

Providing new meta-heuristic algorithm for optimization problems inspired by humans' behavior to improve their positions

Azar, Adel¹; Seyedmirzaee, Seyedmoslem*²

1- Professor of management, Tarbiatmodares university, Tehran, Iran
E-mail: azara@modares.ac.ir

2- Postgraduate master, Tarbiatmodares university, Tehran, Iran
E-mail: s.seyedmirzaee@yahoo.com

ABSTRACT

Nowadays, meta-heuristic algorithms have earned special position in optimization problems, particularly nonlinear programming. In this study, a new meta-heuristic algorithm called "Improvement of position (IMPRO Algorithm)" is recommended to solve the optimization problems. This algorithm, similar to other heuristic and meta-heuristic algorithms starts with production of random numbers. However, the aforementioned algorithm is inspired by humans' behavior to enhance the position which coincidentally detects the best position with respect to various conditions. Subsequently, a position with the least standard deviation (0.01) is created surrounding random numbers around the situation in the form of normal distribution. Afterwards, the new top position is considered and the two positions are compared and the top position is determined. Thus the conditions which created the best position are situated as the motion factors. Naturally, the motion direction is toward the opposite direction of the lower position. Next, this algorithm changes a response during the search process and solves the problem by utilizing the firm decisions.

KEYWORDS

Meta heuristic; Improvement of position; Nonlinear; pattern recognition; approximate algorithm

1. INTRODUCTION

Optimization methods and algorithms are categorized into two groups: exact algorithms and approximate algorithms. The exact algorithms can precisely find the most optimized response; however, they have no efficiency for complicated optimization problems (NPH) and their solution time increases exponentially in these problems. Nevertheless, the approximate algorithms are able to find reasonable responses for few difficult optimization problems. In addition, the approximate algorithms are classified into two groups of heuristic and Meta heuristic algorithms. The two main issues of heuristic algorithms are their placement in local optimization and their disability in practical applications to optimize different problems. The heuristic algorithms or Meta heuristic are represented to eliminate these problem types of heuristic algorithms. In fact, Meta heuristic algorithms are one type of the approximate optimization

algorithms which have exit mechanism from local optimization and can be applied in an extensive spectrum range of optimization problems. Nowadays, there are numerous optimization problems around us which cannot precisely be solved, or it cannot be accomplished in a reasonable time. A number of these problems are mostly related to one kind of optimization problem, namely NP-HARD that cause numerous approximate methods to be presented to solve each problem. Meta heuristic algorithms are a part of these methods. Structure and logic of Meta heuristic algorithms are usually a part of the known structures. Thus by recognizing these logics, we can generate new hybrid algorithms to solve current problems. This trend continues and interesting algorithms will be created on a daily basis; moreover, a number of them will be gradually abolished and are merely worth to be reported as part of algorithms history. In recent decades, many algorithms have been proposed to solve optimization problems. However, presentation of new algorithms or their former improvement is continued. The reason is due to the fact that the efficiency of various algorithms decreases in different problems solving cases. Most important algorithms that many articles are written about them are stated below:

Genetic Algorithm (GA),[4],[6]; Imperialist Competitive Algorithm (ICA),[1]; Particle Swarm Optimization (PSO),[7] ; Ant Colony Optimization (ACO),[3]; Bee algorithm (BA),[10]; Simulated Annealing (SA),[2],[8]; Tabu Search (TS),[5]

2. EXPLANATION PRESENTATION, MAIN EXPRESSIONS AND SYMBOLS

It should be noted that criteria and indicators of social position of every society differ from one sociologist to another as far as their point of views; however with respect to various perspectives and ideas, we can reach a general conclusion. In assessing social movement which is a promotional issue from one social position to another, there are many criteria such as job, education ,wealth and other related issues [9]. Social status is a position and a standing in society that an individual holds in a specific time period. To evaluate social standing, our judgment is comprehensive based on a combination of all standards such as job, income, education and further, age, genetic background, religion, race, nationality which a number of these factors are improvable (such as education), several can be decreased (such as wealth or dignity) and many are unchangeable (such as lineage). Citizens always try to improve their social standings and they occasionally attempt to create new status by imposing minor changes such as dress codes. Moreover, if this change improves their status, they will continue the new modified trend; otherwise if this change leads to demotion of their social standing, a new direction will be continued as a result of learning from this condition change. Consequently as a general rule, one's social status is a function of gender, wealth, profession, education, personality, and hereditary issues.

