The Budgeted Maximum Coverage Problem

Samir Khuller* Anna Moss† Joseph (Seffi) Naor‡

Abstract

The budgeted maximum coverage problem is: given a collection $\mathcal{S}$ of sets with associated costs defined over a domain of weighted elements, and a budget L, find a subset of $\mathcal{S}' \subseteq \mathcal{S}$ such that the total cost of sets in $\mathcal{S}'$ does not exceed L, and the total weight of elements covered by $\mathcal{S}'$ is maximized. This problem is NP-hard. For the special case of this problem, where each set has unit cost, a $(1 - \frac{1}{e})$-approximation is known. Yet, no approximation results are known for the general cost version. The contribution of this paper is a $(1 - \frac{1}{e})$-approximation algorithm for the budgeted maximum coverage problem. We also argue that this approximation factor is the best possible, unless $NP \subseteq DTIME(n^{\log \log n})$.

1 Introduction

The *budgeted maximum coverage problem* is defined as follows. A collection of sets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ with associated costs $\{c_i\}_{i=1}^m$ is defined over a domain of elements $X = \{x_1, x_2, \ldots, x_n\}$ with associated weights $\{w_i\}_{i=1}^n$. The goal is to find a collection of sets $\mathcal{S}' \subseteq \mathcal{S}$, such that the total cost of elements in $\mathcal{S}'$ does not exceed a given budget L, and the total weight of elements covered by $\mathcal{S}'$ is maximized.

The *unit cost* version of the problem, where each set has unit cost, and the goal is to find k sets from $\mathcal{S}$ such that the total weight of covered elements is maximized, has been considered previously. Many applications arising in circuit layout, job scheduling, facility location, and other areas, may be modeled using the maximum coverage problem (see [H97] and the references therein for examples of applications).

The budgeted maximum coverage problem introduces a more flexible model for the applications mentioned above. Consider, for example, the problem of locating k identical facilities so that the market share is maximized, introduced in [MZH83]. Namely, there exist clients with associated profits, situated in known locations. A client will use a facility if the facility is within the specified distance from the client. The goal is to locate k facilities so that the total profit of the clients served by the facilities, is maximized. In another version of this problem, considered in [BLF92, BBL95], and known as *optimal location of discretionary service facilities*, facilities gain profits associated with travel paths of customers; a facility covers a path if it is located in one of the nodes on the path,

[*] Computer Science Dept., University of Maryland, College Park, MD 20742. E-mail:samir@cs.umd.edu. Research supported by NSF CAREER Award CCR-9501355.

[†] Computer Science Dept., Technion, Haifa 32000, Israel. E-mail:annaru@cs.technion.ac.il.

[‡] Computer Science Dept., Technion, Haifa 32000, Israel. E-mail:naor@cs.technion.ac.il.

or at some vertex “close” to the path. Clearly, the problems described above can be modeled by the unit cost maximum coverage problem. However, in practice, the cost of constructing a facility may depend on certain factors associated with the location of the facility. For example, each candidate site for constructing a facility is associated with some cost, and one is assigned a limited budget for constructing the facilities. This generalization of the problem of locating facilities to maximize market share is also discussed in [MZH83]. The budgeted maximum coverage problem is a model that allows us to handle this type of applications.

Unfortunately, even the unit cost version of the maximum coverage problem is NP-hard, by a straightforward reduction from the set cover problem. Therefore, we look for an approximate solution which is near-optimal and computable in polynomial time. An approximation algorithm for a maximization problem is said to achieve approximation ratio δ, if the solution delivered by the algorithm for any instance of the problem is always at least the multiplicative factor of δ from the optimum for this instance.

