

Adaptation of a Mosquito Swarm Algorithm to Solve a Binary Knapsack Problem

Ruqaya Zedan Sha'ban

israalkallak19@gmail.com

Computer Unit

College of Medicine

University of Mosul, Mosul, Iraq

Received on: 08/10/2013

Isra Natheer Alkallak

nnabeel2013@gmail.com

Dep. Basic Sciences

College of Nursing

University of Mosul, Mosul, Iraq

Accepted on: 09/04/2014

ABSTRACT

The paper tackled one of the artificial intelligence algorithms which is Mosquito swarm algorithm as a nature inspired algorithms, to solve 0-1 knapsack an optimization problem through adaptation steps of a Mosquito swarm algorithm and the inspiration coming from the behavior of Mosquitoes, in addition to propose steps (update) for Chemical sensors, Heat sensors and location of Mosquito, to find an optimal solution in small search space and small computational requirements. A program is written in MATLAB 7.10 .language to simulate this algorithm with 0-1 knapsack problem. The experimental results in this research showed effective approach, easy implementation of algorithm to find optimal solution.

Keywords: Mosquito Algorithm, Knapsack Problem.

تكييف خوارزمية سرب البعوض لحل مشكلة الحقيبة الثنائية

إسراء نذير الكلاك

قسم العلوم الأساسية

كلية التمريض

جامعة الموصل، الموصل، العراق

رقية زيدان شعبان

وحد الحاسبة

كلية الطب

جامعة الموصل، الموصل، العراق

تاريخ قبول البحث: 2014/04/09

تاريخ استلام البحث: 2013/10/08

المخلص

تطرق هذا البحث إلى إحدى خوارزميات الذكاء الاصطناعي وهي خوارزمية سرب البعوض ذات الطبيعة الملهمة لحل مسألة من مسائل الأمثلية -مسألة الحقيبة الثنائية- من خلال تكييف خطوات الخوارزمية والإلهام الذي يأتي من سلوك البعوض فضلاً عن اقتراح (تطوير) معادلات الاستشعار الكيميائي، الاستشعار الحراري وموقع البعوضة، للحصول على حل مثالي لمسألة الحقيبة الثنائية بفضاء صغير وحسابات قليلة. تم إعداد برنامج حاسوبي بلغة ماتلاب 7.10 ليحاكي الخوارزمية مع المسألة. أثبتت نتائج البحث سهولة مرنة وكفاءة في انتقاء الحل الأمثل من بين العديد من الحلول في فضاء بحث المسألة.

الكلمات المفتاحية: خوارزمية البعوض، مسألة الحقيبة.

1. Introduction

Recently in computer science, there is an increased interest in computational approaches that are inspired by the principles of nature and that can solve difficult problems. Whereas, physical and biological sciences try to find microscopic laws that extrapolate to the macroscopic realm, computer science is principally synthetic and is concerned with the construction of algorithms that may be inspired by physics or biology.

Computing can be seen as a property of nature, therefore by mimicking natural phenomena; it is possible to construct new intelligent algorithms for solving computationally difficult problems called Nature –inspired algorithms [3,4]. They mimic different natural systems and processes using mathematical models and algorithms.

This research presents the classical 0–1 Knapsack problem to pick up items for a knapsack for maximum total value, so that the total resource required does not exceed the resource constraint R of the knapsack. The classical 0–1 Knapsack problem and its variants are used in many resource management applications such as cargo, loading, industrial production, menu planning, and resource allocation in multimedia servers [1].

In 2012, Yan *et al* used particle swarm optimization algorithm to solve the traveling salesman problems and experiment results are efficient than the genetic algorithm. In 2013, Feng *et al* proposed a new branch of biology inspired algorithm Mosquito host seeking algorithm and they are applied to solve traveling salesman problem. The proposed Mosquito host seeking algorithm is fundamentally different from the other popular nature algorithm in its motivation, principle, the optimization mechanism and so on.. they showed the actual times and iterations used to solve traveling salesman problems on a cluster, which can verify the efficiency and parallelism of Mosquito host seeking algorithm.

