International Conference on Multidisciplinary approaches in Social Sciences, Humanities and Sciences
Sri S.Ramasamy Naidu Memorial College, Sattur, Tamil Nadu, India (MASHS-18)
14th December 2018 www.conferenceworld.in ISBN:978-93-87793-61-3

A Study on Greedy Algorithms for Set Cover Problem

R.Prabamanieswari[1], D.S.Mahendran[2], T.C. Raja Kumar[3],

[1]Research Scholar (Reg No. 11991), Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli -12, (Resarch Center: St. Xavier's College, Palayamkottai)

[2]Principal, Aditanar College of Arts & Science, Tiruchendur, India

[3]Associate Professor, Department of Computer Science, St. Xavier's College, Palayamkottai, India

ABSTRACT

The set cover problem (SCP) is a popular optimization problem that has been applied to a wide range of industrial applications. Several approximation algorithms have been proposed to find approximate solutions for Minimum Set Cover problem and research is still going on to optimize the solution. The widely used approximate solution for the Set Cover Problem is Greedy Set Cover algorithm. This paper studies the existing algorithms of Greedy such as Set Coverage, Set Coverage with Cost, and Chvatal's Algorithm and abstracts the key points of these methods for finding minimum number of sets.

Keywords: *Approximation Algorithm, Greedy Algorithms, Optimization problem, Set Cover, Universal Set*

1.Introduction

A greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. Many Optimization problems can be solved using approximation algorithms. Some problems have no efficient solution, but a Greedy algorithm provides an efficient solution that is close to optimal. For a given set system on a universe of items and a collection of a set of items, Minimum Set Cover Problem (MSCP) [1] finds the minimum number of sets that covers the whole universe. This is a NP hard problem. The optimization has numerous applications in different areas of studies and industrial applications [2]. Bar-Yehuda and Even presented linear time approximation algorithm [3] for the weighted set-covering problem.From that all advancements were based on how to achieve better approximation ratio. Fabrizio Grandoni et al. proposed an algorithm [4] based on the interleaving of standard greedy algorithm that selects the min-cost set which covers at least one uncovered element. Fatema Akhter[5] proposed a heuristic approach to solve the problem using modified hill climbing algorithm. M. Alom, et al proved [6] the better results. Stefan Spasovski et al proposed [7] Optimization of the Polynomial Greedy Solution for the Set Covering Problem. They modified the existing greedy algorithm and obtained improvements comparing to greedy. This paper aims to study some popular existing algorithms of MSCP and analyses these algorithms for getting more ideas to do the research related with set covering problems.

The rest of the paper is organized as follows: Section 2 describes the algorithms such as Greedy Set Coverage algorithm, Greedy Weighted Set Coverage, and Chvatal et al Algorithm with examples. Section 3 analyses their performance with the particular example. Section 4 concludes the study.

2. BACKGROUND THEORY AND STUDY

Algorithms for optimization problems typically go through a sequence of steps, with a set of choices at each step. It returns near optimal solution. A greedy algorithm always makes the choice that looks best at the moment. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution. This section includes Greedy Minimum Set Cover Algorithm, Greedy Weighted Set Cover algorithm and Chvatal's Algorithm with example.

2.1 Set Cover Problem

In the set cover problem, we are given a universe U, such that |U| = n, and sets $S_1, \ldots, S_k \subseteq U$. A set cover is a collection C of some of the sets from S1, . . . , Sk whose union is the entire universe U. Formally, C is a set cover if $\cup_{Si\in C}$ Si = U. We would like to minimize |C|. Suppose we adopt the following greedy approach: In each step, choose the set S_i containing the most uncovered points. Repeat until all points are covered.

Algorithm

procedure GREEDYSETCOVER(X,F)

U $\leftarrow$ X

C $\leftarrow \phi$

while U $\neq \phi$ **do** //while uncovered elements exist

select S $\in$ F that maximizes |S $\cap$ U|

U $\leftarrow$ U - S

C $\leftarrow$ C $\cup$ {S}

end while

return C //C is the cover

end procedure

Example

Suppose S_1 = {1, 2, 3, 8, 9, 10}, S_2 = {1, 2, 3, 4, 5}, S_3 = {4, 5, 7}, S_4 = {5, 6, 7}, and S_5 = {6, 7, 8, 9, 10}. We wish to see how close a greedy algorithm comes to the optimal result. The optimal solution has size 2 because the union of sets S_2 and S_5 contains all points. The Greedy algorithm first selects the largest set, S_1 = {1, 2, 3, 8, 9, 10}. Excluding the points from the sets already selected, we are left with the sets {4, 5}, {4, 5, 7}, {5, 6, 7}, and {6, 7}.