Several results related to the unit cost maximum coverage problem were presented in the literature. Nemhauser and Wolsey [NW81] and Conforti and Cornuejols [CC84] considered the general problem of maximizing submodular functions (a function f is called submodular if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$, for any sets A, B in the domain of f). They have shown, in particular, that the greedy heuristic achieves an approximation factor of $1 - (1 - \frac{1}{k})^k > (1 - \frac{1}{e})$ for maximizing nondecreasing submodular functions satisfying $f(\emptyset) = 0$, where k is the maximum cardinality of sets in the optimization domain. Vohra and Hall [VH93] noted that the unit cost maximum coverage problem belongs to this class of problems. The greedy heuristic for the special case of the maximum coverage problem picks at each step a set maximizing the weight of the uncovered elements. Hochbaum and Pathria [HP94] (see also Hochbaum [H97]) presented an elegant analysis of the greedy heuristic. As we show in the sequel, the greedy heuristic cannot be trivially generalized to achieve a good approximation for the budgeted version of the problem.

In this paper we present a $(1 - \frac{1}{e})$-approximation algorithm for the budgeted maximum coverage problem. This is the first approximation algorithm presented for the budgeted version of the problem. We show that a small modification to the greedy heuristic yields a constant approximation factor for this problem. We then show how to further improve the approximation factor to $(1 - \frac{1}{e})$ using the enumeration technique. We argue that the latter approximation ratio is the best possible for the maximum coverage problem, in the sense that no approximation algorithm with performance guarantee $(1 - \frac{1}{e} + \epsilon)$ exists for any $\epsilon > 0$, unless $NP \subseteq DTIME(n^{\log \log n})$.

The rest of the paper is organized as follows. In Section 2 we show that the greedy heuristic cannot achieve a good performance guarantee for the budgeted maximum coverage problem; we present an algorithm that achieves a constant factor approximation for this problem and analyze its performance guarantee. Section 3 presents an improved algorithm and the analysis of its performance guarantee. In Section 4 we present a bound on the hardness of approximation for the budgeted maximum coverage problem.

2 The Modified Greedy Algorithm

Throughout the rest of this paper we use the following notation. Let $\{c_i\}_{i=1}^m$ denote the costs of the sets in $\mathcal{S}$, let $\{w_i\}_{i=1}^n$ denote the weights of the elements in X, and let W_i be the total weight of the elements covered by set S_i, $i = 1, \ldots, m$. Let $G \subseteq \mathcal{S}$ be a collection of sets. Let $w(G)$ denote the total weight of the elements covered by G. Let W_i', $i = 1, \ldots, m$, denote the total weight of the elements covered by set S_i, but not covered by any set in G. Without loss of generality we may assume that the cost of each set does not exceed L, since sets of cost greater than L do not belong to any feasible solution.

We first observe that the greedy heuristic that picks at each step a set maximizing the ratio $\frac{W_i'}{c_i}$ has an unbounded approximation factor. Consider, for example, two elements, x_1 of weight 1 and x_2 of weight p. Let $S_i = \{x_i\}$, $i = 1, 2$, let $c_1 = 1$, $c_2 = p + 1$, and let $L = p + 1$. The optimal solution contains the set S_2 and has weight p, while the solution picked by the greedy heuristic contains the set S_1 and has weight 1. The approximation factor for this instance is p, and is therefore unbounded.

Next, we show that a small modification to the greedy heuristic achieves a constant approximation factor for the budgeted maximum coverage problem. The algorithm is as follows.

Algorithm 1:

```
$G \leftarrow \emptyset$; $C \leftarrow 0$; $U \leftarrow \mathcal{S}$;
Repeat
        select $S_i \in U$ that maximizes $\frac{W_i'}{c_i}$
        if $C + c_i \leq L$ then
                $G \leftarrow G \cup S_i$
                $C \leftarrow C + c_i$
        $U \leftarrow U \setminus S_i$
Until $U = \emptyset$
Select a set $S_t$ that maximizes $W_t$ over $\mathcal{S}$.
If $W(G) \geq W_t$, output $G$, otherwise, output $\{S_t\}$
```

Algorithm 1 finds a collection of sets according to the greedy heuristic. This collection is the first candidate for the final output. The second candidate is a single set S_t for which W_t is maximized. The algorithm outputs the candidate solution having the maximum weight.

