Tikrit Journal of Pure Science 23 (2) 2018 *ISSN: 1813 – 1662 (Print)*
E-ISSN: 2415 – 1726 (On Line)

TJPS

TIKRIT JOURNAL OF PURE SCIENCE

Journal Homepage: http://main.tu-jo.com/ojs/index.php/TJPS/index

Projects Selection In Knapsack Problem By Using Artificial Bee Colony Algorithm

Armaneesa Naaman Hasoon

Department of Computer Science, College of Computer Science and Mathematics, Tikrit University, Tikrit, Iraq

Abstract

One of the combinatorial optimization problems is Knapsack problem, which aims to maximize the benefit of objects whose weight not exceeding the capacity of knapsack. This paper introduces artificial bee colony algorithm to select a subset of project and represented by knapsack problem to put the best investment plan which achieve the highest profits within a determined costs, this plan is one of the applications of the financial field. The result from the proposed algorithm implemented by matlab (8.3) show the ability to find best solution with precisely and rapidity compared to genetic algorithm.

ARTICLE INFO.

Article history:

-Received: 28 / 9 / 2017
-Accepted: 12 / 12 / 2017
-Available online: / / 2018

Keywords: knapsack problem (KP), Artificial bee colony(ABC), Genetic algorithm(GA).

Corresponding Author:

Name:
Armaneesa Naaman Hasoon
E-mail: nakm1982@gmail.com

Tel:

Affiliation:

1- Introduction

The most popular subset selection problem is knapsack problem (KP). The goal is to set a subset of n items from N original items to a knapsack, such that the summation of profit of the selected items is maximized which sum of weight not exceeding the capacity of knapsack [1].

Knapsack problem has many practical applications are not limited to packing problems: suppose that n projects are available to an investor and that the profit obtained from the j^{th} project is pj, j=1,2,…,n, whose costs wj to invest in project j, by solving (0-1) knapsack problem an optimal investment plan may be found[2].

Many methods such as: branch and bound, cutting planes, and dynamic programming etc., and nature inspired algorithm have been proposed to solve the knapsack problem [3,4].

Lately, evolutionary algorithms such as genetic algorithm (GA)[5,6], particle swarm optimization algorithm (PSO)[7], ant colony optimization algorithm (ACO)[8] and artificial bee colony algorithm (ABC)[4] efficiently have been developed for solving knapsack problem and reaches to the best solution.

In this paper uses ABC algorithm which proved efficiently in select subset of projects in knapsack problem to build best investment plan compared by GA. The rest of paper is organized as follows: section 2 gives related work, section 3 presents description to solve KP by use ABC algorithm, section 4 views the description of the proposed algorithm, experimental result and conclusion are introduced in section 5 and 6.

2- Related works

Many approaches are proposed to solve knapsack problem to obtain the best solution. In [6] introduced the basic form of genetic algorithm to solve knapsack problem. In[4] presents the using of a novel artificial bee colony algorithm to solve knapsack problem, a hybrid probabilistic mutation schemes in this approach are used to guide the search quickly. In [9] Karboga proposed ABC algorithm which its works similar to a bee colony. In [3] introduced a novel binary biogeography-based optimization algorithm for the knapsack problem, the result from [3] show that the new method is effective and efficient. In [5] implement a fast and efficient genetic algorithm to solve 0-1 knapsack problem feasibility and effectively. In [10] presents a discrete artificial bee colony for multiple knapsack problem, the result from that the presented method has enhanced convergence speed and quality than other evolutionary algorithms. In [11] applied bee algorithm to solve knapsack problem and attempt to find the best solution in knapsack problem by the number of random flight.

3- Description to solve knapsack problem by using ABC algorithm

Karboga in 2005 [9] for real parameter optimization proposed the ABC algorithm, which simulates the forging behavior of a bee colony. The colony of artificial bees consists of three groups: employed, onlookers, and scout bees. Randomly search for food source positions (solutions) by employed bees. Then, they share information about food source such as nectar amounts (quality of solutions) with the onlookers bees which waiting at the dance area at the hive[12-14]. For colony communication the dance is essential and contains three pieces of information regarding of a flower patch: the direction in which it will bees crossing, its distance from the hive and its quality (i.e., fitness). Provided that the amount of nectar of the new selected source is higher than the previous one, the new source position is memories by the bee, and forgets the old one. Otherwise, she keeps in her memory the previous position. The dances of employed bees are watch by onlooker and decide to choose the scout bees. Then employed bees start to search in the neighborhood area of the sources of food found by scout bees [10].

