasing, and submodular, i.e.

$$\forall\, X, Y \subset H:\ d(X \cap Y) + d(X \cup Y) \ \le\ d(X) + d(Y)\,.$$

If $d(\{h\}) \le k$ for all $h \in H$, then P is called a *k-polymatroid*. A 1-polymatroid is a matroid.

A set $X \subset H$ is called a *spanning set*, if $d(X) = d(H)$. Given a weight function $w : H \to \mathbb{R}$, the *minimum spanning set* problem is to find a spanning set X of P that minimizes $w(X) := \sum_{x \in X} w(x)$.

Let $\mathcal{H} = (V, H)$ be a connected k-restricted hypergraph with edge weights $w\colon H \to \mathbb{R}$. (That is, $|h| \le k$ for $h \in H$.) Then the *k-restricted minimum spanning subhypergraph* problem (k-MSS) is to:

$$\begin{array}{rl} \text{Minimize} & \sum_{i=1}^{m} w(h_i) \\ \text{such that} & h_1, \ldots, h_m \in H, \\ & \big(V, \{h_1, \ldots, h_m\}\big) \text{ is connected.} \end{array}$$

A minimum spanning tree for a graph can be found using the greedy algorithm in the graphic matroid. In a similar way, polymatroids are related to the minimum spanning subhypergraph problem.

The edge set of a minimum spanning tree is minimum-weight inclusion-maximal circuit-free, as well as a minimum-weight spanning set. In hypergraphs, a minimum-weight spanning edge set needs no longer be circuit free; a spanning tree might not even exist. Hence the minimum spanning tree problem has two generalizations for hypergraphs: one can search for an inclusion-maximal circuit-free set as well as for an inclusion-minimal spanning set of hyperedges. Both problems have a counterpart for polymatroids:

Firstly, one can ask for a minimum-weight inclusion-maximal circuit-free set in a polymatroid. A subset X of a polymatroid is circuit-free if $d(X) = \sum_{h \in X} d(h)$. This can be used, e.g., to find a common independent set of k matroids, since the sum of k matroid rank functions is the dimension function of a k-polymatroid. Korte and Hausmann [10] have shown that the greedy algorithm yields a k-approximation for the latter problem. Secondly, one can look at minimum-weight spanning sets in polymatroids, which is the subject of this paper.

The minimum spanning subhypergraph problem in $(k+1)$-restricted hypergraphs reduces to the minimum spanning subset problem in k-polymatroids in the following way: Given a hypergraph $\mathcal{H} = (V, H)$, consider the complete graph $G = (V, E)$, $E = \binom{V}{2}$, on the same vertex set and denote its graphic matroid by (E, r). Note that the set of graph edges contained in a hyperedge h has rank $r\left(\binom{h}{2}\right) = |h| - 1$, the number of edges of a spanning tree for h. Therefore, we obtain a k-polymatroid (H, d) if we put $d(X) := r\big(\bigcup\{ \binom{h}{2} \mid h \in X \}\big)$ for $X \subset H$. The spanning sets of the polymatroid (H, d) correspond one-to-one to those of the hypergraph (V, H).

Another application of the spanning set problem in polymatroids is the problem of finding a minimum cost set of joints in a bar and joint structure that make it rigid. We sketch this problem here to give an application that is apparently different from the hypergraph spanning set problem. We follow the exposition of Lovász and Plummer [12] and refer the reader there both for details and for other applications of polymatroid theory.

A bar and joint structure in dimension k is a graph $G = (V, E)$, where $V \subset \mathbb{R}^k$. It is a model for the statics of buildings and other constructions. The vertices represent joints and the edges represent bars connecting these joints. Such a bar and joint structure is said to be rigid, if no $\mathbb{R}^k$-motion other than translation and rotation is possible unless bars are stretched or compressed. Two questions naturally arise: Can one tell whether a given structure is rigid and, if not, how can it be fixed? Assume that we are able to "pin down" every vertex in $\mathbb{R}^k$ at given cost. Obviously, pinning down every vertex makes the structure rigid. Our algorithm can be used to approximate the minimum cost vertex subset that needs to be pinned down:

Let $\phi(G, P)$ denote the "degree of freedom" of G with $P \subset V$ pinned down (then rigidity means $\phi = 0$). Let $d(P) := \phi(G, \emptyset) - \phi(G, P)$. Then (V, d) is a k-polymatroid, and spanning sets correspond one-to-one to vertex subsets whose pinning-down makes G rigid.