Next, we analyze the performance guarantee of Algorithm 1. Let OPT denote the collection of sets in an optimal solution. Let r be the number of iterations executed by the greedy heuristic (the "repeat loop") until the first set from OPT is considered, but not added to G, because its addition would violate the budget L. Let l be the number of sets added to G in the first r iterations. Without loss of generality, we may renumber the sets so that S_i is the i-th set added to G, $i = 1, \ldots, l$, and S_{l+1} is the first set from OPT selected by the algorithm but not added to G. Let G_i, $i = 1, \ldots, l + 1$ denote $\cup_{j=1}^i S_j$. Finally, let j_i be the index of the iteration in which set S_i was considered, $i = 1, \ldots, l + 1$.

The following two lemmas generalize the lemmas for the unit cost version of the problem, presented in [HP94] (see also [H97]).

Lemma 1: After each iteration j_i, $i = 1, \ldots, l+1$, the following holds:

$$w(G_i) - w(G_{i-1}) \;\geq\; \frac{c_i}{L}(w(OPT) - w(G_{i-1})).$$

Proof: Clearly, $w(OPT) - w(G_{i-1})$ is no more than the weight of the elements covered by OPT, but not covered by G_{i-1}. For each set in $OPT \setminus G_{i-1}$, the ratio of weight to cost is at most $\frac{W_i'}{c_i}$, since S_i maximizes this ratio over all sets that had not been selected before iteration j_i. Since the total cost of the sets in $OPT \setminus G_{i-1}$ is bounded by the budget L, the total weight of the elements covered by the sets in $OPT \setminus G_{i-1}$ and not covered by G_{i-1}, is at most $L \cdot \frac{W_i'}{c_i}$. Hence, we get

$$w(OPT) - w(G_{i-1}) \;\leq\; L \cdot \frac{W_i'}{c_i}. \tag{1}$$

But, from the definition of W_i', it follows that $W_i' = w(G_i) - w(G_{i-1})$. Substituting $w(G_i) - w(G_{i-1})$ for W_i' in (1), and multiplying both sides by $\frac{c_i}{L}$, we get the required inequality. □

Lemma 2: After each iteration j_i, $i = 1, \ldots, l+1$, the following holds:

$$w(G_i) \geq \left[1 - \prod_{k=1}^{i}\left(1 - \frac{c_k}{L}\right)\right] \cdot w(OPT).$$

Proof: We prove the lemma by induction on the number of iterations in which the sets S_i, $i = 1, \ldots, l+1$, are considered. For $i = 1$, $w(G_1) = W_1 = W_1'$, and we need to prove that $W_1' \geq \frac{c_1}{L} \cdot w(OPT)$. This follows since the ratio $\frac{W_1'}{c_1}$ is maximum over all sets, and since the cost of the optimum is bounded by L. Suppose the statement of the lemma holds for iterations $j_1, \ldots, j_{i-1}$. We show that it also holds for iteration j_i.

$$\begin{aligned}
w(G_i) &= w(G_{i-1}) + [w(G_i) - w(G_{i-1})] \\
&\geq w(G_{i-1}) + \frac{c_i}{L} \cdot (w(OPT) - w(G_{i-1})) \\
&= \left(1 - \frac{c_i}{L}\right) \cdot w(G_{i-1}) + \frac{c_i}{L} \cdot w(OPT) \\
&\geq \left(1 - \frac{c_i}{L}\right) \cdot \left(1 - \prod_{k=1}^{i-1}\left(1 - \frac{c_k}{L}\right)\right) \cdot w(OPT) + \frac{c_i}{L} \cdot w(OPT) \\
&= \left(1 - \prod_{k=1}^{i}\left(1 - \frac{c_k}{L}\right)\right) \cdot w(OPT).
\end{aligned}$$

In the above proof, the first inequality follows by Lemma 1, and the second inequality follows from the induction hypothesis. □ □

Theorem 3: Algorithm 1 achieves an approximation factor of $\frac{1}{2} \cdot (1 - \frac{1}{e})$ for the budgeted maximum coverage problem.

