Volume 6, Issue I, Jan 2018
eISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Genetic approach to solve non-fractional knapsack problem

S. M Farooq[1], G. Madhavi[2] and S. Kiran[3]

[1,2,3] Y. S. R Engineering College, Yogi Vemana University
Korrapad Road, Proddatur 516360, India
[1]*shaikfaroq@gmail.com*,[2]*godi.madhavi@gmail.com*,[3]*rkirans125@gmail.com*

Abstract: *Non-fractional knapsack problem is an age old combinatorial optimization problem which is studied frequently by the researchers in the literature. In the knapsack problem we must maximize the benefits of objects in a knapsack without exceeding its limit. Solving the optimization problems like knapsack using evolutionary approach gives much improved results than traditional approaches such as greedy method and dynamic programming. Results are generated with the implementation of the knapsack problem using genetic operators in C language. The strategic selection of parameters of Genetic operators and its values gives improved results comparing to previous work done on knapsack with genetic approach.*
Keywords: *Evolutionary Algorithms, Knapsack Problem, crossover rate, mutation rate.*

1. INTRODUCTION

Genetic algorithms are broad class of global optimization methods. They applied on many combinatorial optimization problems [1]. They are intelligent probabilistic search algorithms, random and adaptive in nature and part of evolutionary computing which is based on the process of natural evolution "survival of fittest" theory [2-3]. A genetic algorithm works on population of individuals that are re-produced according to evolution function formed which is also called as fitness function and finds solution to a problem by applying genetic operators such as crossover and mutation. Crossover which is also known as recombination operation is the key operation that defines performance of the genetic algorithms. The larger the fitness value is the better fitness function has. The performance of GA differs when we apply different crossover operators. The genetic operations are used in cryptanalysis of cryptographic algorithms [4]. The previous study about genetic algorithms with respect to crossover operator re-examined in [5] and concluded that the genetic algorithms with crossover on larger population size gives better results. The size of population influences the performance of genetic algorithms [6].

A knapsack problem is a combinatorial optimization problem in which we must maximize the benefits of objects in a knapsack without exceeding its limit [7]. The present paper organized as follows. Section 2 outlines the research carried out on finding the solution of a knapsack problem. Section 3 explains about the traditional knapsack problem. Section 4 describes about fundamentals of Evolutionary approach such as genetic operators. We implemented the knapsack problem using genetic operators in C programming language. Section 5 presents the results. Graph was plotted to show the characteristic of crossover operation along with mutation operation. Finally, section 6 conclusion is presented along with future recommendations. The strategic selection of Genetic operators such as crossover and mutation give improved results comparing to traditional and other techniques.

2. LITERATURE REVIEW

The age-old knapsack problem is studied much in the literature to find the solution with different approaches. In [8], the 0/1 knapsack problem is solved using evolutionary algorithm with schema replacement technique which gives better results compared to greedy and simple evolutionary approaches. Schema replacement technique depends on schema order and schema defining length. Low schema order, short defining length and higher fitness will create schema which are high schema order, long defining length and higher fitness, then make the global optimal solution. In [9], Genetic algorithm is applied on Portfolio optimization problem which is a kind of knapsack problem to find expected returns subject to the risk factor. Genetic operators define the performance of the genetic algorithms. Good selection method and elitism are very important for the good performance of a genetic algorithm, but different crossover ratios and single and double crossover points applied on 0/1 knapsack problem gives the results almost same [10].

Mutation is one of the important parameters of genetic algorithms since the type of mutation operation has more effect on the results generated. Different mutation operations applied on different problems like Travelling Sales Person problem (TSP), 0/1 knapsack problem, Shubert function [11]. Insertion mutation, boundary and non-uniform mutation operators are best suitable for TSP and Shubert function respectively. For 0/1 knapsack problem all mutation operators give approximately same results. In genetic algorithms the process of selection is based on the principle of "survival of fittest". Higher fitness values will go through for reproduction by discarding lower fitness value. The 0/1 knapsack is a NP complete problem but after applying genetic algorithm on it the solution gets in polynomial time [12]. In [13], the 0/1 knapsack problem is solved using ant colony optimization technique and compared with genetic algorithms for a single objective problem.