This research aims to adaptive a Mosquito swarm algorithm to solve a binary Knapsack problem coming from the behavior of Mosquitoes, to find an optimal solution in small search space and small computational requirements. This research is proposed (update) for Chemical sensors, Heat sensors and location of Mosquito. Locally, this research is a first study so that other researches are not available to be compared with it.

In this paper, besides this an introductory section; section 2 presents the problem definition. In section 3 described particle swarm optimization algorithm, methodology of Mosquito swarms algorithm. Section 4 described proposed steps to solve knapsack problem using Mosquito algorithm by experimented results, flowchart of the problem and problem instance. Finally, conclusions, recommendations in section 5 and section 6 respectively.

2. Problem Definition

The name knapsack problem invokes the image of the backpacker who is constrained by a fixed-size knapsack, and so must fill it only with the most useful and portable items [9]. The binary knapsack problem is one of the most important models in combinatorial optimization, having numerous real life applications as well as being an important sub- problem in several algorithms for more complex problems [2,7]. For the 0-1 knapsack problem, it involves with selecting items or projects to maximize total

profit without exceeding the existing capital budget or knapsack capacity. The problem has several applications in naval, financial management and steel industry [10]. Suppose a hitch-hiker has to fill up his knapsack by selecting from among various possible objects those which will give him maximum comfort. This knapsack problem can be mathematically formulated by numbering the objects from 1 to n and introducing a vector of binary variables x_j , ($j = 1, \dots, n$) having the following meaning: [6]

$$x_j = \begin{cases} 1 & \text{If object } j \text{ is selected} \\ 0 & \text{Otherwise} \end{cases}$$

Then, if p_j is a measure of the comfort given by object j , w_j its size and C the size of the knapsack, our problem will be to select, from among all binary vectors X satisfying the constraint

$$\sum_{j=1}^n w_j x_j \leq C$$

The one which maximizes the objective function

$$\sum_{j=1}^n p_j x_j$$

The problem is to find an optimal subset of items so as to maximize the total profits subject to the knapsack's weight capacity. The profits, weights and capacities are positive integers. Figure (1) illustrated knapsack problem graphically.

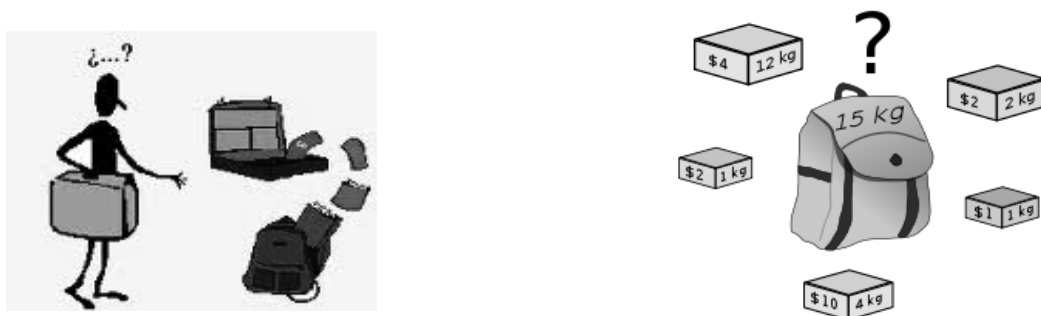


Figure (1) knapsack Problem Graphically

3. Particle Swarm Optimization Algorithm

Particle swarm optimization algorithm was an intelligent technology first presented in 1995 by Eberhart and Kennedy, and it was developed under the inspiration of behavior laws of bird flocks, fish schools and human communities [5].

A basic variant of the particle swarm optimization works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formula. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered, these will then come to guide the movements of the swarm. The process is repeated and by doing so, it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered. Particle swarm optimization achieves its optimum solution by starting from a group of random solution and then searching repeatedly. Particle swarm optimization is a meta-heuristic as it makes few or no assumptions about the problem [11]. Swarm algorithms refers to any algorithm that models the grouping of insects and

zooplankton swarms by social behavior. The swarm algorithms are inspired by the insect swarms and zooplankton swarms [8].