Proof: First, we observe that for $a_1, \ldots, a_n \in \mathrm{R}^+$ such that $\sum_{i=1}^{n} a_i = A$, the function $(1 - \prod_{i=1}^{n}(1 - \frac{a_i}{A}))$ achieves its minimum when $a_1 = a_2 = \ldots = a_n = \frac{A}{n}$.

Let $c(P)$ denote the total cost of a collection of sets $P \subseteq \mathcal{S}$. Applying Lemma 2 to iteration j_{l+1}, we get:

$$\begin{aligned}
w(G_{l+1}) &\geq \left[1 - \prod_{k=1}^{l+1}\left(1 - \frac{c_k}{L}\right)\right] \cdot w(OPT) \\
&\geq \left[1 - \Pi_{k=1}^{l+1}\left(1 - \frac{c_k}{c(G_{l+1})}\right)\right] \cdot w(OPT) \\
&\geq \left[1 - \left(1 - \frac{1}{l+1}\right)^{l+1}\right] \cdot w(OPT) \\
&\geq \left(1 - \frac{1}{e}\right) \cdot w(OPT)
\end{aligned}$$

Note that the second inequality follows from the fact that adding S_{l+1} to G_l violates the budget L, and therefore $c(G_{l+1}) = c(G_l) + c_{l+1} > L$. Therefore, we get:

$$w(G_{l+1}) = w(G_l) + W'_{l+1} \geq \left(1 - \frac{1}{e}\right) \cdot w(OPT). \tag{2}$$

Next we observe that W'_{l+1} is at most the maximum weight of the elements covered by a single set, i.e., the weight W_t of the second candidate solution found by the algorithm. Therefore,

$$w(G_l) + W_t \geq w(G_l) + W'_{l+1} \geq \left(1 - \frac{1}{e}\right) \cdot w(OPT).$$

From the inequality above we have that at least one of the values $w(G_l)$, W_t is greater than or equal to $\frac{1}{2} \cdot (1 - \frac{1}{e}) w(OPT)$, and the theorem follows. □ □

3 Improving the Performance Guarantee

In this section we show how to generalize Algorithm 1 using the enumeration technique, so as to obtain better approximation factors. Let k be some fixed integer. We consider all subsets of $\mathcal{S}$ of cardinality k which have cost at most L, and we complete each subset to a candidate solution using the greedy heuristic. Another set of candidate solutions consists of all subsets of $\mathcal{S}$ of cardinality less than k which have cost at most L. The algorithm outputs the candidate solution having the greatest weight. Below follows the formal description of the algorithm.

Algorithm 2:

```
$H_1 \leftarrow \arg\max\{w(G)$, such that $G \subseteq \mathcal{S}$, $|G| < k$, and $c(G) \leq L\}$
$H_2 \leftarrow \emptyset$
For all $G \subseteq \mathcal{S}$, such that $|G| = k$ and $c(G) \leq L$ do
    $U \leftarrow \mathcal{S} \setminus G$
    Repeat
        select $S_i \in U$ that maximizes $\frac{W'_i}{c_i}$
        if $c(G) + c_i \leq L$ then
```

```
                $G \leftarrow G \cup S_i$
            $U \leftarrow U \setminus S_i$
        Until $U = \emptyset$
        if $w(G) > w(H_2)$ then $H_2 \leftarrow G$
    If $w(H_1) > w(H_2)$, output $H_1$, otherwise, output $H_2$
```

Next, we prove the following theorem about the performance guarantee of Algorithm 2.

Theorem 4: For $k \geq 3$, Algorithm 2 achieves an approximation factor of $(1 - \frac{1}{e})$ for the budgeted maximum coverage problem.

