Available online at www.scholarsresearchlibrary.com

Scholars Research Library

Archives of Applied Science Research, 2014, 6 (5):172-177
(http://scholarsresearchlibrary.com/archive.html)

ISSN 0975-508X
CODEN (USA) AASRC9

Solving Knapsack problem by simulating eurygaster life

Fariborz Ahmadi, Hamid Salehi* and Khosro Karimi

Young research and elites club, Ghorveh branch, Islamic azad university, Ghorveh, Iran

---

ABSTRACT

In this research, 0-1 knapsack problem are solved by simulating eurygaster life in wheat farms. Knapsack problem is a NP-Hard problem that has been focused to solve by researchers. On the other hand, eurygaster algorithm is a method invented to solve non-deterministic polynomial by researchers. This algorithm divides problems to several partitions and distributed eurygasters on each partition separately. In this research the proposed approach to solve knapsack problem is discussed. The assessment results indicate solving 0-1 knapsack problem using this method is faster that genetic one.

Keywords: Evolutionary computation, genetic algorithm, particle swarm optimization, NP-Hard problems, knapsack problem.

INTRODUCTION

Knapsack problem is a classical problem of optimization and combination [1, 2]. This problem is interested in both theory and practicality aspects. In the point of theory view, it is NP-Hard problem and also many problems can be solved using this problem. In the point of practical view, many economical programs can be solved by choosing or not choosing of their elements [3]. Since this algorithm is NP-Hard problem, some method like backtracking and dynamic programming can't be useful [8]. In knapsack problem exists one knapsack with capacity W and also an integer array of items with profit P_i and weight W_i. Some of these items must be selected so that overall profit to be maximized and total weight of selected items doesn't exceed from knapsack capacity. Knapsack problem can be defined mathematically as follows.

$$\sum_{i=1}^{n} w_i * x_i < W$$

And also the following formula be maximum.

$$\sum_{i=1}^{n} p_i * x_i$$

In these formulas x_i determines whether $item_i$ was selected or not selected. X_i can have 2 value, namely 0 or 1[4,6]. The difference between 0-1 knapsacks with fragmentation one is that on item is either taken fully or none of it is taken[8].

In the other hand, eurygaster algorithm is an algorithm developed to solve NP-Hard problems like knapsack problem. In this algorithm, each eurygaster shows one solution and the best solution is the answer of problem. In this algorithm a set of eurygasters attack to wheat farms and ruin it. After ruining one farm, they migrate to adjacent farm to disturb it. This routing is continued until either all of the farms are ruined or the best farm is reached.

Eurygaster behaviors

Eurygaster integricep is an insect pest that predominantly attacks grains, feeding on the leaves, stems and grains, reducing yield and injecting a toxin into the grains which adds a foul smell to the resulting flour, and substantially reduces the baking quality of the dough [1].

In winters eurygasters live under the plants and bushes in hillside, in several numbers and make a group. At the end of winter and at the beginning of spring when it gets warmer, these insects end their winter sleeps and get ready to move and fly to grain fields by moving over the high mountains and leaving the nests in groups. The first group by the use of its instinct finds the best and the nearest grain fields and stays there. Getting there, this group of insect sends signals to the air to show the other groups their being there. Based on the number of eurygasters in a place, the strength of signals will be different. If the number of eurygasters in a grain field is not great, the rate of diffused signals will be little and if the number of eurygasters in a grain fields is greater, the rate of diffused signals will be increased. They diffuse these signals to show the others that reside there. So that the other groups of eurygasters understand that they should not close to the grain field which contains the first group. Of course the other groups based on diffused signals by the first group and the strength of these signals they decide if they can land and stay there or not. If the power of diffused signals is low, it means that some of the other groups of eurygasters can land and stay by the other groups which are resident there and began to eat. While the strength of the signals in the sky is high, it means that the other groups cannot land on the field(s) containing eurygasters, and they must fly to other fields in which there are no eurygasters, to live and eat[1].

According to the passage mentioned above, the next group of eurygasters while flying from their nests to other fields to find the best grain fields searches the best and closest ones to land and eat based on the broadcasted signals by landed group(s). This process will continue until they will find a suitable and useful grain field to eat[1].

We conclude that all the grain fields in a wide area will be attacked by eurygasters, because they do not gather in a one place.so, when there is not enough food in a grain field in which the eurygasters have stayed for a time, they will fly to a new field with no eurygasters according to the process mentioned above.

4. Eurygaster algorithms

In this section, eurygaster algorithm is described. Solving non-linear functions are so necessary in real life today and recently researchers interested in inventing methods to solve them. Thus, our approach contributes to solve NP-class problems. The great advantage of this algorithm is that it's so easy to implement and is also inexpensive in term of memory and speed. The second advantage of this algorithm is its convergence speed compared to other methods like GA and PSO.