3. KNAPSACK PROBLEM

Knapsack problem is a popular class of optimization problems which maximizes the benefit of objects by not exceeding its knapsack capacity. In many applications, knapsack problem is formulated to solve like resource allocation, demand

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

forecasting, portfolio selection, network flows etc. Knapsack problem also used in efficient energy consumption in smart grids [14]. The model of the general non-fractional (0/1) knapsack problem can be described accordingly. Let the weights and profits of n objects are wi and pi respectively. The capacity of knapsack is c. Then, which objects we must select completely to keep inside the knapsack so that profit will be maximized is the problem to be stated.

$$Max \sum_{i=1}^{n} p_i * x_i \quad (1)$$

$$Subject\ to \sum_{i=1}^{n} w_i * x_i \leq c$$

Example of 0/1 KP

If we have a knapsack with capacity of 13 cubic inches and several items with different weights and profits. The problem statement is we must include the only items which gives more profit by with the limit of knapsack capacity. There are four items labeled say, A, B, C and D. The respective profit and weight values taken for illustration of knapsack problem.

Table 1: Knapsack Problem Sample Data

Item	A	B	C	D
Profit	3	2	4	6
Weight	7	5	7	4

We must maximize the total profit,

$$Max(3 * x1 + 2x2 + 4 * x3 + 6 * x) \quad (2)$$

Subject to the constraint,

$$7 * x1 + 5 * x2 + 7 * x3 + 4 * x4 \leq 13$$

For the above problem we have 2^4 possible items:

Table 2: Search space to find max profit

A	B	C	D	Weight	Profit
0	0	0	0	0	0
0	0	0	1	4	6
0	0	1	0	7	4
0	**0**	**1**	**1**	**11**	**10**
0	1	0	0	5	2
0	1	0	1	9	8
0	1	1	0	12	6
0	1	1	1	16	-
1	0	0	0	7	3
1	0	0	1	11	9
1	0	1	0	14	-
1	0	1	1	18	-
1	1	0	0	12	5
1	1	0	1	16	-
1	1	1	0	19	-
1	1	1	1	23	-

The solution is bolded in the above table, profit is 10.

4. GENETIC ALGORITHMS

With the advancement of the technology software complexity and requirements has been changed in past decades. To develop efficient application, software developers need to follow certain set of rules. Before developing any application, developer needs to do requirement analysis. If system developed as per the requirements and software development rules are strictly followed, then there will be a more chance that efficient software will developed.

Genetic Algorithms (GAs) are randomized search techniques tries to find global optima. It is a stochastic process in which strings representing solution space are chosen and combined. GAs depends on objective knowledge not on auxiliary knowledge. The given problem is encoded in different forms like binary, permutation, value and tree encodings. The encoded form is treated as chromosomes. A chromosome is randomly chosen to point in solution space. The strength of the genetic algorithms lies in genetic operators. They are selection, crossover and mutation.

Selection method

Selection methods gives preference to better solution points in the search space based on objective function (fitness) forwards them for further operations. We have many selection methods like roulette wheel selection, rank selection, steady state selection, Boltzmann selection and tournament selection.

Crossover operation

Crossover operation also known as recombination operation applied on randomly chosen chromosomes initially at single location or multiple locations which is treated as single point crossover and multi-point crossover operation. For example, if the given problem is represented in binary encoded form and single point crossover is applied on a location, say 3 on the two parent chromosomes.

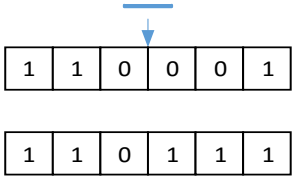

Figure 1: Chromosomes before applying single point cross over.

The bit stream after the given single point or position will be swapped. The resulting offspring chromosomes after single point crossover operation.

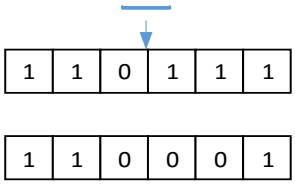

Figure 2: Chromosomes after applying single point cross over.

The double crossover operation, say two positions are taken at 2 and 4 respectively in the example.

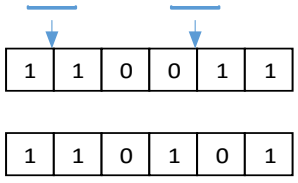

Figure 3: Chromosomes before applying two-point cross over

The resultant chromosomes after applying two-point crossover operation swaps the bits after the given positions.

Volume 6, Issue I, Jan 2018
eISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

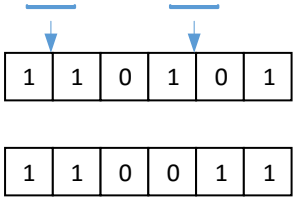

Figure 4: Chromosomes after applying double cross over.