In this article, a unique algorithm is designed and inspired by human behavior to promote the social status which can unlock a new vision to solve optimization problems. The algorithm flow-chart of "position enhancement" is illustrated in figure (1). In this algorithm, conditions mean quantities which are randomly allocated to variables and different social standings mean various quantities of objective function which are produced based on diverse conditions (different quantities).

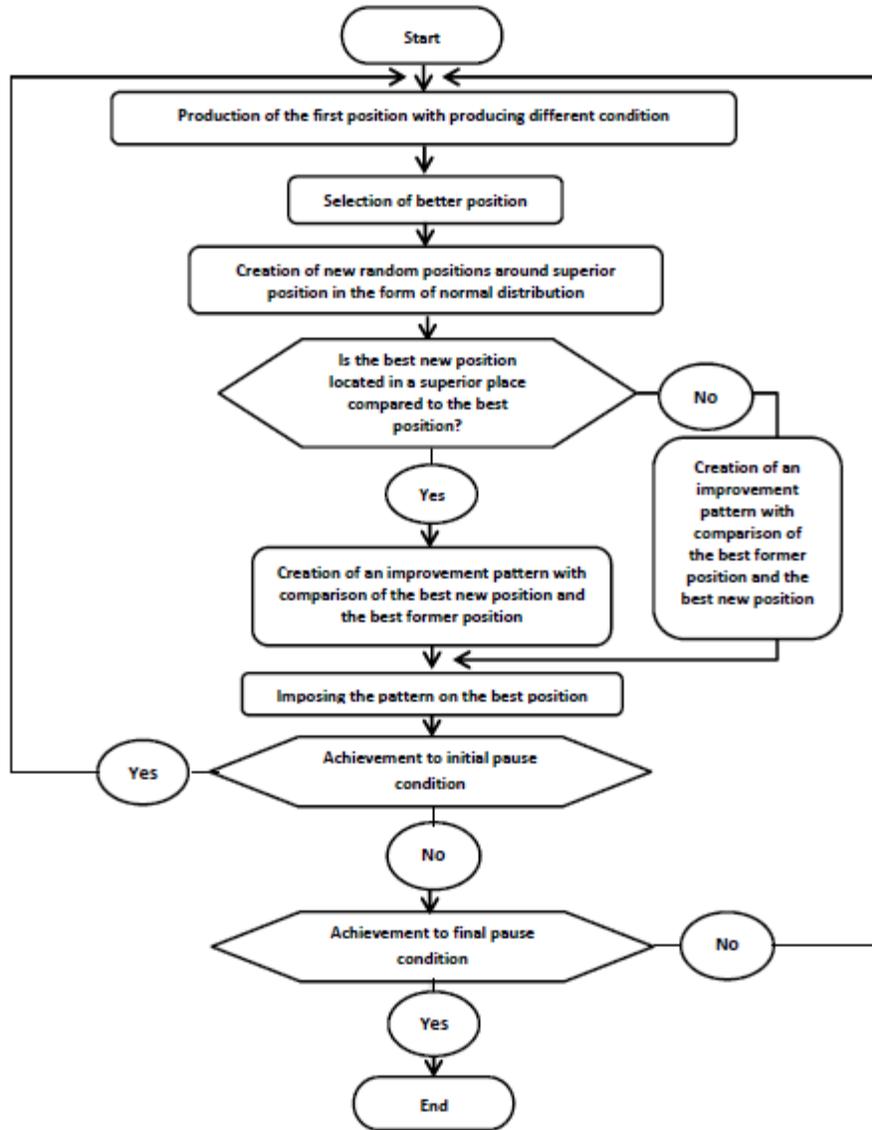


Figure 1.Flowchart of proposal algorithm

3. PROCEDURE

In this section, every step of the algorithm is thoroughly described and a simple function is designated as $P(x_i)$ along the steps to clarify the methodology of this algorithm.

