

Improved Harmony Search with Chaos for Solving Linear Assignment Problems

***Osama Abdel-Raouf**

Department of Operations Research, Faculty of Computers and Information, Menoufia University, Menoufia, Shebin-El-coma, Egypt
E-mail:osamaabd@hotmail.com

Mohamed Abdel-Baset

Department of Operations Research, Faculty of Computers and Informatics, Zagazig University, El-ZeraSquare, Zagazig, Sharqiyah, Egypt
E-mail:henawy2000@yahoo.com

Ibrahim El-henawy

Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, El-ZeraSquare, Zagazig, Sharqiyah, Egypt
E-mail:analyst_mohamed@yahoo.com

Abstract— This paper presents an improved version of a harmony meta-heuristic algorithm, (IHSC), for solving the linear assignment problem. The proposed algorithm uses chaotic behavior to generation a candidate solution in a behavior similar to acoustic monophony. Numerical results show that the IHSC is able to obtain the optimal results in comparison with traditional methods (the Hungarian method). However, the benefit of the proposed algorithm is its ability to obtain the optimal solution within less computation in comparison with the Hungarian method.

Index Terms— Harmony Search Algorithm, Meta-Heuristics, Optimization, Assignment Problem, Chaos, Evolutionary Algorithms

I. Introduction

The assignment problem is one of the fundamental combinatorial optimization problems in the branch of optimization, in it there are a number of agents and a number of tasks [1]. Any agent can be assigned to perform any task, incurring some cost that may vary depending on the agent-task assignment. It is required to perform all tasks by assigning exactly one agent to each task and exactly one task to each agent in such a way that the total cost of the assignment is minimized [2]. Such a problem can be represented by a $n \times n$ or n^2 matrix which constitutes a factorial of n possible ways of making assignments. This problem can be mathematically represented as follows:

$$\min f(x) = \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \quad (1)$$

Subject to constraints

$$\sum_{j=1}^n x_{ij} = 1, \quad i=1,2,3,\dots,n \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j=1,2,3,\dots,n \quad (3)$$

where

$$x_{ij} = \begin{cases} 0, & \left(\begin{array}{l} \text{if the } i\text{th facility} \\ \text{is not assigned to } j\text{th job} \end{array} \right), \\ 1, & \left(\begin{array}{l} \text{if the } i\text{th facility} \\ \text{is ssigned to } j\text{th job} \end{array} \right).$$

The variable x_{ij} represents the assignment of an agent i to a task j , takes a value of 1 if the assignment is done, while 0 otherwise. This formulation allows also fractional variable values, but there is always an optimal solution, where the decision variables take integer values. This is because the constraint matrix is totally unimodular [3, 4]. The first constraint requires that every agent is assigned to exactly one task, and the second constraint requires that every task is assigned exactly one agent.

The assignment problem can be found in many scheduling applications, for example, in assigning planes or crews to commercial airline flights, trucks or drivers to different routes and men to offices and space to departments [2].

In this paper, an improved harmony search is proposed. In it, the nature of chaos is embedded with the

harmony search to help it to escape from local solutions. The proposed algorithm has been tested on three assignment problems. The computational results confirm that the proposed algorithm is able to obtain the optimal solutions with a few seconds, while other traditional methods consumes up to 180 seconds to obtain the optimal solutions.

This paper is organized as follows: after introduction, Literature review of solution techniques for linear assignment problem is shortly displayed. In section 3 the harmony search is briefly introduced. In section 4, the proposed algorithm is described, while the results are discussed in section 5. Finally, conclusions and future work are presented in section 6.

II. Literature Review of Solution Techniques for Linear Assignment Problems

Methods to solve the linear assignment problem can be classified (almost) as:

- i. Algorithm based on maximum flow (such as primal-dual method).
- ii. Algorithm based on shortest paths (such as dijkstra's method).
- iii. Algorithm based on the simplex method.
- iv. Mack's algorithm.
- v. Flood's algorithm (also called Hungarian method)

Using the flood's algorithm for solving this problem can result in a substantial saving in time over other traditional techniques; also we will use it to test the results of the proposed technique (IHSC). Flood's technique is a combinatorial optimization algorithm which was developed and published by [3]. This Technique involves a rapid reduction of the original matrix and finding a set of n independent zeroes, one in each row and column, which gives an optimal solution [3-4]. This technique is also known as the Hungarian method, which consists of the following steps [2]:

1. Row reductions by subtracting the minimum value in each row from all row values.
2. Column reductions by subtracting the minimum value in each column from all column values.
3. In the completed opportunity cost table, cross out all zeros, using the minimum number of horizontal or vertical lines.
4. If fewer than m lines are required (where $m =$ the number of rows or columns), subtract the minimum uncrossed value from all uncrossed values, and add this same minimum value to all cells where two lines intersect. Leave all other values unchanged and repeat step 3.