 ISBN:978-93-87793-61-3

At best, we must select two of these remaining sets for their union to encompass all possible points, resulting in a total of 3 sets. The greedy algorithm could now pick the set {4, 5, 7}, followed by the set {6}. It gives the result as $\{S_1, S_3, S_4\}$. But, the optimal solution is $\{S_2, S_5\}$

2.2 Weighted Set Cover Problem

In the weighted set-cover problem we are given a universe U of n elements, a collection of subsets of U say $S = \{S_1, S_2 \ldots, S_m\}$ where every subset S_i has an associated cost C_i. The problem is to find a minimum cost sub collection (SOL) of S that covers all elements of U. i.e., $\Sigma_{S \in SOL}\, c_i$ is minimized.

Algorithm

procedure GREEDYMSCP with Cost(U,S, c) //Set system {U,S} and cost function, c (S)

X $\leftarrow \phi$

While $\Sigma_{i \in x}\, Xi \neq U$ **do** // Continue until X = U

Calculate cost effectiveness $\alpha = c(S)/ |S\text{-}X|$ for every unpicked set $\{S_1, S_2, \ldots, S_m\}$

// Pick a set S with minimum cost effectiveness

X $\leftarrow X \cup S$

end while

return X // Output X, minimum number of subsets

end procedure

Example: Find the set cover for following collection of sets.

U = {1,2,3,4,5}

$S = \{S_1, S_2, S_3\}$

$S_1 = \{4,1,3\}$, $Cost(S_1) = 5$

$S_2 = \{2,5\}$, $Cost(S_2) = 10$

$S_3 = \{1,4,3,2\}$, $Cost(S_3) = 3$

The minimum cost of set cover is 13 and set cover is {S2, S3}. The greedy algorithm provides the optimal solution for this example, but it may not provide optimal solution all the time. There are two possible set covers $\{S_1, S_2\}$ with cost 15 and $\{S_2, S_3\}$ with cost 13.

2.3 Chvatal's Algorithm

The greedy algorithm for weighted set cover builds a cover by repeatedly choosing a set s that minimizes the weight w_s divided by number of elements in s not yet covered by chosen sets. It stops and returns the chosen sets when they form a cover.

Algorithm

greedy-set-cover (S,w)
Initialize C $\leftarrow \phi$. Define f(C) $= \cup_{s \in C} S$

Repeat Until f(C) = f(S)

Choose $s \in S$ minimizing price per element $w_s / [f(C \cup \{s\}) - f(C)]$

Let $C \leftarrow C \cup \{s\}$

Return C

Chvatal et al [9] proved a greedy-type algorithm to approximate the SCP with a performance guarantee log|S|. Their algorithm also gives the same result for weighted set cover example such as the minimum cost of set cover is 13 and set cover is {S2, S3}.

3. Performance Analysis of three Algorithms

Three algorithms are implemented in java. The java program is downloaded from https://github.com/fawziammache/java-greedy-algorithm and it is executed on the computer with the configuration such as Intel(R) Core(TM) i3CPU, 3 GB RAM, 2.53 GHz Speed and Windows 7 Operating System. It is a menu driven program. It is very simple and easy to execute for analysing the algorithms. The execution of the program is given below for considering five elements and three sets with their cost.

JAVA SET COVER PROBLEM SOLVER

M - Enter SCP model data

P - Print SCP instance

A - Set minimum coverage percentage

C - Solve SCP with the greedy coverage heuristic

S - Solve SCP with the greedy cost heuristic

V - Solve SCP with Chvatal's algorithm

 ISBN:978-93-87793-61-3

X - Compare algorithm performance

Q – Quit

Enter choice: m

Enter number of elements (n): 5

Enter number of sets (m): 3

Set 1 details

Enter cost of set 1: 5

Enter an element covered by set 1 (0 to stop): 4

Enter an element covered by set 1 (0 to stop): 1

Enter an element covered by set 1 (0 to stop): 3

Enter an element covered by set 1 (0 to stop): 0

Set 2 details

Enter cost of set 2: 10

Enter an element covered by set 2 (0 to stop): 2

Enter an element covered by set 2 (0 to stop): 5

Enter an element covered by set 2 (0 to stop): 0

Set 3 details

Enter cost of set 3: 3

Enter an element covered by set 3 (0 to stop): 1

Enter an element covered by set 3 (0 to stop): 4

Enter an element covered by set 3 (0 to stop): 3

Enter an element covered by set 3 (0 to stop): 2

Enter an element covered by set 3 (0 to stop): 0

Enter choice: a

 ISBN:978-93-87793-61-3

Enter minimum coverage (alpha): 1

Enter choice: c

Running 'Greedy coverage heuristic'...