3.1 Mosquito Swarms Algorithm

Ruiz-Vanoye and Díaz-Parra in (2012) propose the Mosquito swarm algorithm. The Mosquito swarm algorithm is considered as a meta-heuristics algorithm, a bio-inspired algorithm, parallel or distributed algorithm based on the research on the social behavior of mosquito swarm. Mosquitoes (gnats) have sensors designed to track their prey:

- A) Chemical sensors, mosquitoes can sense carbon dioxide and lactic acid up to 36 meters away. Mammals and birds give off these gases as part of their normal breathing. Certain chemicals in sweat also seem to attract mosquitoes.
- B) Heat sensors, mosquitoes can detect heat, so they can find warm-blooded mammals and birds very easily once they get close enough. A mosquito swarm exists close to areas with standing water [8].

In this research, an algorithm called Mosquito algorithm has been used, which is a bio-inspired computing technique inspired by host-seeking behavior of mosquitoes swarms. In the real world, both male and female mosquitoes are nectar feeders, but the female of many species is also capable of drinking blood. Only female mosquitoes search the host (object) to attack. Female mosquitoes hunt their blood host by detecting carbon dioxide (CO₂) and 1-octen-3-ol (C₈H₁₆O) from a distance. Female mosquitoes have host-seeking behavior, but male ones do not. As shown in Figure (2) a swarm of mosquitoes is randomly searching the host (object) to attack. We can summarize three steps of mosquito's host-seeking behavior as follows [3]:

- 1. The mosquito looks for carbon dioxide or smelling substance.
- 2. It distinguishes the smell it loves, and then seeks towards a high-concentration location.
- 3. It makes a descent when it feels the radiated heat of the host.

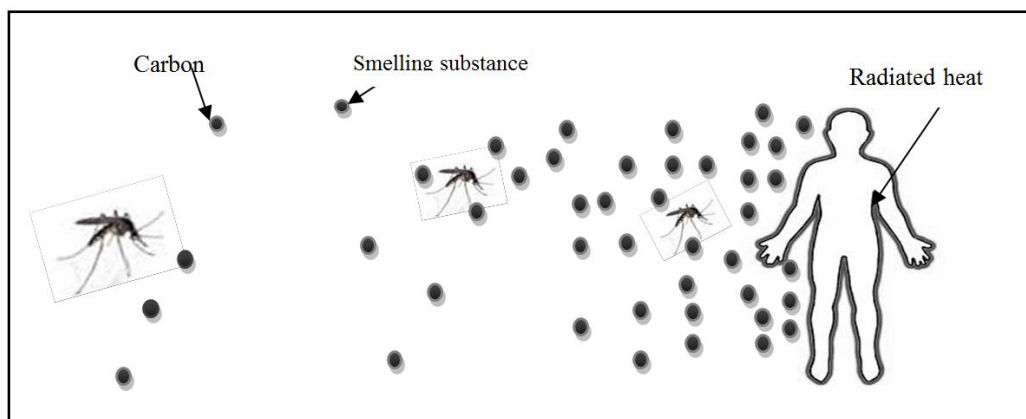


Figure (2) Steps of Mosquitoes Host-Seeking Behavior

3.2 Methodology of Mosquito Swarms Algorithm

Below a description of the structure of the Mosquito algorithm in general [8].

Input: number of mosquitoes (m).

- 1- Initialize a mosquito population with Chemical Sensors (CS) and Heat Sensors (HS).
- 2. Generating the initial locations (x) of the mosquitoes (m).
- 3. Initialize the temperature (t) and maximum temperature (T).

4. Repeat (total of mosquitoes) by parallel and/or distributed processing.
5. Repeat (maximum temperature).
6. Generate new solutions by adjusting the Heats (HS) and updating the locations (x).
7. Verify and assign the feasibility of the solution by the Chemical Sensor (CS).
8. Select the best solution (S).
9. While $t < T$
10. While (m total of mosquitoes)
11. Report the best solutions.

4. Experimental Results with Proposed Steps

Below the proposed steps in this research to solve Knapsack problem by using Mosquito algorithm. Artificial mosquitoes can move towards the host (goal) under the factors as follows:

- The personal host-seeking behavior.
- The aggregate host-seeking behavior.
- The attraction of the host.
- The motion that is related to social coordination's among the swarm of artificial mosquitoes (update of location).