$$\text{MaxP}(x_i) = x_1 - (x_1 \cdot x_2) \quad ; \quad 0 \leq x_i \leq 10$$

3.1 First Step: Generation of the First Positions

In this step, with the generation of N random numbers in the aforementioned limit problem, for example, in a [a b] limit, for any of (x_i) , a different condition will be created;thus different

positions will further be created. The variable (x_i) represents different condition and (p_i) signifies position. (Figure 2)

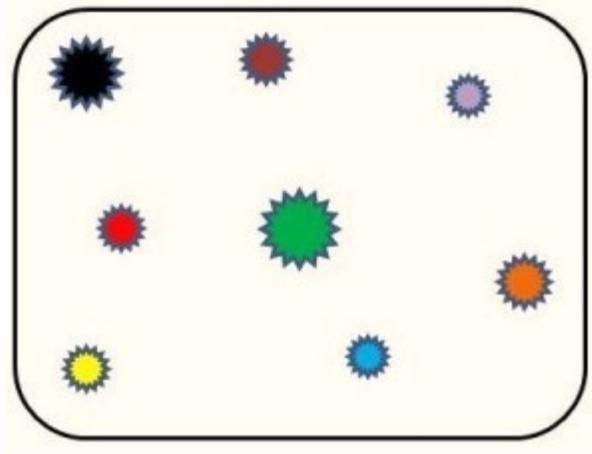


Figure 2. creation of different positions by generating random number

$$\begin{aligned}
 P_1 &= F (X_1 , X_2 , X_3, \dots X_n) \\
 P_2 &= F (X_1 , X_2 , X_3, \dots X_n) \\
 &\dots \\
 P_S &= F (X_1 , X_2 , X_3, \dots X_n) \\
 &\dots \\
 P_N &= F (X_1 , X_2 , X_3, \dots X_n)
 \end{aligned}$$

The first step description along an example:

With respect to the function $P(x_i)$, the first step is reviewed. As aforementioned, the initial positions are primarily created; thus to achieve this task, different conditions are ought to be produced. In fact, it is meant to generate random numbers. Therefore at this stage, 10 random numbers are produced as an example; as a result, different positions of $P(x_i)$ are created.

X1 =	8.1472	9.0579	1.2699	9.1338	6.3236	0.9754	2.7850	5.4688	9.5751	9.6489
X2 =	1.5761	9.7059	9.5717	4.8538	8.0028	1.4189	4.2176	9.1574	7.9221	9.5949
$P(x_i)$ =	-4.693	-78.85	-11.114	-35.199	-44.282	-0.408	-8.961	-44.611	-66.279	-82.931

3.2 Second Step: Selection of Superior Position

In this step, the superior position is determined according to objective function (min or max) and (P_s) represent superior standing (figure 3).

$$P_S = F (X_1 , X_2 , X_3 , \dots X_n)$$

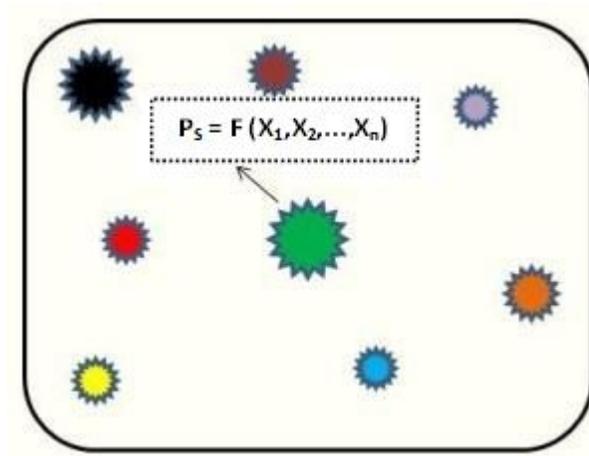


Figure 3. Selection of the best position

The second step description along an example:

In this step, a superior position is selected among the various positions which are produced. The superior position is called (P_s).

$$P_s(x_i) = -0.408$$

$$X_1 = 0.9754$$

$$X_2 = 1.4189$$

Up to this stage of the improvement of position (IMPRO Algorithm) algorithm, its similarity with other meta heuristic algorithms, such as imperialistic competitive algorithm, genetic algorithm is evident. This similarity is due to the fact that it starts with generation of random numbers and the best present position is identified.