The same process applies to multiple positions by increasing the size of chromosomes called as multipoint crossover operations.

Mutation Operation

A bit value position is randomly chosen to change the bit value. For example, by considering the following chromosome, the mutation point is, say 4 changes the offspring.

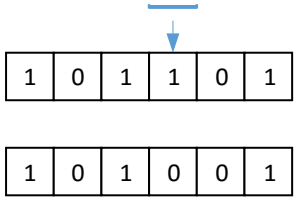

Figure 5: Chromosomes before and after applying flip bit mutation operation.

The mutation operation prevents falling all solutions in the population into a local optimum. Genetic Algorithms starts with selecting random population of chromosomes and selects two parent chromosomes based on fitness value calculated. The algorithm proceeds with two major genetic operator's crossovers followed by mutation to generate new offspring. The algorithm repeats until an offspring is generated with good fitness value. The flow chart of genetic algorithm to find the best chromosomes is depicted in the Figure 6.

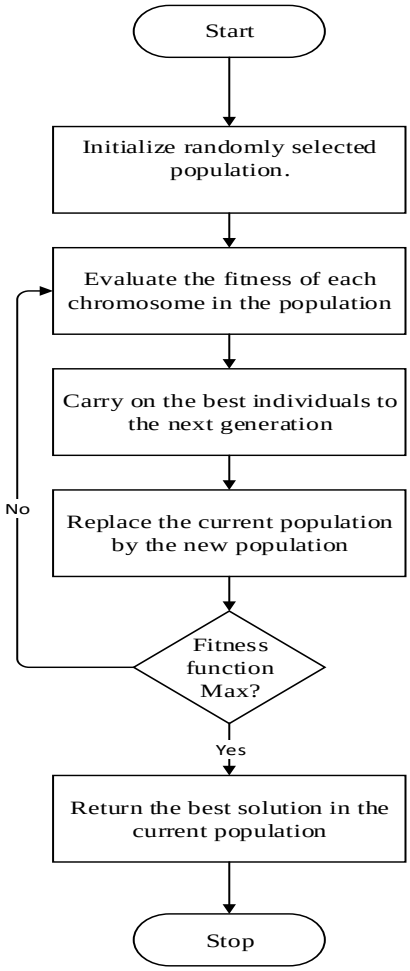

Figure 6: Flow Chart of genetic algorithm

5. IMPLEMENTATION

chromosome representation

The chromosome representation and genetic operators such as crossover and mutation are crucial steps in the performance of the genetic algorithm. The chromosome can be represented in array form and in linked list form to implement.

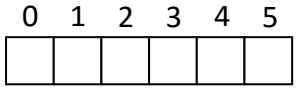

Figure 7: Array representation of chromosomes

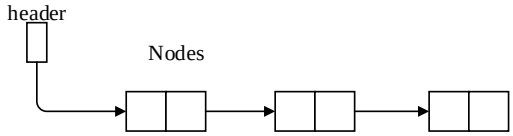

Figure 8: Linked List representation of chromosomes

In the current implementation we have taken array representation. The non-fractional (0/1) problem is encoded in binary form. Chromosome is encoded in the form of sequence of 0 and 1, where 1 indicates that the respective item is kept inside the knapsack and 0 indicates that item is not included in the knapsack.

Table 3: Binary form encoding

Item	1	2	3	4
Chromosome	0	1	0	1
Included or Not Included	No	Yes	No	Yes

Parameters taken for the genetic algorithm.

The parameters chosen is population size, crossover and mutation rates and finally iteration bound. The important parameters are crossover and mutation rates. The adaption of values for these two parameters defines performance of genetic algorithms. There is a slight difference between cross over, mutation probabilities and crossover, mutation rates.

Volume 6, Issue I, Jan 2018
eISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Crossover and mutation probabilities are the probabilities of the respective operators used whereas crossover rate is the likelihood of number of bits on which crossover operation being performed and mutation rate is the likelihood of a single gene changing. In our paper for different values of crossover rate and fixed mutation rate results are generated and compared.

Table 4: Parameters considered for implementation

Parameter	values
Population size	20
Cross over rate	0.2-0.9
Mutation rate	0.1
Iteration bound	20

Generated results

Initial 20 chromosomes are randomly selected for 10 items to be placed in knapsack of size 20. The fitness function is represented as maximization of profit values with in the constraint of knapsack capacity. The genetic operators like crossover and mutation was implemented in C language.

