2009 International Conference on Computer Engineering and Applications
IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore

A Evolutionary Multiobjective Genetic Algorithm to Solve 0/1 Knapsack Problem

Subhra Swetanisha [1], Prof. Bhabani Sankar Prasad Mishra [2]
[1] Lect. Comp.Sc Dept , Trident Academy Of Technology, Bhubaneswar
[2] KIIT University, Bhubaneswar
[+] Corresponding author. *E-mail address*: swetanisha.subhra@gmail.com

Abstract. The 0/1 Knapsack Problem is very well known and it appears in many real life world with different application. The solution to the multi objective 0/1 Knapsack problem can be viewed as the result of a sequence of decisions. The problem is NP - complete and it also generalization of the 0/1 Knapsack problem in which many Knapsack are considered. A evolutionary algorithm for solving multi objective 0/1 Knapsack Problem is introduced in this paper. This algorithm used a genetic algorithm for direct comparison of two solutions. Our motivation for developing a genetic algorithm in this paper in a multi objective optimization framework was that (a) GA is robust search method. (b) GA performs the global search. (c) GA already work with a population of candidate solutions, which makes them naturally suitable for multi objective problem solving, where this algorithm is used to consider a set of optimal solutions at each iteration. Few numerical experiments are realized using the best and recent algorithm in this paper. Experimental outcome show that the new proposed algorithm performs better.

Keywords: 0/1 Knapsack Problem, Evolutionary multi objective optimization. Genetic algorithm, NP completeness.

1. Introduction

Due to practical importance the 0/1 Knapsack Problem is widely used. In last few years the generalization of this problem has been studied and many algorithms have been proposed. Evolutionary approach for solving the multi-objective 0/1 Knapsack Problem is one of them, many real worked papers founded in the literature about multi-objective Knapsack Problem and about the algorithms introduced for solving them. ([6], [7], [8], [9]).

In this paper we introduced a new evolutionary approach for multi objective 0/1 Knapsack Problem. We use the genetic algorithm in order to determine the solution quality.

2. Problem Description

The knapsack problem [8] [9] is a problem in combinatorial optimization. Given a set of items, each with a cost and a value, determine the number of each item to include in a collection so that the total cost is least than a given limit and the total value is as large as possible.

We have n kind of items. 1 through n, each item i has a value Pi and a weight Wi. The maximum weight that we can carry the knapsack is C.

The 0 – 1 knapsack is a special case of the original knapsack problem in which each item of input can not be subdivided to fill a container in which that input partially fits. The 0 – 1 knapsack problem restricts the number of each kind of item x_j to zero or one. Mathematically the 0 – 1 knapsack problem can be formulated as

Maximize $\sum_{i=1}^{n} Pi\ Xi$ Subject to $\sum_{i=1}^{n} Wi\ Xi \leq C$

Example:

Item:	1	2	3	4	5	6	7
Profit:	10	8	9	15	7	7.2	5.5
Weight:	12	8	6	16	4	5	8

Suppose seven numbers of items are there. We have to select such items so that it will fulfill our two objectives as follows.

1. Fill it to get the maximum profit.
2. Knapsack holds a maximum of 22 pounds. So the total weight of the selected items should not exceed our maximum capacity.

3. Proposed Multi-objective Algorithm

1. Encoding: represent the individual.
   Generate random population of n chromosomes (suitable solutions for the problem).
2. Fitness: Evaluate the fitness of each chromosome.
3. [New population] repeating following steps until the new population is complete.
3. Selection: Select the best two parents.
4. Crossover: cross over the parents to form a new offspring (children).
5. Mutation: with a mutation probability
6. Accepting: Place new offspring in a new population.
7. Replace: Use new generated population for a further run of algorithm.
8. Test: If the end condition is satisfied, then stop.
9. [Loop] Go to step 2.

4. Pseudo Code

```
begin
      Initialize population;
      for i = 1 to iterations_no do
begin
    for m = 1 to populationsize do
                for n = 1 to knapsacks_no do
    if exceed(population[m], n)
then discard(population[m], n)
    Find non dominated solutions from population;
for selection_no = 1 to populationsize / 2 do
begin
 Random select two individuals population[individual1] and population[individual2];
   Crossover(population[individual1], population[individual2]);
   Two offspring offspring1 and offspring2 are obtained;
The offspring enter the population of the new generation;
          end;
 end;
```

5. Illustration

5.1 Encoding

In the encoding scheme, randomly the items are selected. If the item is selected then that is represented as 1 and if item is not selected then that is represented as 0.

Encoding: 0 = not exist, 1 = present in the Knapsack
Chromosome: 0011010

Item	1	2	3	4	5	6	7
Chromosome	0	0	1	1	0	1	0
Present	n	n	y	y	n	y	n

From the above figure the items selected are 3, 4, 6.

Suppose randomly populations are generated and are represented as chromosomes. Generate random population of n chromosomes:

a) 1101000
b) 0100101
c) 0010101

5.2 Fitness and Selection

For the first chromosome 1101000, the total profit and weights are to be calculated.