In this research, we proposed the steps to initialize factors as CS, HS and location (x) as follows:

If we have n objects and p_j is a profit value, w_j its size and C the size of the knapsack as in table (1).

Table (1) Structure of General Problem

Items	Weight	Profit Value
1	w1	P1
2	w2	P2
3	w3	P3
:	:	:
:	:	:
:	:	:
n	wn	Pn

Begin

Step1: Initialization of

items as follows:

- Input the number of mosquito as m .
- Each mosquito represent solution in search space, all mosquitoes represent the population of the problem, the size of population as 10 mosquitoes, each solution

contain's subset of items such as: $\{1,2\}, \{3,5,7\}$ to find the objective function for all mosquitoes $\{z_1, z_2, \dots, z_m\}$

- Initialize the temperature as t and the maximum temperature as T .
Assume $t = 0$, $T = 10$

- Initialize the population with factors CS, HS and location as follows:

A- To initialize the Chemical Sensors (CS) for each solution (mosquito) by the formula

$$CS_i = C - \sum_{j=1}^n W_j \quad i = [1 \dots m], j = [1 \dots n]$$

B- To initialize the Heat Sensors (HS):

We assumed that HS depends on the value of CS

$$\text{IF } CS_i \begin{cases} > 0 & \text{then } HS_j = 1 \\ < 0 & \text{then } HS_j = 0 \\ = 0 & \text{then } HS_j = 10 \end{cases}$$

C-Generating the initial locations (x) of the mosquitoes (m) by applying the formula:

$$x_i = (\sum CS_j + HS_i / z_j) * c \quad , z_j \text{ is objective function of the current solution.}$$

, where c is learning factor.
In this research, learning factor $c = 2$.

Step2: Find best solution with objective function:

IF z is the objective function of Mosquitoes swarm $= \{z_1, z_2, \dots, z_m\}$
then best solution is $\max \{z_1, z_2, \dots, z_m\}$.

Let S^* , assign the best solution.

Let x^* , assign best location x .

Step3: Iteration by parallel do the following:

Generate a new solution by adding the item that is not found in the previous solution and, then adjusting heats (HS) and updating the locations (x) by using sub steps A, B and C in step1.

Step4: Verify and assign the feasibility of the solution by the chemical sensor (CS) by:

IF current solution is better than S^* then,

$S^* =$ current solution.

$X^* =$ current position.

Goto step 5

ELSE

Do not change S^*

Step5: IF $T = 10$ then,

report the best solution that contains subset of items that satisfies the maximum objective function and constraints with capacity = C.

Go to step 7

ELSE

Step6: Repeat the steps from step 3.

Step7: End.

4.1 Flowchart of the problem

Figure (3) below illustrates the flowchart of the Knapsack problem solution by Mosquito algorithm:

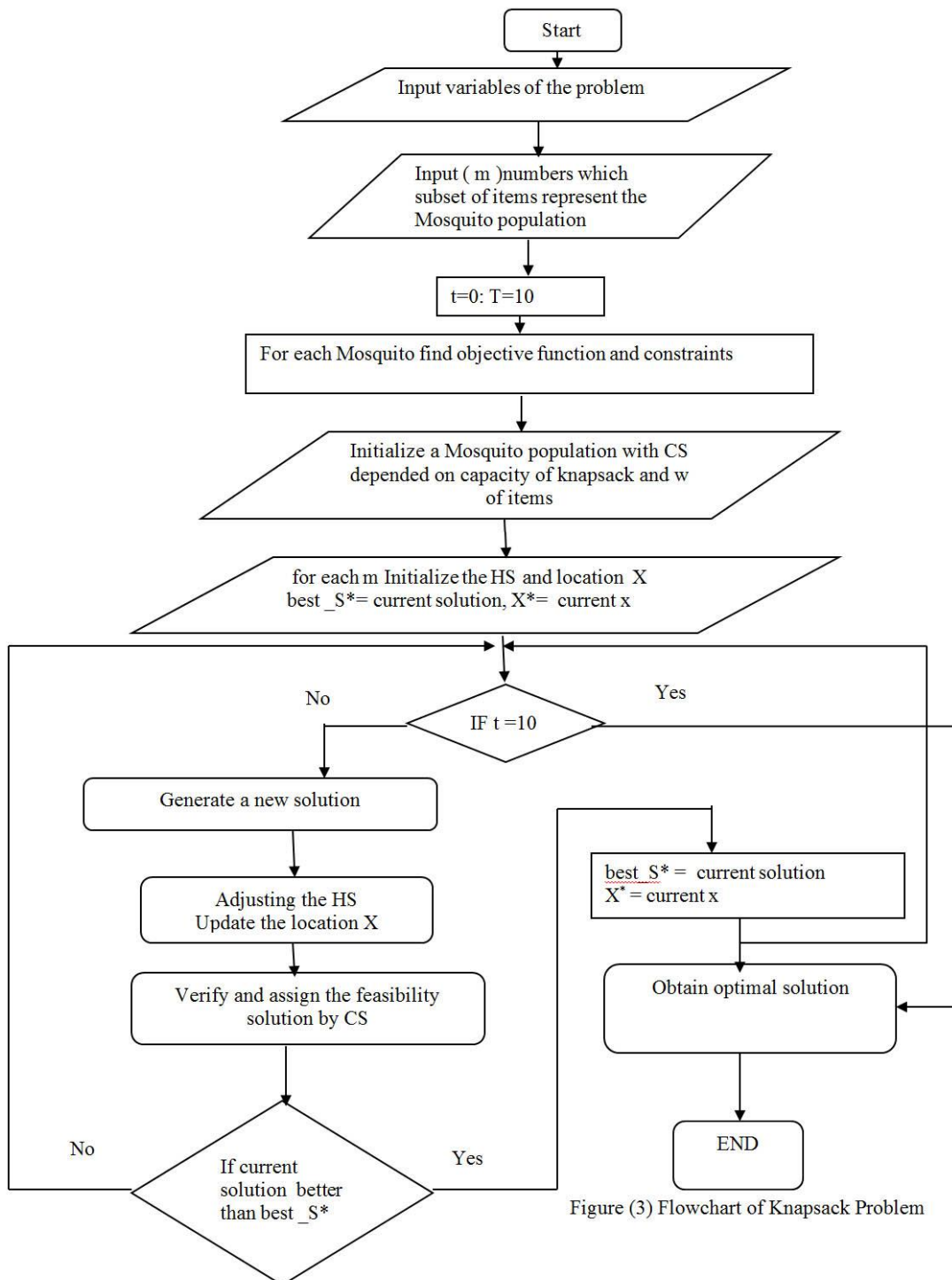


Figure (3) Flowchart of Knapsack Problem

4.2 Problem Instance

There are 8 items where each item j has a weight w_j , a benefit or profit value p_j and the knapsack capacity must carry items of total weight at most (102) as in table (2).

Table (2) Variables of the Knapsack Problem

Items	1	2	3	4	5	6	7	8
Weights	2	20	20	30	40	30	60	10
Profit values	15	100	90	60	40	15	10	1

Solution:

1- The initial population of our problem is as shown in table (3), representation of general problem in Figure (4).

Table (3) Initial Population of Problem

Mosquito	Population	Subset of Items
m1	[0 0 0 0 1 0 1 0]	{5,7}
m2	[0 0 1 0 0 0 0 0]	{3}
m3	[0 0 1 0 0 0 1 0]	{3,7}
m4	[1 1 0 0 0 0 0 0]	{1,2}
m5	[1 0 0 0 1 1 0 0]	{1,5,6}
m6	[0 1 0 0 1 1 0 1]	{2,5,6,8}
m7	[0 0 1 1 0 1 0 0]	{3,4,6}
m8	[0 0 0 1 1 0 0 1]	{4,5,8}
m9	[0 0 1 1 0 0 0 1]	{3,4,8}
m10	[1 0 1 0 0 1 0 1]	{1,3,6,8}

2- For each solution (mosquito) find the objective function, constraint and initialize locations CS, HS as in table (4):