3.3. Third Step: Creation of New Positions around the Best standing with respect To Improvement Condition in the Form of Normal Distribution

Normal distribution of random numbers is utilized in such a manner that the quantity of any variable in (p_s) is considered as the average and the rate 0.01 as standard deviation of random numbers. (Figure4), (Eq.1).

The number of normal random digits (K) that is created in this step is more reasonable to be consider as %1 of the first random numbers (n) for each (x_i), (Eq. 2).

The quantity of %1 is distinguished as suitable according to different experiential tests. It means that since the best position (p_s) is identified by imposing little changes in this condition (such as knowledge, dress code and other criteria, one can question the social status to determine whether or not the standing is promoted or demoted? Nevertheless, it is important to be able to determine a pattern by comparing these two positions.

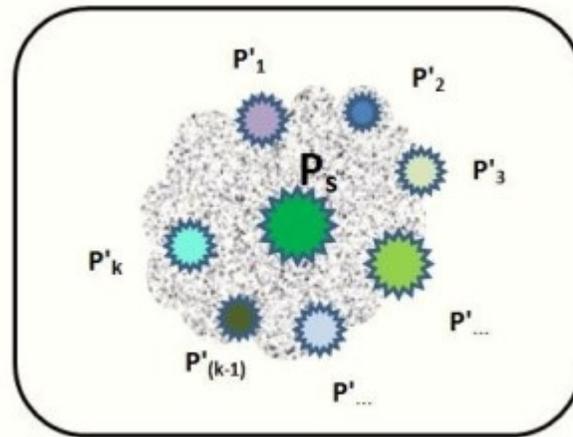


Figure 4. generation of random numbers with using normal distribution

(1)

[X₁'] ==> random N1~ (X₁,0.01)
 [X₂'] ==> random N2~ (X₂,0.01)
 ...
 [X_n'] ==> random Nn~ (X_n,0.01)

(2):

$$K = \frac{1}{100} (n)$$

The third step description along an example:

In this step of the algorithm, with respect to the best position which is initially identified as P_s (X_i) and the conditions which created these positions (X₁ , X₂), new positions are produced around every one of these conditions. This procedure is accomplished by generating the random numbers with respect to the averages of X₁ and X₂ and diversion of minimum criterion as well. It is noteworthy to point out that one should consider the minimum diversion criterion in order to create minimum distribution around the initial superior position. Moreover, various tests reveal that (0.01) as the diversion criterion is a reasonable response. In addition, the quantity of the numbers can be minimal namely 2 or 3 which should be produced in this stage. However, researchers have identified that 1% of the initial numbers are suitable. In this example for every variable, 3 numbers for the average and the specific criterion diversion are randomly produced.¹

[X ₁ '] ==> random N1~ (0.9754,0.01) =	1.4238	0.6324	0.9345
[X ₂ '] ==> random N2~ (1.4189,0.01) =	0.9633	1.7892	1.5346

3.4. Fourth Step: Comparison of new positions (P') created as a result of conditions change and selection of the best position among (P')s which is called (P*).

A change in condition means a change in the quantity of variables. Hence, variables include (X₁ , X₂ , X₃ , ...) and by using random number distribution normally around each variable, their

¹Since the normal numbers are randomly produced by the software, thus the average and criterion diversion of the produced numbers are nearly close to the desired quantities.

conditions are transformed to different ones (X'_1, X'_2, X'_3, \dots). Afterwards, the selection of the best position among (P') s which is called (P^*) is further accomplished (Figure 5).