As already mentioned in the introduction, the k-set cover problem reduces to the $(k + 1)$-restricted minimum spanning subhypergraph problem by adding a common point to all sets. An instance transformation in the reverse direction is unknown. However we can lift Chvátal's analysis of the greedy algorithm for set cover to MSS. The dimension of a set of hyperedges $X \subseteq H$ is the maximal size of a forest $F \subseteq \bigcup\{ \binom{h}{2} \mid h \in X \}$. Intuitively, the edges in F correspond to the points of a set cover instance which are covered by X. This does not yield an instance transformation, because there are many possibilities to choose F. Instead, we use the polymatroid properties to "deform" the algorithmic solution incrementally into an optimal one, while keeping the weights under control.

3 The Greedy Algorithm

We study the following variant of the greedy algorithm for the weighted spanning subset problem in a polymatroid (H, d).

GREEDY WEIGHTED SPANNING SET (H, d)

Init: $X \leftarrow \emptyset$;

While $d(X) < d(H)$ do

$$X \leftarrow X \cup \operatorname{argmin} \left\{ \frac{w(h)}{d(X \cup \{h\}) - d(X)} \;\middle|\; \{h\} \subseteq H \right\};$$

Output X.

In the rest of the paper we shall prove the following

Theorem 1. *Let $P = (H, d)$ be a k-polymatroid, and let $w : H \to \mathbb{R}$. If* ALG *denotes the output of* GREEDY WEIGHTED SPANNING SET (H, d) *and* OPT *is a minimum weight spanning set of P then*

$$w(\mathrm{ALG}) \;\leq\; H(k) \cdot w(\mathrm{OPT}) .$$

In other words, GREEDY WEIGHTED SPANNING SET *is an $H(k)$-approximation.*

4 Analysis

Let $n := d(H)$ and assume that ALG $= \{g_1, \ldots, g_m\}$, $m \in \mathbb{N}$, where g_i is the i-th element picked by the algorithm. We may assume without loss of generality that w is nonnegative.

Helgason [7], McDiarmid [13], and Lovász [11] have shown that each polymatroid can be obtained from some matroid $M = (E, r)$ by the following construction: An element h of the polymatroid corresponds to some $A_h \subset E$, such that $d(X) = r\big(\bigcup_{h \in X} A_h\big)$. For simplicity we identify h and A_h, i.e. we assume that $h \subset E$ for all $h \in H$ and that $d(X) = r(\bigcup X)$ for all $X \subset H$. Then $P = (H, d)$ is a k-polymatroid for $k = \max_{h \in H} r(h)$. In the following, we assume that P and M are related in this way.

Let $S_0 := A_0 := \emptyset$ and for $i = 1, \ldots, m$ let S_i be the extension of S_{i-1} to a basis of $A_i := \bigcup\{g_1, \ldots, g_i\}$. It is an essential property of matroids that this always works and since ALG is a spanning set, $S := S_m$ is a basis of E. Let $\ell_i := |S_i|$ and denote the elements of S by e_j, $j = 1, \ldots, n$, such that $S_i \setminus S_{i-1} = \{e_{\ell_{i-1}+1}, \ldots, e_{\ell_i}\}$.