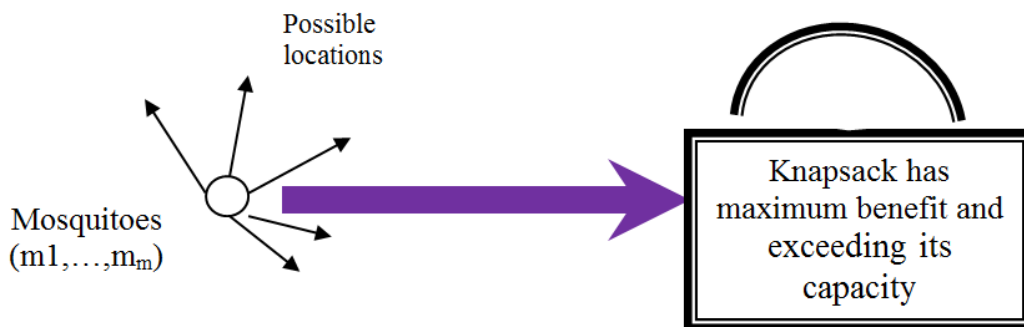


Figure (4) Representation of General Problem

Table (4) Initial Variables of Problem

Mosquito	Objective function (Z)	Satisfy the constraint	CS	HS	Locations (x)
m1	50	100	2	1	0.12
m2	90	20	82	1	1.84
m3	100	80	22	1	0.46
m4	115	22	80	1	1.40
m5	70	72	30	1	0.88
m6	156	100	2	1	0.03
m7	165	80	22	1	0.27
m8	101	80	22	1	0.45
m9	151	60	42	1	0.56
m10	121	62	40	1	0.67

feasible solution

Figure (5) describes how the initial populations ($m1, \dots, m10$) distributed in different locations and the best solution near to optimal that has the items $\{2,5,6,8\}$. See column 3 of table 4, ($m1$ and $m6$) has the same weight but we take $m6$ because it satisfies maximum z , so, it becomes best z^* solution up now.

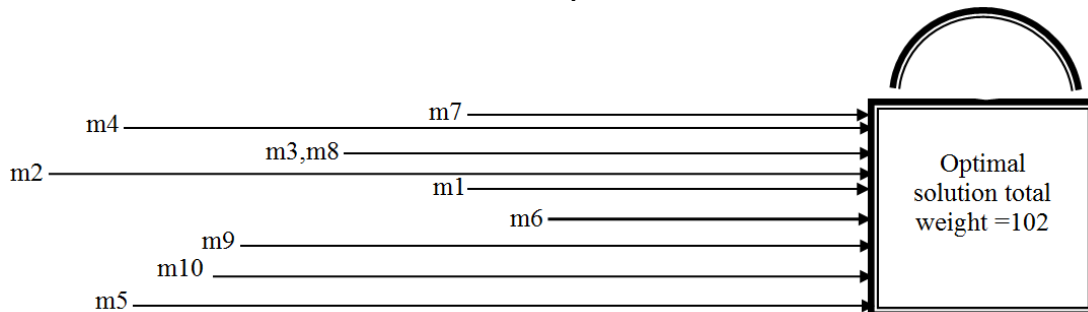


Figure (5) Representation of Ten Solutions and How Far From Optimal Solution

3- During the steps of mosquito algorithm to this problem, we found two types of solutions feasible and not feasible; not feasible solution that is far from optimal solution. after generating a new solution and updating HS, CS and locations in iteration 2. We obtain table (5), also $m6$ is the optimal solution.

Table (5) Result of Iteration 2

Mosquito	Objective function (Z)	Satisfy the constraint	CS	HS	Locations (x)
m1	65	102	0	10	0.3
m2	105	22	80	1	1.54
m3	115	82	20	1	0.36
m4	205	42	60	1	0.59
m5	170	92	10	1	0.12
m6	171	102	0	10	0.11
m7	180	82	20	1	0.23
m8	116	82	20	1	0.36
m9	166	62	40	1	0.49
m10	221	82	20	1	0.19

After iteration 3, we obtain the results described in table (6). ($m3, m7$ and $m8$) all of them satisfy the capacity of knapsack, but we consider $m7$ as an optimal solution because it satisfies the maximum benefit of items, see Figure (6). Finally, the optimal solution is $[1\ 1\ 1\ 1\ 0\ 1\ 0\ 0]$ which has items $\{1,2,3,4$ and $6\}$. Figure (7) provided a graphic illustration of relationship between the solutions and their locations in our example in three iterations.