Item	1	2	3	4	5	6	7
Chromosome	1	1	0	1	0	0	0
Profit	10	8	9	15	7	7.2	5.5
Weight	12	8	6	16	4	5	8

The total profit = (10 + 8 + 15) = 33
and total weight = (12 + 8 +16) = 36

Then we have to check whether these items can be selected or not. As the weight exceeds our maximum capacity so we can not select these items for further steps.

For the second chromosome i.e. for 0100101

Item	1	2	3	4	5	6	7
Chromosome	0	1	0	0	1	0	1
Profit	10	8	9	15	7	7.2	5.5
Weight	12	8	6	16	4	5	8

The total profit = (8 +7+ 5.5) = 20.5
and total weight = (8 + 4 + 8) = 20

Here these set of items can be selected as weight is less than our maximum capacity i.e. 22.

Similarly for the third chromosome 0010101

Item	1	2	3	4	5	6	7
Chromosome	0	0	1	0	1	0	1
Profit	10	8	9	15	7	7.2	5.5
Weight	12	8	6	16	4	5	8

The total profit = (9 + 7 + 5.5) = 21.5

and total weight = (6 + 4 + 8) = 18

From the above discussion it is clear that the second and third chromosomes are selected as parent upon which the crossover will be operated in order to get the child chromosome.

5.3 Crossover and Mutation

The selected chromosomes are

0 1 0 0 | 1 0 1 parent 1

0 0 1 0 | 1 0 1 parent 2

After crossover the resultant offspring will be

0 1 0 0 1 0 1 child 1

0 0 1 0 1 0 1 child 2

if we put the mutation over the child 2 and on the last bit then 1 will be converted to bit 0.

Before mutation : 0 0 1 0 1 0 1

After mutation: 0 0 1 0 1 0 0

After the crossover is finished, the new offspring are kept in the new population.

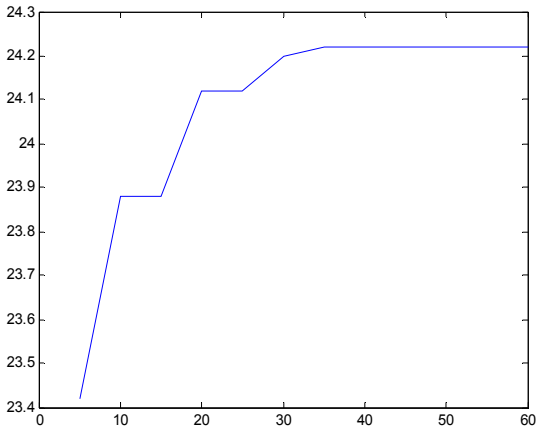
If the end condition is there then the process will stop there otherwise again the fitness calculation and selection will occur to find out the better solution [9] [10].

6. Experimental Results

We will test our algorithm considering a knapsack having 7 items. The results obtained are compared with the result obtained by Greedy Programming. The result is obtained as follows:

The number of iteration is 1000.

The results obtained in 40 runs are average.



The distance is kept for each individual is minimum. The average of these distances represents the measure of convergence towards the optimization.

7. Conclusion

In this paper we have proposed a most effective evolutionary algorithm which is derivative free stochastic optimization methods based loosely on the concept of natural selection and evolutionary process. For solving 0/1

multi objective knapsack problem. This algorithm uses the genetic algorithm concept. For these experiment one knapsack and 5 items are consider a comparison to other existing method is also consider.

8. References

[1]. Ingargiola, G. P., Korsh, J. F. A reduction algorithm for zero-one single knapsack problems, Management Science 20, 460-463, 1975;

[2]. Grosan, C., Oltean, M., Dumitrescu, D. Performance Metrics for Multiobjective Optimization Evolutionary Algorithms, In Proceedings of Conference on Applied and Industrial Mathematics (CAIM), Oradea, 2003;

[3]. Grosan, C. How to compare the multiobjective evolutionary algorithms performances? Zilele Academice Clujene 2003;

[4]. Ko, I. Using AI techniques and learning to solve multi-level knapsack problems. PhD thesis, University of Colorado at Boulder, Boulder, CO, 1993;

[5]. Loots, W., Smith, T. H. C. A parallel algorithm for the zero-one knapsack problem. International Journal Parallel Program 21 (5), 313-348, 1992;

[6]. Martello, S. and Toth, P. Knapsack problems: Algorithms and computerimplementation, J. Willey and Sons, Chichester, 1990;

[7]. Penn, M., Hasson, D., Avriel, M. Solving the 0/1 proportional Knapsack problem by sampling, J. Optim. Theory Appl. 80, 261-272, 1994;

[8]. Sahni, S. Approximate algorithms for the 0/1 knapsack problem, Journal of ACM 22, 115-124, 1975;

[9]. Vasquez, M., Hao, J. K. A hybrid approach for the 0/1 multidimensional knapsack problem. Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI 01) 1 , 328-333, 2001;

[10]. Zitzler, E., Thiele, L. Multi objective Evolutionary Algorithms: A comparative case study and the Strength Pareto Approach, IEEE Transaction on Evolutionary Computation, 3(4), November 1999, pp 257-271;