Set Cover Research Proposal

Mark Kasun (3806554), Ali Mowazi (5879953)

February 12, 2016

1 Introduction to Set Cover

Set Cover in decision form is a NP- Complete problem that has a few different forms such as Vertex Cover or Weighted Set Cover. Our team will be researching and covering Weighted Set Cover. Set Cover is best explained as minimum number of subsets needed that when taken the union of these subsets they together cover all elements in the given set. In decision form for weighted set cover we are given a Set u of n elements, a collection of subsets $S_1, S_2, \ldots, S_m$ of subsets with weight $wt(i)$ for each subset and a number k, then we are asked if there is a collection of at most k subsets with at most weight of $wt(L)$[1]. Richard Manning Karp showed in his 1972 paper "Reducibility Among Combinatorial Problems" that there is a polynomial time reduction from SAT to Set Cover, proving it is NP-Complete[2]. There are a few well known attempts at solving set cover such as the greedy algorithm or through linear programming. Set Cover has many real world applications, some are fairly famous applications. IBM realized it had 5000 known computer viruses that had 9000 substrings, they were able to find that only 180 needed subsets to cover the 5000 viruses[3]. So instead of searching for 5000 viruses they only need to search for 180 subsets of consecutive bytes that would cover the 5000 viruses. We will go on to show two methods to tackle the Weighted Set Cover problem. We will have one greedy algorithm and algorithm using simulated annealing.

2 Test Data

The OR-Library includes numerous test data sets for set cover which we can use to test our algorithms[4]. Originally, we were going to look at unweighted set cover but the libraries we found all dealt with weighted set cover and our the algorithms we wished to look at were easily adaptable to deal with it. The library includes references to where we can find optimal solutions for the test data which will greatly enhance our work.

[1] Yair Bartal (2005) Advanced Algorithms – Lecture retrieved from http://www.cs.huji.ac.il/course/2005/algo2/scribes/lecture2.pdf

[2] Richard M. Karp (1972). "Reducibility Among Combinatorial Problems"

[3] David P. Williamson(1998) Lecture Notes on Approximation Algorithms Retrieved from people.orie.cornell.edu/dpw/cornell.ps

[4] http://people.brunel.ac.uk/ mastjjb/jeb/orlib/scpinfo.html

3 Greedy Set Cover

For the project, we will be looking at the greedy algorithm for solving general set cover problems. The greedy algorithm is an approximation algorithm with a guaranteed performance compared with the optimal solution to the problem. Unlike most approximation algorithms, the guarantee is not a constant multiple of the optimal but instead a it is a function multiple of the optimal based on the size of the input for the instance. The greedy algorithm is guaranteed provide a solution that is less than or equal to $\alpha \cdot OPT(I)$ where α is given by

$$\alpha = H_k = \sum_{i=1}^{k} \frac{1}{i} \leq 1 + \log(k)$$

where k is the size of the largest subset and H_k is the kth partial sum of the harmonic series. Since k will always be less than n, we can re-frame α in terms of n if so desired.

The implementation of the algorithm is fairly simple. Initialize an array that holds the cost effectiveness for each subset where cost effectiveness is the cost of the subset divided by the number of elements in the subset. Take the set with the lowest cost effectiveness and consider all elements in the subset covered. Update the cost effectiveness of all sets excluding the elements covered by the previous set. Repeat the process until all elements in the set are covered. The subsets chosen through the process are the result of the algorithm. The algorithm works on unweighted set cover where the weights of each subset is simply 1. The algorithm has a worst-case running time of $O(N \log(N))$ where N is the actual size of the input including all subsets.

The guarantee of $H_k \cdot OPT(I)$ seems weak compared to algorithms for other NP-Hard problems where there are 2-approximation guarantees. The harmonic series quickly increases past 2, passing it for $k > 4$. However, it has been shown that the efficiency of the greedy algorithm for general set cover problems cannot be improved to a constant nor to anything significantly more efficient than the greedy algorithm[5] (Pusztai, 2008).

[5] Pusztai, p. 408-409

4 References

Pusztai, Pál. "An Application of the Greedy Heuristic of Set Cover to Traffic Checks." Central European Journal of Operations Research 16.4 (2008): 407-14. Print.

Yair Bartal (2005) Advanced Algorithms – Lecture retrieved from http://www.cs.huji.ac.il/cc

Young, Neal E. "Greedy Set-Cover Algorithms." Encyclopedia of Algorithms. Ed. Ming-Yang Kao. New York, NY: Springer, 2008. 379-81.