The program implemented takes the input of 20 random chromosomes and applies the crossover operation with different rates followed by a mutation operation. The generated results are summarized in the following tables numbers 5 to 14.

Table 5: Initial chromosomes

S.NO	Chromosome	Profit	Weight
1	1100101101	24	27
2	0011001101	21	24
3	0010110010	25	21
4	**1011110000**	**30**	**25**
5	0100100111	18	24
6	0101011001	25	26
7	0100111000	26	15
8	0101110000	25	16
9	1000111100	30	18
10	1001011100	28	20
11	1000000111	11	24
12	1001001100	19	25
13	1100001101	16	25
14	1100000000	5	12
15	1000100000	11	9
16	0101011000	24	17
17	1100000010	9	19
18	0000100011	13	18
19	1111000101	19	33
20	1100000111	13	29

Table 6: Child chromosomes

S.NO	Chromosome	Profit
1	0100111000	26
2	0101110000	25
3	1000111100	30
4	1001011100	28
5	1001001100	19
6	1100000000	5
7	1000100000	11
8	0101011000	24
9	1100000010	9
10	0000100011	13

For the randomly selected 20 chromosomes we have selected 10 chromosomes based on fitness function, then the resultant chromosomes are shown in Table 6.

Child chromosomes after applying cross over and mutation operations with 50% and 10% crossover and mutation rates respectively. The resultant chromosomes are shown in Table 7. It is observed that crossover operation with different rates gives improved results than applying different types of mutation operations.

Table 7: Child chromosomes

S.NO	Chromosome	Profit
1	0100010000	11
2	0101011000	24
3	1000011100	22
4	1001100000	17
5	1100101100	23
6	1000011000	19
7	0101100000	16
8	0000000011	5

Child chromosomes after applying cross over and mutation rates with 40% and 10% respectively.

Table 8: Child chromosomes

S.NO	Chromosome	Profit
1	0100100000	10
2	0101101000	23
3	1000101100	21
4	1001001100	19
5	1001010000	18
6	1000111000	27
7	0101000000	8

Child chromosomes after applying cross over and mutation rates with 30% and 10% respectively.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Table 9: Child chromosomes

S.NO	Chromosome	Profit
1	0100110000	19
2	0101111000	32
3	1000110100	23
4	1001010100	21
5	1100001100	15
6	1000101000	18
7	0101010000	17
8	0000101011	20

Child chromosomes after applying cross over and mutation rates with 20% and 10% respectively.

Table 10: Child chromosomes

S.NO	Chromosome	Profit
1	0100111100	29
2	0101110100	28
3	1000111000	27
4	1001011100	25
5	1001001000	16
6	1100000100	8
7	1000100100	14
8	0101011100	27
9	1100000110	12
10	0000100110	15

Child chromosomes after applying cross over and mutation rates with 90% and 10% respectively.

Table 11: Child chromosomes

S.NO	Chromosome	Profit
1	1100110000	22
2	0001011100	25
3	0000111100	27
4	0100001100	12
5	0001000000	6
6	0101010000	17
7	1000101000	18
8	0000100010	12

Child chromosomes after applying cross over and mutation rates with 80% and 10% respectively.

Table 12: Child chromosomes

S.NO	Chromosome	Profit
1	1010000100	10
2	0110100000	14

Child chromosomes after applying cross over and mutation rates with 70% and 10% respectively.

Table 13: Child chromosomes

S.NO	Chromosome	Profit
1	1010000100	10
2	0110100000	14

Child chromosomes after applying cross over and mutation operations with 60% and 10% crossover and mutation rates respectively.

Table 14: Child chromosomes

S.NO	Chromosome	Profit
1	0101110000	25
2	0100111000	26
3	1001011100	28
4	1000111100	30
5	1000000100	6
6	1101001000	18
7	1001011000	25
8	0100100000	10
9	0001000011	11

The above sample results on crossover rates between 0.2 to 0.6 shows that the problem slowly will move towards to convergence. So lower crossover rates give better results.