- Selected: Set ID: 3 Cost: 3.00 Element IDs: [1, 2, 3, 4]

- Selected: Set ID: 2 Cost: 10.00 Element IDs: [2, 5]

Done.

Greedy coverage heuristic' results:

'Greedy coverage heuristic' Time to solve: 16ms

'Greedy coverage heuristic' Objective function value: 13.00

'Greedy coverage heuristic' Coverage level: 100.00% (100.00% minimum)

'Greedy coverage heuristic' Number of sets selected: 2

'Greedy coverage heuristic' Sets selected: 2 3

Enter choice: s

Running 'Greedy cost heuristic'...

- Selected: Set ID: 3 Cost: 3.00 Element IDs: [1, 2, 3, 4]

- Selected: Set ID: 2 Cost: 10.00 Element IDs: [2, 5]

Done.

'Greedy cost heuristic' results:

'Greedy cost heuristic' Time to solve: 0ms

'Greedy cost heuristic' Objective function value: 13.00

'Greedy cost heuristic' Coverage level: 100.00% (100.00% minimum)

'Greedy cost heuristic' Number of sets selected: 2

'Greedy cost heuristic' Sets selected: 2 3

 ISBN:978-93-87793-61-3

Enter choice: v

Running 'Chvatal's algorithm'...

- Selected: Set ID: 3 Cost: 3.00 Element IDs: [1, 2, 3, 4]

- Selected: Set ID: 2 Cost: 10.00 Element IDs: [2, 5]

Done.

'Chvatal's algorithm' results:

'Chvatal's algorithm' Time to solve: 0ms

'Chvatal's algorithm' Objective function value: 13.00

'Chvatal's algorithm' Coverage level: 100.00% (100.00% minimum)

'Chvatal's algorithm' Number of sets selected: 2

'Chvatal's algorithm' Sets selected: 2 3

Enter choice: x

Alpha: 100.00%

Algorithm	Time (ms)	Obj Fn Val	Coverage (%)
Greedy coverage heuristic	16	13.0000	100.00
Greedy cost heuristic	0	13.0000	100.00
Chvatal's algorithm	0	13.0000	100.00

Overall winner: Unclear

Enter choice: q

4. Conclusion

There is no general template on how to apply the greedy method to a given problem; however the problem specification might give us a good insight. There are a lot of greedy assumptions one can make in some cases, but only few of them are correct. They can provide excellent challenge opportunities. In this paper, three algorithms are discussed and analyzed. The analysis of the results direct to do the research in finding minimum number of sets which help to solve many problems such as scheduling, mining representative pattern sets etc.

References

[1] G. Gens and E. Levner, Complexity of approximation algorithms for combinatorial problems: a survey, ACM SIGACT News, 12(3), 1980, 52-65.

[2] T. M. Chan and E. Grant, J. Knemann and M. Sharpe, Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling, Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms, 2012, 1576-1585.

[3] R.Bar-Yehuda and S. Even, A linear time approximation algorithm for the weighted vertex cover problem, Journal of Algorithms, 2,1981,198–203.

[4] F. Grandoni, A.Gupta, S. Leonardi, P. Miettinen, P. Sankowski and M. Singh, Set covering with our eyes closed, SIAM Journal on Computing, 42(3), 2013, 808-830.

[5] Fatema Akhter, A Heuristic Approach for Minimum Set Cover Problem, International Journal of Advanced Research in Artificial Intelligence, 4(6), 2015.

[6] M. Alom, S. Das and M. A. Rouf Performance Evaluationof Vertex Cover and Set Cover Problem using Optimal Algorithm DUET Journal, Vol. 1, Issue 2, June 2011

[7] Stefan Spasovski and Ana Madevska Bogdanova, Optimization of the Polynomial Greedy Solution for the Set Covering Problem, The 10th Conference for Informatics and Information Technology (CIIT 2013).

[8] V. Chvatal, Jhonson,Lovasz and Stein, Greedy Set Cover Algorithm, Encyclopedia of Algorithms, 2008,379-381.