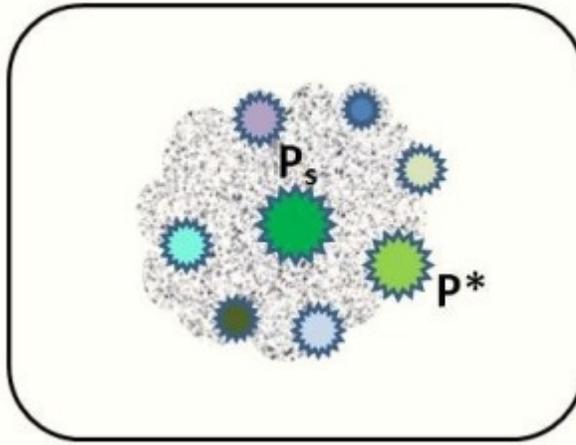
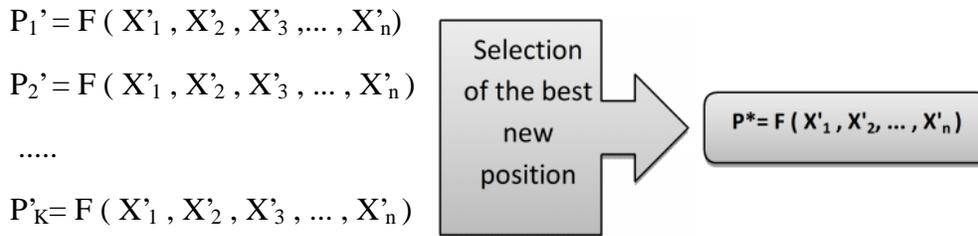


Figure 5. The selection of (P^*)



In this step, (P^*) and (P_s) are primarily compared. There are two states:

- 1- (P_s) > (P^*)
- 2- (P_s) < (P^*)

It is noticeable that in this step as different [X_i] are created, thus different (P') will be created as well.

Explanation: By a little modification in conditions, one's position changes are examined and once more another superior position is selected which is signified as (P^*).

The fourth step description along an example:

With respect to the similar conditions which is produced in the previous step (the produced random numbers), the superior position is designated as (P^*).

[X ₁] ==>random N1~ (0.9754,0.01) =	1.4238	0.6324	0.9345
[X ₂] ==> random N2~ (1.4189,0.01) =	0.9633	1.7892	1.5346
P'(x _i) =	0.0522	-0.4990	-0.4995
P*(x _i) =	0.0522		

3.5 Fifth step: Pattern Creation

In this step P_s and P* are primary compared. Furthermore in these step two states is illustrated.

- 1 -(P_s) > (P*)
- 2 -(P_s) < (P*)

According to each of the generated state (1 or 2) and by considering the desired goal (Max or Min), (P*) or (P_s) as the motion factor are selected. For example, if the first state is confirmed and the objective function is at maximum numerical value, it is obvious that we select (P_s) as the superior position. Next, variables of (P*) and (P_s) which have identical labels are compared with one another; consequently, a pattern for improving position is determined.

$$P_s = F (X_1 , X_2 , X_3, \dots) ; P^* = F (X'_1 , X'_2 , X'_3 , \dots)$$

For example, if objective function is at maximum and the comparison of (P_s) and (P*) leads to (P_s) > (P*), variables of (P_s) which include (X₁ , X₂ , X₃,...) are accepted as the present superior positional conditions. Afterwards, we compare them with (X'₁, X'₂, X'₃...) which are position variables of (P*). This step is to find the variables of motion direction of (X₁, X₂, X₃...). For example, if (X₁) < (X'₁), the (X₁) motion direction is toward "a" that is the beginning of the limit and the motion step will be a random number between (0, 1) during all steps. Furthermore, if (X₂) > (X'₂), the motion direction of (X₂) is toward "b" which is the end of limit interval (Figure 6).