Table (6) Result of Iteration 3

Mosquito	Objective function (Z)	Satisfy the constraint		CS	HS	Locations (x)
m1	165	122	not feasible	-20	0	-0.24
m2	205	42	feasible	60	1	0.59
m3	215	102	feasible	0	10	0.09
m4	265	72	feasible	30	1	0.23
m5	260	112	not feasible	-10	0	-0.07
m6	261	122	not feasible	-20	0	-0.15
m7	280	102	feasible	0	10	0.07
m8	216	102	feasible	0	10	0.09
m9	266	82	feasible	20	1	0.15
m10	281	112	not feasible	-10	0	-0.07

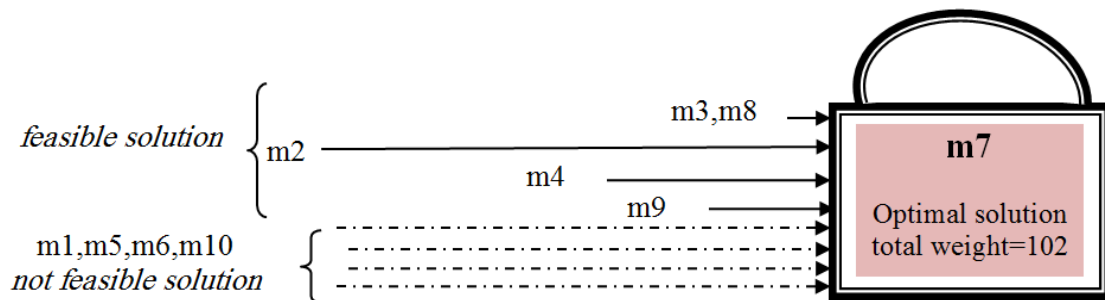


Figure (6) Representation of Optimal Solutions, Feasible and Not Feasible Solutions