Proof: We may assume without loss of generality that $|OPT| > k$, since otherwise Algorithm 2 finds an optimal solution.

Let us order the sets in OPT by selecting at each step the set in OPT that covers uncovered elements of maximum total weight. Let Y be the first k sets in this order. Consider the iteration of Algorithm 2 in which subset Y is considered. Let Y' denote the collection of sets that is added to Y by the algorithm, and let $G = Y \cup Y'$. Clearly, $w(G) = w(Y) + w(Y')$. Here, and in the sequel of this section, we slightly abuse notation and let $w(P)$ denote the total weight of the elements covered by P but not covered by Y, for any collection of sets $P \subseteq \mathcal{S} \setminus Y$.

The completion of Y to G can be viewed as an application of the greedy heuristic from Algorithm 1. Therefore, the results from the previous subsection can be used in this case. Let r be the number of iterations executed by the greedy heuristic during the completion of subset Y to G, until the first set from $OPT \setminus Y$ is considered, but not added to Y' because its addition would violate the budget L. Let l be the number of sets added to Y' in the first r iterations. Let S_i be the i-th set added to Y', $i = 1, \ldots, l$, and let S_{l+1} be the first set from $OPT \setminus Y$ that is considered but not added to Y'. Applying Inequality (2), we get:

$$w(Y') + W'_{l+1} \geq \left(1 - \frac{1}{e}\right) \cdot w(OPT \setminus Y).$$

Next, we observe that when the sets of OPT are ordered, the weight of the uncovered elements covered by each set in Y is at least W'_{l+1}, otherwise set S_{l+1} would have belonged to Y. Therefore,

$$W'_{l+1} \leq \frac{1}{k} w(Y). \tag{3}$$

We now get,

$$\begin{aligned}
w(G) &= w(Y) + w(Y') \\
&\geq w(Y) + \left(1 - \frac{1}{e}\right) \cdot w(OPT \setminus Y) - W'_{l+1} \\
&\geq w(Y) + \left(1 - \frac{1}{e}\right) \cdot w(OPT \setminus Y) - \frac{w(Y)}{k} \\
&\geq \left(1 - \frac{1}{k}\right) \cdot w(Y) + \left(1 - \frac{1}{e}\right) \cdot w(OPT \setminus Y)
\end{aligned}$$

But, $w(Y) + w(OPT \setminus Y) = w(OPT)$, and hence we get:

$$w(G) \geq \left(1 - \frac{1}{e}\right) \cdot w(OPT) \text{ for } k \geq 3,$$

proving the theorem. □ □

4 A Lower Bound

We now show that even the unit cost version of the maximum coverage problem cannot be approximated within a factor better than $(1 - \frac{1}{e})$, unless $NP \subseteq DTIME(n^{\log\log n})$. We note that this was also independently observed by Feige [F97]. We use an argument similar to that in [GK98] for proving a hardness result for a facility location problem.

Suppose that there exists an approximation algorithm, denoted by A, guaranteeing approximation within a factor $\alpha > (1 - \frac{1}{e})$ for the unit cost maximum coverage problem. We show that in this case, the set cover problem could be approximated within an approximation factor better than $\ln n$, where n is the number of elements. By the following result of Feige [F96] this implies that $NP \subseteq DTIME(n^{\log\log n})$.

Theorem 5: If the set cover problem is approximable within a factor of $(1 - \epsilon) \ln n$ for any $\epsilon > 0$ then $NP \subseteq DTIME(n^{\log\log n})$.