For the optimum solution let $T \subset \bigcup$ OPT be another basis of E. For every $f \in T$ choose a $\psi(f) \in$ OPT such that $f \in \psi(f)$. Observe that, for all $h \in$ OPT, $\psi^{-1}(h)$ is independent, and therefore

$$\left|\psi^{-1}(h)\right| \;\leq\; r(h) \;\leq\; k. \tag{1}$$

In order to compare the two solutions we need a *deformation* φ of S to T, i.e. $\varphi : S \leftrightarrow T$ is a bijection such that, for every $j = 0, \ldots, n$,

$$T_j \;:=\; T \setminus \varphi\big(\{e_1, \ldots, e_j\}\big) \cup \{e_1, \ldots, e_j\} \tag{2}$$

is a basis for E. We define that deformation φ incrementally: Let $T_0 := T$. For each $j = 1, \dots, n$ there are two possibilities: If $e_j \in T_{j-1}$ then let $\varphi(e_j) := e_j$ and $T_j := T_{j-1}$. Otherwise, $T_{j-1} \cup \{e_j\}$ contains a circuit C. Since S is a basis, there exists an $f \in C \setminus S$. Let $\varphi(e_j) := f$ and $T_j := T_{j-1} \cup \{e_j\} \setminus \{f\}$. Then φ is injective by (2), and bijective for cardinality reasons.

The following piece of notation will be useful later: For $i = 0, \dots, m$, let

$$U_i := T_{\ell_i} \setminus S_i = T \setminus \varphi(S_i) = \varphi(S) \setminus \varphi(S_i) = \varphi(S \setminus S_i) = \varphi(\{e_{\ell_i+1}, \dots, e_n\}).$$

For $h \in \mathrm{OPT}$ let $U_i(h) := U_i \cap \psi^{-1}(h)$ and $u_i(h) := |U_i(h)|$. Observe that $U_i(h)$ is independent and, therefore, $u_i(h) = r(U_i(h))$. Since T is the disjoint union of the $U_{i-1} \setminus U_i$, we have

$$\psi^{-1}(h) \;=\; \dot{\bigcup}_{i=1}^{m} U_{i-1}(h) \setminus U_i(h). \tag{3}$$

We now turn to the analysis of the algorithm. For $i = 1, \dots, m$ and $h \in H$ define $c_{i-1}(h) := r(A_{i-1} \cup h) - r(A_{i-1})$. Then the choice of g_i in the algorithm is made such that

$$\frac{w(g_i)}{c_{i-1}(g_i)} \;\leq\; \frac{w(h)}{c_{i-1}(h)} \tag{4}$$

for all $h \in H$. S_i is a basis of A_i and for $h \in \mathrm{OPT}$, $S_i \cup U_i(h) \subset T_{\ell_i}$ is independent. Therefore,

$$r(A_i) + r(U_i(h)) \;=\; r(S_i) + r(U_i(h)) \;=\; r(S_i \cup U_i(h)) \;\leq\; r(A_i \cup h),$$

where the last inequality follows from the monotonicity of r and $\psi^{-1}(h) \subset h$. This shows that

$$u_i(h) \;=\; r(U_i(h)) \;\leq\; c_i(h). \tag{5}$$

To estimate $w(\mathrm{OPT})$ in terms of $w(\mathrm{ALG})$, we distribute $w(\mathrm{ALG})$ among S, i.e. for all $i = 1, \dots, m$ let

$$w(e_j) \;:=\; \frac{w(g_i)}{c_{i-1}(g_i)} \;=\; \frac{w(g_i)}{\ell_i - \ell_{i-1}}, \qquad j \;=\; \ell_{i-1} + 1, \dots, \ell_i. \tag{6}$$

Then

$$
\begin{aligned}
w(\mathrm{ALG}) &= \sum_{i=1}^{m} w(g_i) \;=\; \sum_{e\in S} w(e) \\
&= \sum_{f\in T} w\big(\varphi^{-1}(f)\big) \;=\; \sum_{h\in\mathrm{OPT}} \; \sum_{f\in\psi^{-1}(h)} w\big(\varphi^{-1}(f)\big) \\
&= \sum_{h\in\mathrm{OPT}} \sum_{i=1}^{m} \sum_{f\in U_{i-1}(h)\setminus U_i(h)} w\big(\varphi^{-1}(f)\big) && \text{by (3)} \\
&= \sum_{h\in\mathrm{OPT}} \sum_{i=1}^{m} \big(u_{i-1}(h) - u_i(h)\big)\, \frac{w(g_i)}{c_{i-1}(g_i)} && \text{by (6)} \\
&\le \sum_{h\in\mathrm{OPT}} w(h) \sum_{i=1}^{m} \frac{u_{i-1}(h) - u_i(h)}{c_{i-1}(h)} && \text{by (4)} \\
&\le \sum_{h\in\mathrm{OPT}} w(h) \sum_{i=1}^{m} \frac{u_{i-1}(h) - u_i(h)}{u_{i-1}(h)} && \text{by (5).}
\end{aligned}
$$