The related semi-code of the proposed algorithm is as algorithm1. This algorithm is formed by combining of 3 sub-algorithms. In each phase, we described how to apply this algorithm to knapsack problem in a way it can be solved.

4-1 Initialization

In this phase the problem of knapsack is divided into some partition and each partition is investigated separately. Also the structure of eurygaster is constituted. Suppose we have one problem according to table1 with capacity of knapsack 20.

Table1. A sample of knapsack problem

Items	Profit	weight
1	10	8
2	12	9
3	17	15
4	19	12
5	10	10

And also suppose we have a vector with values 0 and 1. These values show whether corresponding item is selected or not selected. According to these descriptions, the structure of one eurygaster can be showed as figure1.

0	1	1	0	0

Figure1. sample of eurygaster

In this figure items 2 and 3 have been selected. Total weight of this eurygaster is 24 with profit 29. This eurygaster is not a good choice because an overall weight of their items is greater than knapsack capacity of 20. The best choice is eurygaster that choose items 1 and 4. Also, in this problem the area of problem is divided into 3 partitions as follows.

1. The set of items sorted ascending
2. The set of items sorted descending
3. The set of items with arbitrary choice

In this research, first the domain of partition1 is investigated and if the answer is not obtained eurygastes move to another partition, namely partition2. And if the answer of problem be in first partition, the algorithm is terminated and solution that is eurygaster is reported. The main point in this algorithm is representing of eurygasters that we described it in figure1.

```
I← the number of clusters

While I <> 0 do

  1.  Initialization:  produce euragasters or particles according to characteristic of one partition
  2.  Distribution: distribute eurygasters on the regions of the partition
  3.  Evaluation: evaluate suitability of each eurygaster or particle depend on the problem
     3.1  If the suitable result of the partition is not obtained
          3.1.1.  Change the position of Eurygasters in the partition
          3.1.2.  goto 3
     3.2  If the result of the problem is not obtained
          3.2.1.  I--
          3.2.2. goto 1
Else
          3.2.3.  Stop algorithm or break
End while
```

Algorithm1. Eurygaster Algorithm

  4. Report the solution of the problem

4-2 Distribution

In this phase, eurygasters are distributed over partitions. First, they distributed in first partition. If optimal result is obtained, the algorithm is terminated and if the optimal solution is not gained new eurygaster is produced to search remaining section of partition. Finally, after searching almost all of the regions of partition, if the solution is not find eurygasters switch to another partition that is near to solution of the problem.

4-3 Siutability

In order to find the solution of the problem, eurygasters should be evaluated so that the best answer is obtained. For this reason, a fuction is used to assess eurygasters suitability. Let p_i be a profile of $item_i$, and w_i be its weight. And also x_i shows whether corresponding item has been selected or not selected. The suitability function can be calculated by this formula.

$$\sum_{i=1}^{n} p_i * x_i$$

It is bearing in mind that the total weights of items should not exceed from knapsack capacity.

5. Evaluation results

To evaluate our approach, we produced some files with different items and capacity. We test our approach in two ways. In one way, we produced different items with fix capacity. The number of items is increased and decreased to evaluate the efficiency of proposed approach. In second way, the number of items is fixed and knapsack is variant to consider suitability of researchers approach. Our test was runned on desk-top computer with following specifications.

CPU: Xeon L3014
RAM: 8 GB DDR3
OS: windows 8

In this paper, three samples are used with 100, 300, 400 items, respectively. The weight and profit of each item have been produced randomly. In first test, that knapsack capacity is 700. To evaluate our works, this algorithm has been compared with genetic algorithm. Also, to assess both algorithms, each sample was executed 20 times. Table2 shows the best solution of each algorithm with capacity 700.

Table2. Evaluation result of proposed algorithm and genetic algorithm in 3 produced sample

Samples	Best weight of proposed approach	Best weight of genet-ic algorithm	Best profit of proposed approach	Best profit of genet-ic algorithm
100 items	690	645	4983	4398
200 items	700	700	11879	9562
400 items	700	683	5734	4624

As you can see in table2, by executing 20 times of three items the best solution by each algorithm is obtained. In this table the best weight is obtain when the maximum profit is calculated. From this table, it is concluded that our approach has better solution than genetic one. Also figure2 shows average running time of both algorithm on three items in term of second.