Consider a set cover instance where each set has unit cost. Assign unit weight to each element. Let n be the number of elements in the instance. Now, apply algorithm A to the resulting maximum coverage instance, where the number of allowed sets is fixed to k, the cardinality of an optimal set cover (we may assume that the number of sets in the optimal set cover is known; otherwise we may try each $k \in \{1, \ldots, n\}$). Recall that in every feasible solution for the set cover instance, each element is covered by some set. Hence, the number of elements covered by an optimal solution for the maximum coverage instance is n, the total number of elements. Therefore, we are guaranteed to cover an α fraction of all the elements, where $\alpha > (1 - \frac{1}{e})$. Now we discard the sets that have been selected and the elements they cover, and apply again Algorithm A with the same bound on the number of sets. We repeat this procedure until all elements are covered.

Suppose in iteration i, the number of uncovered elements at the start of the iteration is n_i. The algorithm picks k sets and covers αn_i elements, where $\alpha > (1 - \frac{1}{e})$. Clearly, $n_{i+1} = n_i(1 - \alpha)$ and $n_1 = n$. Suppose after ℓ iterations $n_{\ell+1} = 1$. The total number of sets that are picked is ℓk, with an approximation factor of ℓ.

$$n_{\ell+1} = 1 = n(1 - \alpha)^{\ell}.$$

Hence

$$\ell = \frac{\ln n}{\ln(\frac{1}{1-\alpha})}.$$

Using the fact that $\alpha > (1 - \frac{1}{e})$, we obtain

$$\ln\left(\frac{1}{1-\alpha}\right) > 1.$$

This implies that we have an approximation factor of $c \ln n$ for some $c < 1$. On the other hand, it was shown by Feige [F96] that the set cover problem is not approximable within a factor $(1-\epsilon)\ln n$ for any $\epsilon > 0$ unless $NP \subseteq DTIME(n^{\log\log n})$. We conclude that no approximation algorithm with ratio better than $(1-\frac{1}{e})$ exists for the maximum coverage problem unless $NP \subseteq DTIME(n^{\log\log n})$.

References

[BBL95] O. Berman, D. Bertsimas, and R. C. Larson. *"Locating Discretionary Service Facilities, II: Maximizing Market Size, Minimizing Inconvenience"*, Operations Research, vol. 43(4), pp. 623-632, 1995

[BLF92] O. Berman, R. C. Larson, N. Fouska. *"Optimal Location of Discretionary Service Facilities"*, Transportation Science, vol. 26(3), pp. 201-211, 1992

[CC84] M. Conforti and G. Cornuejols. *"Submodular Set Functions, Matroids and the Greedy Algorithm: Tight Worst-Case Bounds and Some Generalizations of the Rado-Edmonds Theorem"*, Discrete Applied Mathematics, vol. 7, pp. 251-274, 1984

[F96] U. Feige. *"A Threshold of ln(n) for Approximating Set Cover"*, Proc. of the 28-th Annual ACM Symposium on the Theory of Computing, pp. 314-318, 1996

[F97] U. Feige, Private communication, November 1997.

[GK98] S. Guha and S. Khuller. *"Greedy Strikes Back: Improved Facility Location Algorithms"*, to appear, 9th Annual ACM-SIAM Symposium on Discrete Algorithms, 1998

[H97] D. S. Hochbaum, ed. *"Approximation Algorithms for NP-hard Problems"*, PWS Publishing Company, 1997

[HP94] D. S. Hochbaum and A. Pathria, *"Analysis of the greedy approach in covering problems"*, Unpublished manuscript, 1994

[NW81] G. L. Nemhauser and L. Wolsey. *"Maximizing Submodular Set Functions: Formulations and Analysis of Algorithms"*, in: Studies of Graphs and Discrete Programming, North-Holland, Amsterdam, pp. 279-301, 1981

[MZH83] N. Megiddo, E. Zemel, and S. L. Hakimi. *"The Maximum Coverage Location Problem"*, SIAM Journal on Algebraic and Discrete Methods, vol. 4(2), pp. 253-261

[VH93] R. V. Vohra and N. G. Hall. *"A probabilistic Analysis of the Maximal Covering Location Problem"*, Discrete Applied Mathematics, vol. 43, pp. 175-183, 1993