To solve knapsack problem involves packing the knapsack with projects which have a weight capacity C, and the number of projects 1,2,…,n, each project has weight corresponds to weight(wi) and profit (pi), the goal is to find a set of projects whose weight does not exceed the capacity of knapsack and to maximize the overall profit.

Mathematically, the zero-one knapsack problem can be represented by a vector of binary values X1,X2 …… Xn, Where Xi = 0 or 1 ($1 \le i \le n$). given ($C > 0$), $Wi > 0, Pi > 0$, $1 \le i \le n$, the zero-one knapsack problem can be represented by a vector of binary values X1,X2 …… Xn, Where Xi = 0 or 1 ($1 \le i \le n$). satisfy the constraint in equation 1 is the goal from finding a vector [5].

$$\sum_{i=1}^{n} Wi\, Xi \le C \qquad \dots\dots(1)$$

And maximize the total profit as:

$$\sum_{i=1}^{n} Pi\, Xi \qquad \dots\dots(2)$$

To select project by using ABC algorithm a possible solution for the problem can be represented by a food source, and the amount of neater for this food source meet the sum of profit for the selected projects in knapsack. Each bee in ABC algorithm is represented as a string of binary value each bit corresponds to absence or presence of that project in the solution of the bee [4].

One of approach to construct the initial solution by calculate heuristic information by divide profit to weight as in eq. 3 for all item before entered in steps of ABC algorithm.

$$Ri = Pi/Wi \qquad \dots\dots(3)$$

And compute the probability of presence each item in initial solution from eq. 4.

$$Ppi = \left(\frac{C}{T}\right).\left(\frac{Ri}{Mean(R)}\right) \qquad \dots\dots(4)$$

Where the sum of weight for all original items corresponds to T and C represents capacity of the knapsack and Mean(R) is the mean of the relation computed from eq. 4.

After finding a food source by each bee in each iteration, it must be modified where the sum of weight for overall items not exceeds the capacity of knapsack, then evaluate the gathered solution generated by employed bee by compute the fitness (nectar amount) of the k^{th} bee corresponds to the sum of the total profit of the selected items in knapsack by that bee, then apply the selection strategy to choose the best bee as scouts the chance of selecting a bee by onlooker computed from eq.5 [10].

$$\text{Chk} = \text{Fk}/\sum_{i=1}^{M} Fi \qquad \dots\dots(5)$$

F_k: is the profit of knapsack represented by the k^{th} bee.

$\sum_{i=1}^{N} Fi$: is the total amount of achieved nectar by all the explorer bees.

M: number of employed bee.

Then continue in running ABC algorithm which includes the following algorithm[15][4]:

1- Initialization: parameters of ABC algorithm are initialized K, N, CC_i =0 (i=1,…,K).

2- Initial solution: generate initial solution (food sources) { $FS_i(0)$ | i=1,…,K } by calculate the value of profit to weight by apply eq.3, Then compute probability of presence of each item in initial solution from eq.4.

3- Loop:

For r=1:N do

{

(1) For i=1:K do (select food sources by employed bees and run local search respectively).

{

- Assassinate each employed bee with a food sources $FS_i(r)$ and calculate its amount of nectar.
- In the neighborhood of $FS_i(r)$ find new $FS_i'(r)$ and compute its amount of nectar;
- Put the better one in { $FS_i'(r)$, $FS_i(r)$ } as a new position of the employed bees;

}

(2) For j=1:K do (perform further local search by onlooker bees which help employed bees)

{

- A food source $FS_i(r)$selected from { $FS_i(r)$ } for every onlooker bee;
- A new food source $FS_i'(r)$ in the neighborhood of $FS_i(r)$ is find then calculate its nectar amount;

- Consider the better one in { $FS_i'(r)$, $FS_i(r)$ } as a new position of the corresponding bees;
}

(3) A new food sources exploiting (scout bees randomly perform global search).
For i=1:K do (food sources)
{
If $FS_i(r)$= { $FS_i(r-1)$ then CC_i= CC_i+1;
If CC_i=Limit (Limit= K*N) then
{
- The $FS_i(r)$ is abandon and the associated employed bee becom scout;
- Generate randomly a new $FS_i(r)$ and the scout becomes an employed bee again ;
- CC_i=0;
}
}
- Register the best food source FS_{best} found so far; r=r+1;
}

4- output: if the predefined end condition are met then return FS_{best}.

