

Modified Genetic Algorithms Based Solution to Subset Sum Problem

Harsh Bhasin
computergrad.com
Faridabad, India

Neha Singla
Student, YMCAUST
Faridabad, India

Abstract—Subset Sum Problem (SSP) is an NP Complete problem which finds its application in diverse fields. The work suggests the solution of above problem with the help of genetic Algorithms (GAs). The work also takes into consideration, the various attempts that have been made to solve this problem and other such problems. The intent is to develop a generic methodology to solve all NP Complete problems via GAs thus exploring their ability to find out the optimal solution from amongst huge set of solutions. The work has been implemented and analyzed with satisfactory results.

Keywords—subset sum problem; genetic algorithms; NP complete; heuristic search.

I. INTRODUCTION

Theoretical computer scientists are in an agreement on the issue that a minimum requirement of any efficient algorithm is that, it runs in polynomial time that is $O(n^c)$, for some constant c . But there are certain problems that cannot be solved in polynomial time. Cook, Karp, and others, defined such class of problems as NP-hard problems [1]. Some of the NP hard problems include Travelling Salesman Problem (TSP), Boolean Satisfiability problem, Subset Sum problem, Knapsack problem, Hamiltonian Path problem, Post Correspondence Problem (PCP) and Vertex Cover Problem (VCP). TSP and VCP have already been dealt with in the previous works [1], [2]. As regards PCP the solution that was proposed had some constraints but considering the fact that on analysis it gave acceptable results, it can be said that even PCP can be dealt with Genetic Algorithms (GAs) [3]. The problem that has been discussed in the following work is subset sum problem. A Genetic Algorithm based solution has been proposed and analyzed. If accepted, it will help in solving many other such problems via the concept of reducibility.

II. LITERATURE REVIEW

Many papers as regards NP Hard problems have been studied and analyzed. In the earlier works various NP Hard and NP Complete problems have been solved using GAs which has been explained below.

A. Vertex Cover Problem

A vertex cover of a graph G is a set of vertices such that each edge of G is incident to at least one vertex in the set. The resultant set is said to cover the edges of G . A minimum vertex cover is a vertex cover of smallest promising size. The vertex cover number is the size of a minimum vertex cover. This problem has been solved using GA [2], giving an

effective and efficient solution. In the process, Initial population was generated and encoding was performed on it. An index has been assigned to each chromosome and its fitness value and threshold were calculated. A mutation and crossover operator were also applied on above generated population and above process was repeated. Then reproduction was carried out giving the solution to vertex cover problem [2].

B. Post Correspondence Problem

Given a collection of dominos each of the form $[x/y]$, where x and y are strings the problem is to determine if there is a sequence of dominos that results in a match where the top string is the same as the bottom string [4]. An Artificial Intelligence based solution using GA has been used to solve the PCP problem [3]. Initial population was generated and divided into cells. Each cell was encoded and converted into 1D array. After that, each part of row of 1D array is matched with every other part of the other rows. If strings match, solution is achieved. GA operators like crossover and mutation were applied to get the desired results [3].

C. Travelling Salesman Problem

Given a list of cities and their pair wise distances, the mission is to find the shortest possible tour that visits each city exactly once still keeping the new cost minimum. TSP has been solved using randomness by applying Cellular Automata (CA) and heuristics by applying GAs. Elementary CA was generated and was reduced and analyzed, out of which some rules were considered and crossover and mutation operators were applied to reanalyze the rules giving the optimal solution to the TCP problem. From the selected rules, paths were generated and from them path with minimum cost was given as solution [1]. The solution to the above problem using GAs is also being developed in a related work.

D. Subset sum Problem

The previous attempts to solve subset sum problem have also been analyzed. In total 6 papers have been studied and analyzed. It was observed that all the implementations work well under certain constraints. The following work uses GAs based approach to find out the solution of Subset Sum Problem.

III. NP COMPLETE

The class P consists of those problems that are solvable in polynomial time. They are problems that can be solved in time

$O(n^k)$ for some constant k , where n is the size of the input to the problem [5].

The class NP consists of those problems that are "verifiable" in polynomial time. This means that if we were somehow given a "certificate" of a solution, then we could verify that the certificate is correct in time polynomial in the size of the input to the problem [6].

A problem in P is also in NP, since if a problem is in P then we can solve it in polynomial time without even being given a certificate [6]. P is subset of NP.