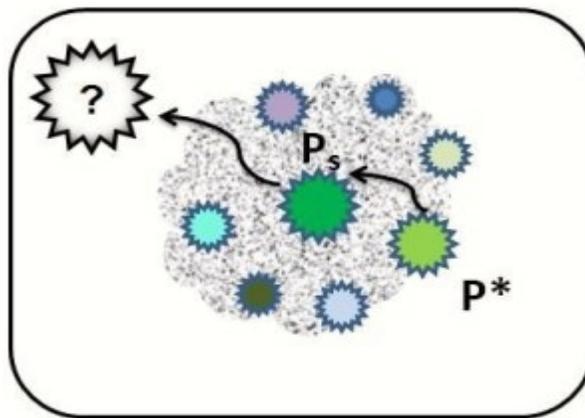


Figure 6.Determination of motion direction

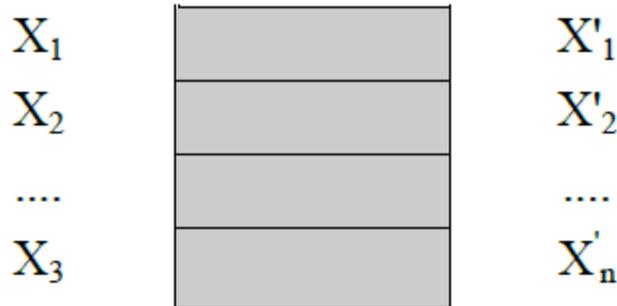
Thus in this example:



The position between (P_s) and (P^*) are considered; consequently, the variables of this position (P_s) are designated as the motion factor. The (P_s) is moved toward the improved position and whenever a superior position is found, it is replaced as the present superior position. (P_s)

Comparison between similar variables:

$$(<=>)$$



The fifth step description along an example:

Along the four former steps, it determined how to select the positions of P_s and P^* .

According the suggested algorithm, a comparison is made between these two positions and the conditions which caused the two positions to be created.

$X_1=0.9754$	$<$	$X_1' = 1.4238$
$X_2=1.4189$	$>$	$X_2' = 0.9633$
$P_s(x_i) = -0.4085$		$P^*(x_i) = 0.0522$

Since the position P^* is situated in a superior position among the positions of P_s and P^* , Thus the conditions which produce the positions of (X'_1, X'_2) are considered. In this stage, a motion pattern for the motion variables is determined.

As it can observed from the results of P^*, P_s , a superior position is created as the value of X'_1 increases and the value of X'_2 decreases based on the set of range of given values. Therefore in this example, X'_1 is moved toward the beginning of value range and X'_2 toward the end of value range.

In every loop, for the motion step of each variable, a random number between (0, 1) is allocated. Consequently in this example, a motion pattern is determined as below:



For instance, if two numbers such as 0.054 and 0.011 are randomly selected; then the variables motion of X'_1 is moved by step of 0.054 toward the value range of 10 and the variable X'_2 is moved by the step of 0.011 toward the value range of 0 as presented below:

$X'_1=$	1.4238	1.4778	1.5318	1.5858...
$X'_2=$	0.9633	0.9523	0.9413	0.9303 ...
$P(x_i) =$	0.0522	0.0704	0.0899	0.1105...

As it is evident in this example, the motion of variables occurs according to the given pattern. Hence, the function is moved toward improvement and as long as improvement is continued, the motion will not be stopped.

It should explained that when position P_s is selected as a superior position among positions of P_s and P^* , then the identified pattern should be imposed on variables of (X_1, X_2) .

3.6 Sixth step: Presentation Of initial pause conditions

If the improvement is stopped, the motion of variables will continue for several steps namely 3 steps in order to avoid the extreme position trap.

This position is matched with the improved condition which causes promotion; however, when the improvement is not obtained by continuing the conditions, one will continue changing the conditions toward taking several steps in order to obtain a relative assurance. If the conditions continue to be demoted, then the motion will be stopped.

Consequently, no pause occurs in the aforementioned function due to the fact that position improvement is continued.

$X'_1=$	1.4238	1.4778	1.5318	1.5858	1.6398	1.6938	1.7478	...	10
$X'_2=$	0.9633	0.9523	0.9413	0.9303	0.9193	0.9083	0.8973	...	0
$P(x_i) =$	0.0522	0.0704	0.0899	0.1105	0.1323	0.1553	0.1794	...	10

3.7 Seventh step: Presentation Of final pause conditions

This algorithm is formed with respect to the number of repeating loops and one can determine the conditions of the final pause. Furthermore, this step can be accomplished by computer software in such way that if improvement is not obtained after several repeated loops, the algorithm will be stopped.

4. TEST OF ALGORITHM

To examine the efficiency of improvement of position (IMPRO Algorithm) algorithm, six standard functions are examined.

The algorithm was run ten times and the average results are presented in the table below.(Table1) Tested functions are part of standard functions which are explained in genetic algorithm book and many related articles.