4- Description of the proposed algorithm

In this paper, ABC algorithm used to select subset of projects in KP to build good investment plan, these subset represent the best solution which have maximized gross-profit and the sum of total cost not exceeds the capacity of knapsack. There are set of projects available to investor each project has value of sale-revenue and value of cost-making which consist from four types: cost of direct materials, the cost of direct wages, variable industrial costs and non-variable industrial costs, the goal from proposed algorithm select best subset of projects from this set in less executing time.

The proposed algorithm contains the following steps

1- Initialization: parameters of ABC algorithm are initialized K=number of population=no. employed bee=no. onlooker bee, N=maximum iteration, CC_i =Abandonment counter=0 (i=1,…,K).

2- Enter the array of sales-revenue which size [1…no. of projects], and matrix of cost-making whose dimension [no. of project, 4 = types of cost-making].

3- Calculate the total cost from summation of cost-making for each project.

4- Compute Gross-profit for each project by subtract total cost from sales-revenue.

5- Initial solution: generate initial solution (food sources) {$FS_i(0)$ | i=1,…,K } by calculate the value of profit to weight by apply eq.3, Then compute probability of presence of each item in initial solution from eq.4, and generate initial solution based on the computed value of probability.

6- Loop:
For r=1:N do
{

(4) For i=1:K do (select food sources by employed bees and run local search respectively).
{
- Assassinate each employed bee with a food sources $FS_i(r)$ and calculate its amount of nectar (corresponds to the sum of the gross-profit of the selected project in knapsack by that bee).
- In the neighborhood of $FS_i(r)$ find new $FS_i'(r)$ and compute its amount of nectar;
- Put the better one in { $FS_i'(r)$, $FS_i(r)$ } as a new position of the employed bees;
}

(5) For j=1:K do (perform further local search by onlooker bees which take help from employed bees)
{
- A food source $FS_i(r)$selected from { $FS_i(r)$ } for every onlooker bee by apply roulette wheel selection method which pass the probability of choosing calculated from eq. 5;
- A new food source $FS_i'(r)$ in the neighborhood of $FS_i(r)$ is find then calculate its nectar amount;
- Consider the better one in { $FS_i'(r)$, $FS_i(r)$ } as a new position of the corresponding bees;
}

(6) A new food sources exploiting (scout bees randomly perform global search).
For i=1:K do (food sources)
{
If $FS_i(r)$= { $FS_i(r-1)$ then CC_i= CC_i+1;
If CC_i=Limit (Limit= K*N) then
{
- The $FS_i(r)$ is abandon and the associated employed bee become scout;
- Generate randomly a new $FS_i(r)$ and the scout becomes an employed bee again ;
- CC_i=0;
}
}
- Register the best food source FS_{best} found so far whose gross-profit maximized and the sum of the total cost not exceeds the capacity of knapsack (suppose C calculated from multiply 0.25 by sum of the total cost); r=r+1;
}

7- Output: if the predefined end condition (corresponding to access maximum of iteration) are met then return FS_{best}.

5- Experimental results

The proposed algorithm was implemented by matlab 8.3 that contribute in building efficient investment plan with zero-one knapsack problem. Initialize the parameters of ABC algorithm which illustrates in table (1).

Table (1) parameters of ABC algorithm

Parameter	Value
Iteration maximum	400
Number of employed bees	100
Number of onlooker bees	100
Number of scouts bees	10

There are n projects each one has profit and cost for explain the steps of proposed algorithm in simplicity get the example with eight projects, in table (2) lists the sales-revenue of the projects.

Table (2) the sales-revenue for each project

Project number	the sales-revenue of the projects
project1	92045
project 2	90789
project 3	58047
project 4	106336
project 5	49013
project 6	50292
project 7	51336
project 8	105390

Table (3) illustrate cost-making which consist from four types: cost of direct materials, the cost of direct wages, variable industrial costs and non-variable industrial costs for each project.