A problem is in the class NP-Complete if it is in NP and it is as "hard" as any problem in NP. No polynomial-time algorithm has yet been discovered for an NP-complete problem, nor has anyone yet been able to prove that no polynomial-time algorithm can exist for any one of them [6].

A problem is in class NP-Hard if the problem is "at least as hard as the hardest problems in NP". A problem H is said to be NP-hard if and only if there is a NP-complete problem L that is polynomial time Turing reducible to H. NP-hard problems can be of any type: decision problems, search problems and optimization problems [7]. Subset Sum Problem is an NP-Complete problem.

IV. GENETIC ALGORITHMS

After Genetic Algorithms (GAs) are search algorithms based on the theory of natural selection with an innovative flair of human touch. The central idea of research on GAs has been robustness. This class not only takes into account the efficiency but also efficacy [8]. The implications of robustness are the elimination of costly resigns and higher level of adaptations.

The depiction of a natural population is done using, what is called chromosomes which are nothing but a set of numbers, generally binary. Each number represents a cell and can be perceived as an affirmative or negative answer. For example, a chromosome 10110 if applied to knapsack problem can be assumed as selecting the first, third and fourth item from amongst a set of five items, as we have 1 at the first, third and fourth position. The initial population can be generated using any Pseudo Random Number Generator. Each chromosome is then assigned a fitness value. Based on this fitness value replication is done as explained in the following Table 1.

TABLE I. ROLLET WHEEL SELECTION IMPLEMENTATION

| Chromosomes | Frequency | Cumulative Frequency |
|--------------|-----------|----------------------|
| Chromosome 1 | 33 | 33 |
| Chromosome 2 | 24 | 57 |
| Chromosome 3 | 17 | 74 |
| Chromosome 4 | 10 | 84 |
| Chromosome 5 | 16 | 100 |
| Total | 100 | |

Now generate a random number % 100. Let it be 63. Now Cumulative Frequency 63 lies in Chromosome 3. Therefore, Chromosome 3 is replicated.

The above population is enhanced by using basic operations like crossover and mutation.

A. Crossover

Crossover operator has the significance as that of crossover in natural genetic process. In this operation two chromosomes are taken and a new is generated by taking some attributes of first chromosome and the rest from second chromosome. In GAs a crossover can be of following types

1) *Single Point Crossover*: In this crossover, a random number is selected from 1 to n as the crossover point, where n being the number of chromosome. Any two chromosomes are taken and operator is applied.

2) *Two Point Crossover*: In this type of crossover, two crossover points are selected and the crossover operator is applied.

3) *Uniform Crossover*: In this type, bits are copied from both chromosomes uniformly.

B. Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of population to the next. It is similar to biological mutation [9]. Mutation allows the algorithm to avoid local minima by preventing the population chromosomes from becoming too similar to each other [10]. GAs involves string-based modifications to the elements of a candidate solution. These include bit-reversal in bit-string GAs or shuffle and swap operators in permutation GAs [2], [3].

C. Selection

It is quantitative criterion based on fitness value to choose which chromosomes from population will go to reproduce. Intuitively the chromosome with more fitness value will be considered better and in order to implement proportionate random choice, Roulette wheel selection is used for selection [2], [3], [11].

GAs are different from the other search processes owing to the fact that they work on coding of the parameter set and not on the parameters [12]. It is also general belief that GAs use payoff and not auxiliary knowledge. Moreover, determinism is not needed in GAs.

The initial population for GAs is generated by applying the following procedure.

Initial population is stored in a 2D array, let it be called `init_pop[][]`.

for $i = 0$ to n

begin

for $j = 0$ to m

begin

Generate a random number x modulo 100

if $(x \leq 50)$ then `init_pop[i][j] = 0`

else `init_pop[i][j] = 1`

end

end

V. PROPOSED WORK

The proposed work is based on the premises that GAs imitates the process of natural selection in robust and efficient manner. In the work, the list of numbers on which subset sum is to be applied is told in the array A []. s denotes the expected sum.

The array is sorted since sorting the array by quick sort has the complexity of $O(n \log n)$. Even if this is taken into consideration then also the proposed solution gives the better result than the existing ones, since subset sum is an NP Complete problem. The various steps of the proposed algorithm have been explained below.

A. Sorting

The given array A[] is sorted.

B. Calculating Limit Point

The following procedure is applied to find out the limit point, where the limit point is defined as the point in the array after which the members need not to be considered.

```
For i = 0 to n
begin
if(A[i] > s) then take i as the position of the limit point and
break;
end
```

If no limit point is found then all the array elements need to be considered.