In this study, the tested function have two dimensions and it is attempted to present and select tested algorithms in such a manner they include different forms such as: limited range, infinite range, polynomial, trigonometric functions. It's noteworthy that positions justified in the first, third and fifth algorithm; in such a way that random numbers which are produced in the first step should be placed in the limit of [a, b].

In the third step, for regeneration of the random numbers with normal distribution around the position (P^*), standard division is rationally reduced to 0.01.However in this step, one should consider the limit response of condition algorithm that it is not exceeded from the limit. Finally in fifth step, the motion of each variable will be parallel to motion of the limit toward the end or the beginning. Therefore in this step of the algorithm, it will not go beyond the (X_i) limit. (X_i). If functions have limitation, they must be considered in these three algorithm steps.

Min $(1-x)^2+100(y-x^2)^2$ For $-10 < x,y < 10$	Max $\sin(x,y)+\cos(x-y)$ For $-1 < x,y < 1$	Min $20 + \sum_{i=1}^2 [x_i^2 - 10\cos(2\pi x_i)]$ For $-10 < x,y < 10$	Function
0.0005	1.841	2.1349e-005	Result average (improvement Algorithm)
0.0014	1.830	1.1509e-004	Result average (PSO)
0.0156	1.815	0.0025	Result average (GA)
1000	100	1000	Random number
100	10	100	loop
0	1.841	0	optimum value
Min $x\sin(4x)+1.1y\sin(2y)$ For $0 < x,y < 10$	Min $ x +\sin(x)$ For $-\infty < x < \infty$	Min $\sum_{i=1}^2 x_i^2$ For $-100 < x,y < 100$	Function
-18.5547	0	4.1723e-08	Result average (improvement Algorithm)
-18.5458	1.1882e-010	6.1723e-03	Result average (PSO)
-18.5547	3.7251e-006	3.0214e-01	Result average (GA)
1000	1000	1000	Random number
10	10	100	loop
-18.5547	0	0	optimum value

(Table 1).The result obtained from running the algorithm

5. CONCLUSIONS AND RECOMMENDATIONS:

All of Meta heuristic algorithms have strengths and weaknesses, further there is no algorithm that can merely present reasonable results. The improvement of position (IMPRO Algorithm) algorithm which is inspired by humans' behavior absolutely has strengths and weaknesses. As the improvement of position (IMPRO Algorithm) is a new recommended algorithm, it is hoped that researches examine this algorithm efficiency by various tests. This algorithm is being tested at the present and it is encouraged that after testing it, they exert changes and reexamine its results. For example, by a change in step length, result will differ or the number of random digits can influence the algorithm accuracy. Even it can change the standard deviation; moreover, the above algorithm should be examined in function with higher dimensions.

REFERENCE:

- [1] Atashpaz-Gargari E. and Lucas C., "Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition", IEEE Congress on Evolutionary Computation, Singapore, 2008.
- [2] Cerny, V. , A thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, Journal of Optimization Theory and Applications, Vol. , (1985)
- [3] Coloni, A., Dorigo, M., Maniezzo, V., 1991. Distributed optimization by ant colonies. In: Proceedings of the First European Conference on Artificial Life, Paris. 134–142.
- [4] DrAlex Rogers , CM2408 – Symbolic AI Lecture 8 – Introduction to Genetic Algorithms ,December 2002.
- [5] Glover F. and Laguna, M. ,Tabu Search, Kluwer Academic Publishers, 1997.
- [6] Haupt, R. L., and Haupt, S. E., "Practical Genetic Algorithms". John Wiley & Sons. Inc., New York.,(1998).
- [7] J. Kennedy and R.C. Eberhart, "Particle swarm optimization", in: Proceedings of IEEE International Conference on Neural Networks, Piscataway: IEEE,)1995(.
- [8] Kirkpatrick, S. ,Gelatt, C. D. , and Vecchi, M. P. , Optimization by simulated annealing, Science, Vol. 220, (1983)
- [9] Masters of Sociological Thought: Ideas in Historical and Social Context by Lewis A. Coser (Sep 1977)
- [10] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M. The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK,)2005(