Approximability of Combinatorial Problems with Multi-agent Submodular Cost Functions

GAGAN GOEL, CHINMAY KARANDE, PUSHKAR TRIPATHI and LEI WANG
Georgia Institute of Technology

Applications in complex systems such as the Internet have spawned a recent interest in studying situations involving multiple agents with their individual cost or utility functions. In this paper, we introduce an algorithmic framework for studying combinatorial optimization problems in the presence of multiple agents with submodular cost functions. We study several fundamental covering problems in this framework and establish upper and lower bounds on their approximability.

Categories and Subject Descriptors: F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms: Theory, Economics, Algorithms, Design

Additional Key Words and Phrases: Submodular Functions, Combinatorial Optimization

1. INTRODUCTION

A multitude of fundamental computational problems with real-world applications can be cast in the following framework: We are given a set X of elements, a collection C of subsets of X (i.e. $C \subseteq 2^X$) and a cost function f over the subsets of X. The collection C is typically specified via a combinatorial structure like a matroid or a graph property (for instance, the set of all spanning trees in a graph). The objective is to select a set $S \in C$ that minimizes $f(S)$.

A major focus in theoretical computer science has been on linear cost functions. The study of combinatorial problems with linear cost functions has led to great developments in the theory of exact and approximation algorithms. However, linear cost functions do not always model the complex dependencies of the costs in a real-world setting. Often, they only serve as an approximation to the original functions. As a result, even though we might have a good algorithm for solving some linear optimization problem, the output solution can still be suboptimal.

Another feature that arises in a real-world setting is the presence of multiple agents, where each agent has her own cost function. Thus, in the optimal solution, each agent might build only a part of the required combinatorial structure. For example, the Internet is a complex multi-agent system where each service provider owns only a part of the network. For linear cost functions, it is easy to see that having multiple agents doesn't change the complexity of the problem. However, this is not the case for more general cost functions.

Motivated by these considerations, we define the following class of *combinatorial problems with multi-agent submodular cost functions* (MSCP) - We are given a set of elements X and a collection $C \subseteq 2^X$. We are also given k agents, where each

Authors' addresses: `gagan@gatech.edu`, `ckarande@cc.gatech.edu`, `tpushkar@cc.gatech.edu`, `lwang@cc.gatech.edu`

agent i specifies a normalized monotone submodular cost function $f_i : 2^X \to R^+$. The goal is to find a set $S \in C$ and a partition $S_1, ..., S_k$ of S such that $\sum_i f_i(S_i)$ is minimized.

Submodular functions form a rich class and capture the natural properties of economies of scale or the law of diminishing returns . A function $f : 2^X \to R^+$ is said to be submodular iff for any two sets S and T, such that $S \subset T \subseteq X$, and any $i \in X - T$, $f(T \cup i) - f(T) \leq f(S \cup i) - f(S)$. A function f is said to be monotone if $f(S) \leq f(T)$ for any $S \subseteq T$, and normalized if $f(\emptyset) = 0$.

To see that multiple agents with submodular cost functions do not necessarily lead to an overall submodular cost function, consider the following scenario: two agents, three tasks {A, B, and C}; each agent can do every task at cost 1 each, except agent 1 can do tasks {A,B} together at a cost of only 1 (and {A,B,C} at a cost of only 2), and agent 2 can do tasks {A,C} together at a cost of only 1 (and {A,B,C} at only 2). Then each agent's cost function is submodular, but the overall cost function is not submodular because the marginal cost of B relative to {A} is 0 which is smaller than the marginal cost of B relative to {A,C}.

In the past, there has been some work along these lines (See [Calinescu et al. 2007; Feige et al. 2007; Svitkina and Fleischer 2008; Vondrák 2008; Goemans et al. 2009]), but to the best of our knowledge, none of them has studied multi-agent submodular functions over a truly combinatorial structure. For instance, the work of [Calinescu et al. 2007] studies submodular function maximization over matroid constraints only in the single agent setting, the work of [Svitkina and Fleischer 2008; Vondrák 2008] considers the Multi-Agent submodular functions setting but the combinatorial structure (or the collection C) used in their problem is either the set of all subsets of X or just the set X.

Notice that by fixing the collection C to any particular combinatorial structure, one can define a subclass of the problems of interest. In this contribution, we study the following fundamental subclass of problems in MSCP :

(1) **Submodular Minimum Vertex Cover**: We are given an undirected graph $G(V, E)$. Element set X is same as the set of vertices V, and the collection C consists of all the vertex covers of graph G.

(2) **Submodular Shortest Path**: We are given a connected undirected graph $G(V, E)$, and a pair of vertices $s, t \in V$. Element set X is the set of all edges E, and the collection C consists of all the paths from s to t.

(3) **Submodular Minimum Perfect Matchings**: We are given an undirected graph $G = (V, E)$ with cost functions over E. G contains at least one perfect matching. Element set X is the set of all edges, and the collection C is defined as the set of all perfect matchings of graph G.

(4) **Submodular Minimum Spanning Tree**: We are given a connected undirected graph $G = (V, E)$ with cost functions over E. Element set X is the set of all edges, and the collection C is the set of all spanning trees of graph G.

For each of the above problems, we study both the single agent and the multi-agent setting.