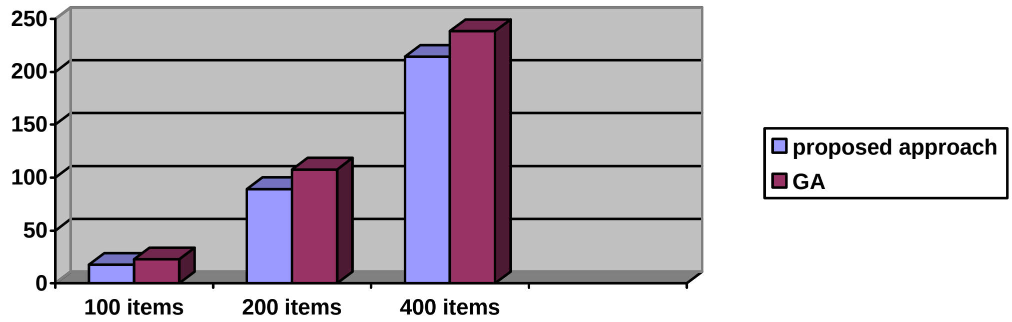

Figure2. Average running time of both algorithms

In second case of assessment, we change the knapsack capacity to 200, 500 and 1000, respectively. All of the three items are tested on these capacities and evaluation results are shown in table3, table4 and table5, respectively.

Table 3. Evaluation result of proposed algorithm and genetic algorithm with capacity 200

Samples	Best weight of proposed approach	Best weight of genetic algorithm	Best profit of proposed approach	Best profit of genetic algorithm
100 items	200	200	725	725
200 items	200	200	740	747
400 items	200	200	897	783

Table 4. Evaluation result of proposed algorithm and genetic algorithm with capacity 500

Samples	Best weight of proposed approach	Best weight of genetic algorithm	Best profit of proposed approach	Best profit of genetic algorithm
100 items	486	453	2920	2854
200 items	500	482	3440	3120
400 items	477	461	4387	3945

Table5. Evaluation result of proposed algorithm and genetic algorithm with capacity 1000

Samples	Best weight of proposed approach	Best weight of genetic algorithm	Best profit of proposed approach	Best profit of genetic algorithm
100 items	948	982	6930	6584
200 items	1000	961	8120	7451
400 items	980	980	7680	7680

The evaluation results show our approach is faster and more accurate than genetic one. This approach unlike genetic algorithm lacks the local optimum so the probability of getting the more accurate solution in this method is much more than the genetic algorithm. Moreover, in this method every space of the problem is searched for once while in the genetic algorithm every part of the problem space can be searched several times in different generations, so the rate of convergence in this algorithm is much more than the genetic algorithm. Figure 2 shows the convergence speed of proposed approach is faster than genetic ones. Also by table1 to table5, it is concluded that researchers work has accurate solution in comparison to genetic algorithm.

CONCLUSIONS

In this article, one approach based on the behaviors of eurygasters has been presented to solve knapsack problem. This approach unlike genetic algorithm lacks the local optimum so the probability of getting the more accurate solution in this method is much more than the genetic algorithm. Moreover, in this method every space of the problem is searched for once while in the genetic algorithm every part of the problem space can be searched several times in different generations, so the rate of convergence in this algorithm is much more than the genetic algorithm. Also, this algorithm is easy to implement by computer. If takes a few lines to programming and doesn't need a huge memory or CPU speed. Evaluation results on three produced files show that convergence speed by proposed approach is faster and also more accurate. Knapsack problem can be solved by ways of several methods like dynamic programming, greedy algorithm, PSO and genetic algorithm. According to researchers result, one of the best algorithm by which knapsack can be solved is genetic algorithm. But our proposed method is more suitable than genetic algorithm not only in convergence speed but also in accuracy of solution. It can be proved that our proposed approach is optimal in used memory in comparison to other heuristics algorithm like genetic and PSO.

REFERENCES

[1]Han K H, Kim J H. Genetic quantum algorithm and its application to combinatorial optimization problem[C]. Proceedings of the 2000 Congress on Evolutionary Computation, Piscataway, **2000**, 2: 1354 -1360.

[2] Han K H, Park K H, Lee C H, et al. Parallel quantum-inspired genetic algorithm for combinatorial optimization problem[C]// Proceedings of the 2001 IEEE Congress on Evolutionary Computation, **2001**: 1422-1429.

[3] S. Martello, D. Pisinger, P. Toth, *European Journal of Operational Research* 123 (**2000**) 325–332.

[4] D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. (Addison-Wesley, New York, **1989**).

[5] J.H.Holland, Adaptation in Natural and Artificial Systems (University of Michigan Press, 1975; MIT, London, **1992**).

[6] A. S. Anagun, and T. Sarac, ''Optimization performance of genetic algorithm for 0-1 knapsack problems using Taguchi method,'' Lect. Notes *Comput. Sc.,* vol. 3982, pp. 678-687, **2006**.

[7] G., Taguchi, S. Chowdhury, and Y. Wu, "Taguchi's Quality Engineering Handbook," Wiley-Interscience, New York, NY, **2004**.

[8] A. K. Bhatia, and S. K. Basu, *Soft Comput*., vol. 8, no. 1, pp. 1-9, **2003**.