The Set Cover Problem: Approximation Algorithms and Analysis

Analysis Report

1 Problem Definition

The Set Cover problem is defined as follows: Given a universe $U = \{1, 2, ..., n\}$ and a collection $S = \{S_1, S_2, ..., S_m\}$ of subsets of U, where each subset S_i has a non-negative weight w_i, find a minimum-weight subcollection $C \subseteq S$ such that $\bigcup_{S_i \in C} S_i = U$.

2 Theoretical Analysis

2.1 Greedy Approximation Algorithm

The greedy algorithm iteratively selects the subset with maximum number of uncovered elements divided by its weight.

Algorithm 1 Greedy Weighted Set Cover

$C \leftarrow \emptyset$
$U' \leftarrow U$
while $U' \neq \emptyset$ **do**
    Select $S_i \in S$ maximizing $\frac{|S_i \cap U'|}{w_i}$
    $C \leftarrow C \cup \{S_i\}$
    $U' \leftarrow U' \setminus S_i$
end while
return C

2.2 Approximation Ratio Analysis

Let $H(d) = \sum_{i=1}^{d} \frac{1}{i}$ denote the d-th harmonic number.

The greedy algorithm achieves an approximation ratio of $H(d)$ where $d = \max_i |S_i|$.

Proof. Let OPT be the optimal solution cost. Assign cost to elements when they get covered:

$$\text{cost}(e) = \frac{w_i}{|S_i \cap U'|}$$

when e is covered by S_i containing k uncovered elements.

The total cost of greedy solution:

$$\sum_{e \in U} \text{cost}(e) = \sum_{S_i \in C} w_i$$

Consider any set S in optimal solution. Order elements by when greedy covers them. The k-th element in S has cost at most $\frac{w(S)}{|S|-k+1}$. Thus:

$$\sum_{e \in S} \text{cost}(e) \leq w(S) \cdot H(|S|) \leq w(S) \cdot H(d)$$

Summing over all sets in optimal solution:

$$\sum_{S \in OPT} \sum_{e \in S} \text{cost}(e) \leq H(d) \cdot OPT$$

Since every element is covered by at least one set in OPT:

$$\text{Greedy Cost} \leq H(d) \cdot OPT$$

□

2.3 LP Rounding Analysis

Consider the linear programming relaxation:

$$\begin{aligned} \min \sum_{i=1}^{m} w_i x_i \\ \text{s.t.} \sum_{i: e \in S_i} x_i \geq 1 \quad \forall e \in U \\ x_i \geq 0 \quad \forall i = 1, ..., m \end{aligned}$$

There exists a rounding scheme achieving $O(\log n)$ approximation.

Proof. Let x^* be optimal LP solution. Round x_i to 1 if $x_i^* \geq 1/f$ where f is maximum frequency. Each element is covered with probability at least $1-(1-1/f)^f \geq 1-1/e$. Repeating $O(\log n)$ times yields constant probability all elements covered, giving $O(\log n)$ approximation. □

2.4 Lower Bounds

Unless $P = NP$, Set Cover cannot be approximated within $(1 - \epsilon)\ln n$ for any $\epsilon > 0$.

Proof. Reduction from Dominating Set. Given graph $G = (V, E)$, create instance where $U = V$ and for each $v \in V$, create set containing v and its neighbors. Dominating set of size k corresponds to set cover of size k. Since Dominating Set is hard to approximate within $(1-\epsilon)\ln n$, so is Set Cover. □

3 Experimental Results

The implemented algorithms show different trade-offs:

Algorithm	Approximation Ratio	Time Complexity	Space Complexity
Greedy Unweighted	$H(d)$	$O(mn)$	$O(m+n)$
Greedy Weighted	$H(d)$	$O(mn)$	$O(m+n)$
LP Rounding	$O(\log n)$	$O(\text{poly}(m,n))$	$O(mn)$

Table 1: Algorithm Comparison

4 Conclusion

The Set Cover problem admits efficient approximation algorithms with provable guarantees. The greedy algorithm provides $H(d)$-approximation, which is optimal up to constant factors under standard complexity assumptions. LP-based methods offer alternative approaches with different trade-offs between approximation ratio and computational complexity.