C. Generating Initial Population

Generate Genetic Population as explained in section IV.

D. Mapping

Mapping of genetic population to the array till the limit point is performed as follows

```
Let A = {1, 3, 4, 5, 9, 10}
i = 0 1 2 3 4 5
```

Let s = 6

The limit point become i = 4

Now/ the numbers {1, 3, 4, 5} needs to be considered. Let's call this modified array.

Let us assume that the chromosome of genetic population is 101101

Now take that many cells that are equal to the length of modified array.

i.e. 1011 is considered.

Now, 1 denotes accepting the element and 0 denotes not accepting the element.

As per the above chromosome items {1, 4, 5} are selected.

E. Calculating the Sum

Calculate the sum of selected items. For above chromosome, sum comes to be 10.

F. Reducing the Population

Accept or Reject the chromosome on the following basis.

if(sum > s) then reject the chromosome

else accept the chromosome.

G. Crossover

Crossover operator is applied on the above reduced population and step E. and F. are repeated.

H. Mutation

Mutation operator is applied on the population obtained above.

I. Moderation

The above process is used for small values of s. For large values moderation is used which is explained as follows:

Initial Population generated in Step C. is considered and fitness value is calculated for each chromosome using the formula,

$$\text{Fitness} = (1 / (1 + N1)) * 100$$

Where, N1 = Number of One's in second half of chromosome.

High fitness indicates that the chromosome is more relevant. Roulette Wheel Selection is applied on initial population based on the fitness value.

The above process is explained in the figure 1.

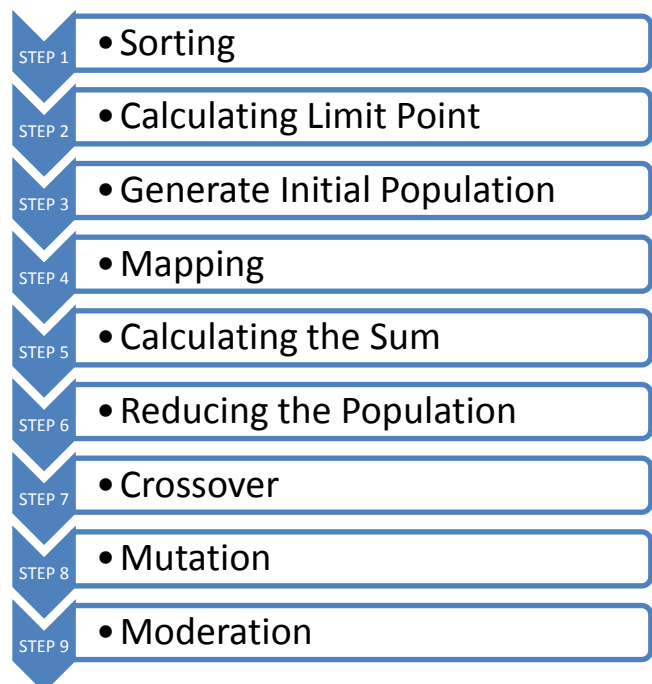


Figure 1. Modified Genetic Algorithms Based Subset Sum Solution

VI. CONCLUSION AND FUTURE SCOPE

In the analysis, 30 items were considered and were randomly generated such that the maximum number generated was 100. Analysis was done for various values of the factor s. For small values of s, i.e. less than 30, limit is calculated i.e. items which can contribute to this sum were taken and rest were not. Genetic process was applied and results were analyzed. The results have been shown in Table 2 and Fig. 2

In Fig. 2, a) Represents the percentage of sample runs which give the accurate result, b) represents the percentage of sample runs which give optimal solution and c) represents the cases where required sum is not possible, but gives the best possible result.

If the value of sum is taken as 60 or greater in this experiment, then the results were not satisfactory. On analysis, it was found that this is due to fact that the number of one's in the right half are more. So as to handle this situation, a process of moderation was applied. The results obtained have been listed in Table 3.

The rate of replication was taken as 4% and Roulette Wheel Selection was applied which resulted in the replication of favorable data, thus making the population fitter.

The overall results are encouraging. Some of them have also been shown in the following figures and tables.

It must also be remembered that it is not always the case the a finite solution to subset sum problem can be found. Moreover GAs give best solutions but are not guaranteed to do so always.