Table (3) cost-making for the projects

No. project Cost –making type	Project 1	Project 2	Project 3	Project 4	Project 5	Project 6	Project 7	Project 8
cost of direct materials	5475	5353	6096	7496	7866	7503	5251	6832
cost of direct wages	5606	6765	7775	6211	7815	7220	5198	5431
variable industrial costs	7600	5967	7289	7642	6516	6412	7055	5885
non-variable industrial costs	6965	7720	7409	7155	7082	6800	6933	6134

Then calculate the total cost from summation of cost-making for each project, the result explained in table(4).

Table (4) summation of cost-making for each project

No. project	Project1	Project2	Project3	Project4	Project5	Project6	Project7	Project8
Total cost	25646	25805	28569	28504	29279	27935	24437	24282

Then compute gross profit by subtract total cost from sale-revenue showed in table(5).

Table (5) total profit for the projects

No. project	Project1	Project2	Project3	Project4	Project5	Project6	Project7	Project8
Total profit-loss	66399	64984	29478	77831	19734	22357	29899	81108

In this paper the initial solutions are not generated randomly, obtained from eq.3 by divide gross-profit at total cost for each project to find the relation, the result in table (6).

Table (6) the relation between gross profit to the total cost

No. project	Project1	Project2	Project3	Project4	Project5	Project6	Project7	Project8
Relation (R)	2.5891	2.5183	1.0318	2.7305	0.6740	0.8003	1.1007	3.3403

then calculate the probability of each project from eq.4 illustrated in table (7).

Table (7) the probability of presence each project in initial solution

No. project	Project1	Project2	Project3	Project4	Project5	Project6	Project7	Project8
Probability	0.3502	0.3407	0.1396	0.3694	0.0912	0.1083	0.1489	0.4518

Generate the initial solution depending on the probabilities computed in previous step, then calculate gross -profit-loss and total cost for each solutions explained in table (8).

Table (8) the initial solution

initial solution	cost	profit
[0,0,1,1,0,1,1,0]	109445	156565
[1,0,1,0,1,0,1,1]	132213	223618
[1,1,0,1,1,1,0,0]	137169	251305
[1,1,0,0,1,1,1,1]	157384	281481
[1,0,0,0,0,1,0,0]	53581	88756
[0,0,0,0,0,0,1,0]	24437	26899
[1,1,0,1,1,0,1,0]	133671	255847
[0,1,0,0,0,0,0,0]	25805	64984
[0,0,0,0,0,0,1,1]	48719	108007
[0,1,1,0,0,0,0,0]	54374	94462

Take the solutions which corresponding to the total cost not exceeds the capacity of knapsack obtained from multiply 0.25 by the sum of the total cost, and save as the best solution, then compute the value of fitness from eq.5. for the generated solution from divide the gross profit for each string at the sum of gross profit then passing the value of selection probability computed from eq.5 to the roulette wheel selection, then select new food position and evaluate this position by compute the value of nectar amount and compare with the previous food positions, then change with the worst and continue with the rest steps of ABC algorithm until reach to the stopping criteria of algorithm. In table (9) illustrates the final solution by select subset of best project whose maximum gross - profit and sum of the total cost not exceed capacity of knapsack.

Table (9) the best solution from apply ABC algorithm

Best solution	Cost	Profit	Executing time
[1,0,0,0,0,1,0,0]	53581	136025	0.159606

To compare ABC algorithm with other method such as genetic algorithm which its parameters explained in table (10).

Table (10) parameter of GA

Parameter	Value
Itermax	400
Number of chromosome	10
Type of selection	Roulette wheel selection
Type of crossover	One point
Type of mutation	Flip bit
Probability of crossover	0.6
Probability of mutation	0.1

The result from comparison two methods with examples with 8, 10, 25, 50, and 100 projects for finding the best solution showed in table (11), which explains the efficiency of ABC algorithm in access to the good quality solution than applying GA.