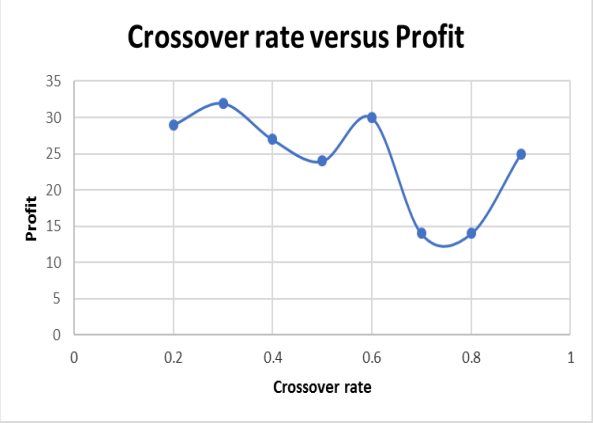

Figure 8: Crossover versus profit convergent graph.

6. CONCLUSION

The age-old knapsack problem is a combinatorial optimization problem studied and attempted to find the solution for the problem using genetic algorithm by implementing in c programming. Genetic operators play key role in defining the performance of the genetic algorithms. In this paper, we have applied crossover and mutation rates to get the solution. The results generated shows that crossover operation with lower rate reaches to quick convergent than higher rates. The paper can be further extended by different mutation operators on the knapsack problem and other optimization problems to find the performance of genetic algorithms.

Volume 6, Issue I, Jan 2018
eISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

REFERENCES

[1] Jorge Magalhaes-mendes, Dr. Antonio Bernardino de Almeida "A comparative Study on crossover operators for genetic algorithms to solve the job shop scheduling problem", WSEAS Transactions on computers, E-ISSN: 2224-2872, Issue 4, Volume12, and April 2013.

[2] Holland J H. Adaptation in natural and artificial systems [M], AnnArbor: University Michigan Press, 1975.

[3] David E. Goldberg, Addison Wesley, 1990. "Genetic Algorithms in Search, Optimization and Machine Learning".

[4] S. M Farooq, S. Kiran, G. Iliyas, "A study on techniques that improve the strength of encryption algorithms based on data structures", International conference on recent trends in engineering science, humanities and management, ISBN: 978-93-86171-18-4, 2017.

[5] William M. Spears, Vice Anand, "A study of crossover operators in genetic programming" ISMIS'91 proceedings of the 6th international symposium on methodologies for the intelligent systems, pp.409-418, Springer-Verlag London, UK© 1991, ISBN: 3-540-54563-8.

[6] De Jong, K. A. and William M. Spears (1990). An analysis of the interacting roles of population size and crossover in genetic algorithms, in the international workshop parallel problem from nature, university of Dortmund, Oct 1-3, 1990.

[7] Ellis Horowitz, Sartaj Sahni, S Rajasekaran, "Fundamentals of Computer Algorithms", University Press, Second Edition (2008).

[8] Kangshun Li, Yuzhen Jia, Wensheng Zhang, Yang Xie, "A New Method for solving 0/1 knapsack problem based on evolutionary algorithm with schema replaced" Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September 2008.

[9] Derya TURFAN, Cagdas Hakan ALADAG, Ozgur YENIAY, " A new genetic algorithm to solve knapsack problems" Journal of social and economic statistics No.2, Vol.1, Winter 2012 published by JSES.

[10] Maya Hristakeva, Dipti Shresta, "Solving 0/1 Knapsack Problem with Genetic Algorithms".

[11] Basima Hani Hasan, Moutaz Saleh Mustafa, "Comparative study of mutation operators on the behavior of genetic algorithms applied to non-deterministic (NP) problems", 2011 Second International Conference on Intelligent Systems, Modelling and Simulation", 978-0-7695-4336-9/11 © 2011 IEEE, DOI:10.1109/ISMS.2011.11

[12] Charu Sachdeva, Shivani Goel, "An improved approach for solving 0/1 knapsack problem in polynomial time using genetic algorithms", IEEE international conference on recent advances and innovations in engineering, 978-1-4799-4040-0/14 ©2014 IEEE.

[13] Sourav Samanta, Sayan Chakraborty, Suvojit Acharjee, Aniruddha Mukherjee, Nilanjan Dey, "Solving 0/1 knapsack problem using Ant Weight Lifting Algorithm", IEEE International conference on computational Intelligence and computing research, 978-4799-1597-2/13 © 2013 IEEE.

[14] Omid Ameri Sianaki, Omar Hussain, Azadeh Rajabian Tabesh, "A Knapsack problem approach for achieving efficient energy consumption in smart grid for end user's life style", IEEE conference on Innovative technologies for an efficient and reliable electricity supply (CITRES), DOI: 10.1109/CITRES.2010.5619873, 2010©IEEE.