TABLE II. DATA FOR ANALYSIS 1

| | |
|-----------------|----|
| Number of items | 30 |
| Required Sum | 20 |

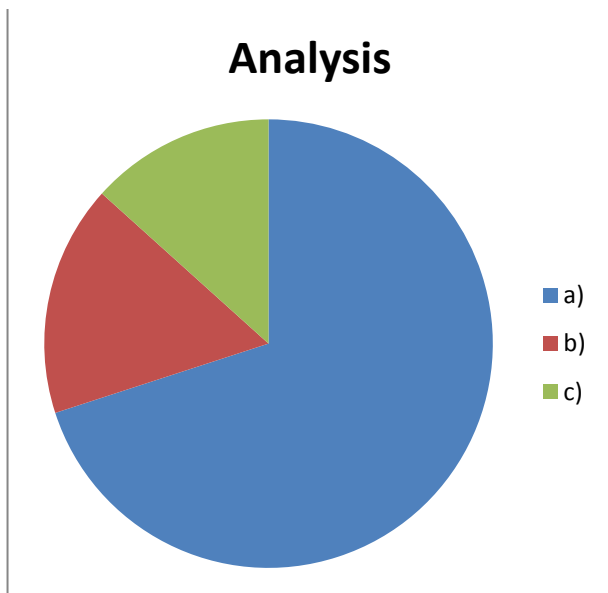


Figure 2. Analysis 1

TABLE III. RESULTS

| Fitness Value | Number of chromosomes in initial population having this fitness value | Number of chromosomes in replicated population having this fitness value |
|---------------|---|--|
| 25 | 2 | 9 |
| 20 | 11 | 36 |
| 16 | 16 | 44 |
| 14 | 29 | 70 |
| 12 | 42 | 96 |
| 11 | 36 | 74 |
| 10 | 36 | 60 |
| 9 | 19 | 36 |
| 8 | 5 | 8 |
| 7 | 4 | 7 |

REFERENCES

- [1] Harsh Bhasin and Neha Singla, "Harnessing Cellular Automata and Genetic Algorithms to solve Travelling Salesman Problem", communicated.
- [2] Harsh Bhasin and Gitanjali, "Harnessing Genetic Algorithm for Vertex Cover Problem", International Journal on Computer Science and Engineering (IJCSSE), Vol. 1, Issue 2, pp. 218 - 223.
- [3] Harsh Bhasin and Nishant Gupta, "Randomized algorithm approach for solving PCP", IJCSSE 2012; Vol 4, Issue 1, pp. 106-113.
- [4] Nasseef Abukamail, "Specialized Version of the Post Correspondence Problem (PCP)", Presented to: Dr. David Juedes CSS06, Computation Theory, 1988.
- [5] P C. Bell and Igor Potapov, "The Identity Correspondence Problem and its Applications", 2010.
- [6] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, "Introduction to Algorithms", the MIT Press, Cambridge, Massachusetts, USA, 2009.
- [7] Michael R. Garey and David S. Johnson (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness". W.H. Freeman.
- [8] Cormen, Thomas H.; Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford (2001) [1990]. "35.5: The subset-sum problem". Introduction to Algorithms (2nd ed.). MIT Press and McGraw-Hill. ISBN 0-262-03293-7.
- [9] H. Bhasin and S. Bhatia, "Application of Genetic Algorithms in Machine learning", IJCSIT, Vol. 2 (5), 2011.
- [10] Pisinger D (1999). "Linear Time Algorithms for Knapsack Problems with Bounded Weights". Journal of Algorithms, Volume 33, Number 1, October 1999, pp. 1-14
- [11] Harsh Bhasin, "Use of Genetic Algorithms for Finding Roots of Algebraic Equations", IJCSIT, Vol. 2, Issue 4.
- [12] S. Thrun, "Learning to Play the Game of Chess", In G. Tesauro, D. Touretzky, and T. Leen, editors, Advances in Neural Information Processing Systems(NIPS) 7, Cambridge, MA, 1995. MIT Press.

AUTHORS PROFILE

Harsh Bhasin has completed his B. Tech in Computer Science and Engineering, M. Tech (C. E.).He is presently working in computergrad.com, proprietor Sahib Soft and a member of International Association of Computer Science and Information Technology. He has published many papers in the domain of Random Number generation and NP Hard Problem by Cellular Automata; Cryptography, Machine Learning, Equation Solving and NP Hard Problems by Genetic Algorithm; conversion of south Asian languages by Natural Language Processing. His various papers have been published in IJCA online, IJCSIT, IJCSST, IJCSSE, IJCNWC, IJIEE , IJCSI (To be published on 31st March) and IEEE and National and International conferences.

Neha Singla is a student of M. Tech. , YACA UST , Faridabad.