Using the bound

$$
\sum_{i=1}^{m} \frac{u_{i-1}(h) - u_i(h)}{u_{i-1}(h)} \;\le\; H\big(u_0(h)\big) - H\big(u_m(h)\big),
$$

$u_m(h) = 0$, and that $u_0(h) \le k$ by (1), this implies

$$
w(\mathrm{ALG}) \;\le\; H(k) \sum_{h\in\mathrm{OPT}} w(h) \;=\; H(k)\cdot w(\mathrm{OPT}),
$$

proving the theorem.

5 Acknowledgement

We are grateful to Toshihiro Fujito for bringing the article of Laurence Wolsey [18] to our attention.

References

1. Piotr Berman and Viswanathan Ramaiyer. Improved approximations for the Steiner tree problem. *Journal of Algorithms*, 17:381–408, 1994.
2. Al Borchers and Ding-Zhu Du. The k-Steiner ratio in graphs. *SIAM J. Computing*, 26:857–869, 1997.
3. Vašek Chvátal. A greedy heuristic for the set covering problem. *Math. Oper. Res.*, 4:233–235, 1979.

4. Ding-Zhu Du, Yanjun Zhang, and Qing Feng. On better heuristics for euclidean Steiner minimum trees. In *FOCS*, pages 431–439, 1991.
5. Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
6. Toshihiro Fujito. Approximation algorithms for submodular set cover with applications. *IEICE Trans. Inf. Syst*, E83-D(3):480–487, 2000. http://search.ieice.or.jp/.
7. Thorkell Helgason. Aspects of the theory of hypermatroids. In C. Berge and C. Ray-Chaudhuri, editors, *Hypergraph Seminar, Ohio State University 1972*, number 411 in Lecture Notes in Mathematics, pages 191–213. Springer, 1974.
8. Stefan Hougardy and Hans Jürgen Prömel. A 1.598 approximation algorithm for the Steiner problem in graphs. In *Proc. Symposium on Discrete Algorithms*, pages 448–453, 1999.
9. Marek Karpinski and Alexander Zelikovsky. New approximation algorithms for the Steiner tree problems. *Journal of Combinatorial Optimization*, 1:47–65, 1997.
10. Bernhard Korte and D. Hausmann. An analysis of the greedy algorithm for independence systems. *Ann. Discrete Math.*, 2:65–74, 1978.
11. László Lovász. Flats in matroids and geometric graphs. In P. J. Cameron, editor, *Proc. sixth British combinatorial conference*, Combinatorial Surveys, pages 45–86. Academic Press, 1977.
12. László Lovász and Michael D. Plummer. *Matching Theory*. North Holland, 1986.
13. Colin McDiarmid. Rado's theorem for polymatroids. *Math. Proc. Cambridge Philos. Soc.*, 78:263–281, 1975.
14. Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.
15. Hans Jürgen Prömel and Angelika Steger. RNC-approximation algorithms for the Steiner problem. In *Proc. Symposium on Theoretical Aspects of Computer Science*, volume LNCS 1200, pages 559–570. Springer-Verlag, 1997.
16. Gabriel Robins and Alexander Zelikovsky. Improved Steiner tree approximation in graphs. In *Proc. Symposium on Discrete Algorithms*, pages 770–779, 2000.
17. H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Math. Jap.*, 24:573–577, 1980.
18. Laurence A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.
19. Alexander Zelikovsky. An 11/6-approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.
20. Alexander Zelikovsky. Better approximation bounds for the network and euclidean Steiner tree problems. Technical Report CS-96-06, University of Virginia, 1996.