Table (11) comparison between applying ABC algorithm and genetic algorithm

No. of project	Capacity of knapsack	Sum of gross-profit		Sum of the total cost		Executing time	
		ABC	GA	ABC	GA	ABC	GA
8	53614.75	136025	100842	53581	53561	0.159606 sec	3.046434 sec
10	64770.75	36169	30747	58702	58191	0.435974 sec	40.988467 sec
25	1647715.5	260177	256249	163993	163453	0.598020 sec	69.291353 sec
50	325416.75	685087	577105	324473	325197	1.878421 sec	95.296004 sec
100	645463	1345983	988996	645450	645462	16.623129 sec	186.56857 sec

6- Conclusion

In this paper a proposed ABC algorithm used with knapsack problem to select subset of project with maximum gross-profit and the sum of total cost for them not exceeds the capacity of knapsack. The obtained result shows the highly speed ,accuracy, and good quality of solution from applying ABC algorithm compared with genetic algorithm to find best solution that contribute in put optimal investment plan.

References

1- M. shokouhifar, F. Farokhi, " Feature selection using supervised fuzzy C-Means algorithm with anty colony optimization", 3 rd international conference on machine vision (ICMV), 2010.

2- D. Pisinger, "Algorithms for knapsack problems", PhD. Thesis, university of Copenhagen, Denmark, 1995.

3- B. Zhao, C. Deng, Y. Yang, H. Peng, " Novel binary biography based optimization algorithm for knapsack problem", Springer, pp. 217-224, 2012.

4- S. Sabet, F. Farokhi, M. Shokouhifar, " A novel artificial bee colony algorithm for the knapsack problem", IEEE conf., 2012.

5- M. Gupta, " A fast and efficient genetic algorithm to solve 0-1 knapsack problem", International journal of digital application & contemporary research, vol. 1, Issue 6, 2013.

6- R. Spillman, " Solving large knapsack problems with a genetic algorithm", IEEE, 1995.

7- Y. Liang, L. Liu,D. Wang, R. Wu, " particle swarm optimization to solve knapsack problem", springer, ICICA 105, pp.437-443, 2010.

8- Z. Hu, R. Li, " Ant colony optimization algorithm for the 0-1 knapsack problem based on genetic algorithm", advanced materials research, vol. 230-232, pp.973-977, 2011.

9- D. Karboga, " An idea based on honey bee swarm for numerical optimization", (Technical Report- TR-06) , Turkey, 2005.

10-S. Sabet, F. Farokhi, M. Shokouhifar, " A discrete artificial bee colony for multiple knapsack problem", International journal reasoning based intelligent systems, vol.5, no.2, 2013.

11-S. Soleymani, A. Hatamloo, " To solve knapsack problem using bee algorithm", International academic journal of science and engineering, vol.2, no.11, pp.23-30, 2015.

12-B. Akay, D. Karaboga, "A modified Artificial Bee Colony algorithm for real-parameter optimization", Information Sciences 192 , 2012.

13-G. Tankasala, "Artificial bee colony optimization for economic load dispatch of a modern power system", International journal of scientific & engineering research, vol.3, Issue.1, 2012.

14-F. Abo-Mouti, M. El-Hawary, "overview of artificial bee colony (ABC) algorithm and its application", IEEE, 2012.

15-J. Ji, H. Wei, C. Liu, B. Yin, "Artificial bee colony algorithm merged with pheromone communication mechanism for the 0-1 multidimensional knapsack problem", hindwai publishing corporation, mathematical problems in engineering, vol. 2013.

انتقاء المشاريع في مسألة حقيبة نابساك باستخدام خوارزمية مستعمرة النحل الاصطناعية

ارمانيسه نعمان حسون علي

قسم علوم الحاسوب ، كلية علوم الحاسوب والرياضيات ، جامعة تكريت ، تكريت ، العراق

nakm1982@gmail.com

الملخص

تعد مسألة حقيبة نابساك واحدة من مسائل الأمثلية التي تهدف الى اختيار العناصر التي تمتلك الفائدة الاكبر ووزنها لا يعبر حد الحقيبة. نُقدم في هذا البحث خوارزمية النحلة لاختيار مجموعة جزئية من المشاريع وتمثيلها بمسألة الحقيبة لوضع الخطة الاستثمارية الأفضل التي تحقق أعلى الفوائد ضمن الكلف المحددة، وهذه الخطة هي إحدى تطبيقات المجال المالي. وقد بيَّنت النتائج المستحصلة من الخوارزمية المقترحة والممثلة باستعمال الماتلاب إمكانية ايجاد الحل الأفضل بدقة وسرعة عالية مقارنة بالخوارزمية الجينية.