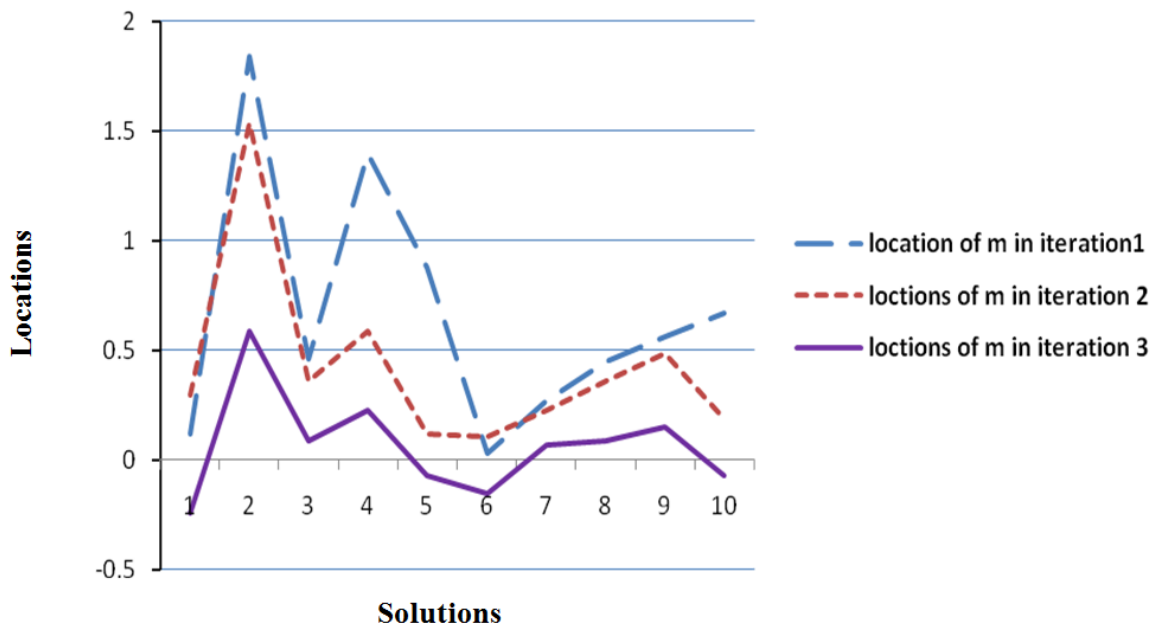


Figure (7) Relationship between the Solutions and Locations

5. Conclusions

In this research, we present proposed steps (update) for Chemical sensors, Heat sensors and location of Mosquito to solve Knapsack problem by using Mosquito algorithm. This research shows a great efficiency in solving this problem by Mosquito algorithm. The advantage of this research is to reach easily into optimal solution of Knapsack problem by partial swarm optimization as Mosquito algorithm for any size of a knapsack problem, so that, we used a big size as 8 items and size 10 of population that generate all possible solutions, there are 2^n subsets generated compared with the classical methods.

6. Recommendations

The Mosquito swarms algorithm considered as a new method to solve combinatorial optimization problem to reach the optimal solution of types for Knapsack problem such as Bounded knapsack problem, Subset-sum Knapsack problem, Change making Knapsack problem and Binary Multiple knapsack problem. Solving Knapsack problem by meta heuristic algorithms as Tabu search, Ant colony and Cuckoo search method.

REFERENCES

- [1] Akbar, M. M., Rahman, M. S., Kaykobad, M., Manning, E.G. and Shojaa, G.C.,2006, Solving the Multidimensional Multiple-Choice Knapsack Problem by Constructing Convex Hulls, ELSEVIER, Computers & Operations Research, 33, pp. 1259–1273. www.elsevier.com/locate/cor
- [2] Al-Bayati, A. Y. and Qubat, N. N., 2007, An Implementation of an Initial Scale in Solving Binary Knapsack Problem Using a Genetic Algorithm, Raf. J. of Comp. & Math's., Vol. 4, No. 2,pp.43-57.
- [3] Feng, X., Lau F. C.M. and Gao, D., 2009, A New Bio-inspired Approach to the Traveling Salesman Problem, ICST Institute Computer Sciences, Social Informatics and Telecommunications Engineering, pp.1310-1321.
- [4] Feng, X., Lau, F. C. M. and Yu, H., 2013, A Novel Bio-Inspired Approach Based on the Behavior of Mosquitoes, ELSEVIER, Information Sciences, 233, pp. 87-108. www.elsevier.com/locate/ins
- [5] Kennedy, J. and Eberhart, R. C., 1995, Particle swarm optimization, IEEE International Conference on Neural Network, pp. 1942-1948.
- [6] Martello, S. and Toth, P.,1990, Knapsack Problems Algorithms and Computer Implementations, John Wiley & Sons, New York, Singapore, England.
- [7] Pisinger, D., 2003, A Toolbox for Solving Knapsack Problem, DIKU, European Journal of Operational Research, Denmark, Vol. 10, No. 88, pp. 122-145.
- [8] Ruiz-Vanoye, J. A., Díaz-Parra, O., Cocón, F. and Soto, A, 2012, Meta-euristics Algorithms Based on the Grouping of Animals by Social Behavior for the Traveling Salesman Problem, International Journal of Combinatorial Optimization Problems and Informatics, Vol.3, No.3, pp.104-123.
- [9] Skiena, S.S., 2008, The Algorithm Design Manual, Second Edition, New York, Springer-Verlag London, pp 427.
- [10] Srisuwannapa, C. and Charnsethikul, P., 2007, An Exact Algorithm for the Unbounded Knapsack Problem with Minimizing Maximum Processing Time, Journal of Computer Science, Vol. 3, No. 3, pp.138-143.
- [11] Yan, X., Zhang C., Luo, W., Li, W., Chen, W. and Liu, H., 2012, Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 2, pp 264- 271. www.IJCSI.org.