1.1 Motivation and Applications

From a practical viewpoint, each of the problems we study is meaningful in its own right. For instance, shortest path and spanning trees are used in network design problems, and it is natural to assume that different agents could have different submodular cost functions depending on the set of edges they can construct cheaply.

From a theoretical perspective, one would like to extend the tools and techniques developed for combinatorial problems with linear cost functions to as general a setting as possible. Submodular functions are a natural generalization where one would expect to be able to extend the techniques. Despite a significant progress on some of the fundamental problems in this area recently [Calinescu et al. 2007; Feige et al. 2007; Svitkina and Fleischer 2008; Vondrák 2008; Goemans et al. 2009], the algorithmic theory for combinatorial problems with submodular cost functions is not substantially developed yet. Many more basic questions remain to be identified and solved, which could form the basis of tools and techniques for solving more complex problems.

Our multi-agent setup can also be interpreted as a reverse (procurement) auction where the auctioneer wants to buy a combinatorial structure, such as a spanning tree.

2. OUR RESULTS

We give an approximation algorithm and a lower bound on the approximability for each of the problems that we mentioned earlier. We present these results in the table below.

Table I. Results

	Single-agent		Multi-agent	
	Lower bound	Upper bound	Lower bound	Upper bound
Vertex Cover	$2-\epsilon$	2	$\Omega(\log n)$	$2\log n$
Shortest Path	$\Omega(n^{2/3})$	$O(n^{2/3})$	$\Omega(n^{2/3})$	$O(n)$
Perfect Matching	$\Omega(n)$	n	$\Omega(n)$	n
Spanning Tree	$\Omega(n)$	n	$\Omega(n)$	n

Since a submodular function is defined over an exponentially large domain, we assume access to a *value oracle* which returns the value of $f(S)$, when queried with the set $S \subseteq X$. Our hardness results are, thus, information theoretic, i.e., limits on the approximability of a problem when only polynomially many queries to the oracle are allowed. Our upper bounds are based on algorithms that use polynomially many queries as well as polynomial computation time.

We would like to draw attention to our lower bound result for the vertex cover problem in the single agent case. In the classical vertex cover problem, the best known approximation factor is 2, and the best known hardness of approximation is 1.3606 (assuming $P \neq NP$) [Dinur and Safra 2005]. Khot et al. [Khot and Regev 2008] showed that achieving a factor of $2-\epsilon$ might be hard by showing a hardness result based on the UGC conjecture [Khot 2002]. Our results for vertex cover in the single agent case implies that, if the cost function over the set of vertices is submodular, the optimal approximation factor is indeed 2.

2.1 Related Work

Svitkina and Fleischer [Svitkina and Fleischer 2008] studied submodular objective functions for problems like Sparsest cut, load balancing, and knapsack. They gave tight $O(\sqrt{\frac{n}{\log n}})$ upper and lower bounds on the approximability of these problems. For the submodular procurement auctions, where a set of n goods has to be allocated to k agents (i.e. collection set $C = \{X\}$) with submodular cost functions to minimize the overall cost, a simple greedy algorithm is known to have a factor $\log(n)$ [Hayrapetyan et al. 2005]. Our lower bound for vertex cover with multiple agents can be easily modified to give a lower bound of $\Omega(\log(n))$ for this problem. Some other related work in optimization that uses submodular functions include [Sharma et al. 2007; Sviridenko 2004; Vondrák 2008; Goemans et al. 2009].

3. DISCUSSION

The setting that we have considered in this work is quite general and is a very exciting avenue of research. There are many other interesting problems in this class such as minimum graph cut and edge cover which could be studied in the future work. We have considered covering problems in this work, but one can ask the same questions for packing problems like maximum matching. One could also consider even more general functions like Subadditive functions. The extension to multiagent makes a natural connection to game theory and mechanism design, which could also be explored in the future work.

REFERENCES

Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J. 2007. Maximizing a submodular set function subject to a matroid constraint (extended abstract). In *IPCO*. 182–196.

Dinur, I. and Safra, S. 2005. On the hardness of approximating minimum vertex cover. *Annals of Mathematics 162*, 2005.

Feige, U., Mirrokni, V. S., and Vondrák, J. 2007. Maximizing non-monotone submodular functions. In *FOCS*. 461–471.

Goemans, M. X., Harvey, N. J. A., Iwata, S., and Mirrokni, V. S. 2009. Approximating submodular functions everywhere. In *SODA*. 535–544.

Hayrapetyan, A., Swamy, C., and Tardos, É. 2005. Network design for information networks. In *SODA*. 933–942.

Khot, S. 2002. On the power of unique 2-prover 1-round games. In *STOC*. 767–775.

Khot, S. and Regev, O. 2008. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci. 74,* 3, 335–349.

Sharma, Y., Swamy, C., and Williamson, D. P. 2007. Approximation algorithms for prize collecting forest problems with submodular penalty functions. In *SODA*. 1275–1284.

Sviridenko, M. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett. 32,* 1, 41–43.

Svitkina, Z. and Fleischer, L. 2008. Submodular approximation: Sampling-based algorithms and lower bounds. In *FOCS*. 697–706.

Vondrák, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*. 67–74.