5. If m lines are required, the tableau contains the optimal solution and m unique assignments can be made. If fewer than m lines are required, repeat step 4.

III. Harmony Search Algorithm

Harmony search (HS) algorithm is a metaheuristic algorithm for solving optimization problems [5]. It has been applied to solve diverse types of problems in the past few years producing very effective results compared to other meta-heuristic algorithms and traditional techniques that computationally expensive.

In HS, the harmony memory (HM) stores feasible vectors, which are all in the feasible space. When a musician improvises one pitch, one of three rules is used:

- i. Generating any one pitch from his/her memory, i.e. choosing any value from harmony memory, defined as memory consideration;
- ii. Generating a nearby pitch of one pitch in his/her memory, i.e. choosing an adjacent value of one value from harmony memory, defined as pitch adjustments;
- iii. Generating totally a random pitch from possible sound ranges, i.e. choosing totally random value from the possible value range, defined as randomization.

Similarly, when each decision variable chooses one value in HS, it can apply one of the above mentioned rules in the entire algorithm. If a new harmony vector is better than the worst harmony vector in HM, then the new harmony vector replaces it. This procedure is repeated until a stopping criterion is satisfied [6]. The procedure of HS is composed of five steps:

Step 1: Parameters Initialization

The optimization problem is specified as follows:

$$\begin{aligned} & \text{Minimize } f(x) \\ & \text{Subject to} \\ & x_i \in X_i = 1, 2, \dots, N \end{aligned} \quad (4)$$

Where $f(x)$ is an objective function; x the set of each decision variable x_i ; N , the number of decision variables, X_i a set of possible values for each decision variable, within the range, $x_i^{\text{lower}} \leq x_i \leq x_i^{\text{upper}}$ where x_i^{lower} and x_i^{upper} are the lower and upper boundaries for each decision variable, respectively.

For the algorithm to perform the following parameters are required:

- Harmony Memory Size (HMS),
- Maximum number of Improvisations (MaxImp) or stopping criteria,

- Harmony Memory Considering Rate (HMCR),
- Pitch Adjusting Rate (PAR), that shows the rate of choosing a neighboring value
- Bandwidth vector (BW), which is an arbitrary distance bandwidth.

Step 2: Harmony Memory Initialization and Evaluation

The harmony memory (HM) is a memory location where all the solution vectors and corresponding objective function values are stored. In this step, the HM matrix is filled with solution vectors by taking the possible ranges into account and fitness function values are calculated for each solution vector.

Step 3: Improvise a new harmony from the HM

A new harmony vector $X' = (x'_1, x'_2, \dots, x'_N)$ is generated based on three mechanisms, namely, random selection, memory consideration, and pitch adjustment. The value of a design variable can be selected from the values stored in HM with a probability HMCR. It can be further adjusted by moving it to a neighbor value of a selected value from the HM with a probability of pitch adjusting rate (PAR), or, it can be selected randomly from the set of all candidate values without considering the stored values in HM, with the probability of q , where $q = (1 - HMCR)$ [6].

Step 4: Update the HM

If the new generated harmony vector is better than the current worst vector, based on the objective value and/or constraint violation, the new vector will replace the worst one.

Step 5: Stopping criterion check

If the stopping criterion (or maximum number of improvisations) is satisfied, the computation is terminated. Otherwise, Steps 3 and 4 are repeated.

HS has proved to be a powerful tool for solving several optimization problems [7-17]. It does not need any mathematical calculations to obtain the optimal solutions. In recent years, HS was applied to many optimization problems, demonstrating its efficiency compared to other heuristic algorithms and other Meta mathematical optimization techniques. Continuous development improvements to the algorithm and various applications to new types of problems (operations research, economy, computer science, civil engineering and electrical engineering), indicate that HS is a good choice. As a consequence, many studies have been proposed to increase its efficiency.

IV. The IHSCCH for Solving Linear Assignment Problem

Generating random sequences with a longer period and good consistency is very important for easily simulating complex phenomena, sampling, numerical analysis, decision making and especially in heuristic optimization [1]. Its quality determines the reduction of storage and computation time to achieve a desired accuracy [18]. Chaos is a deterministic, random-like process found in nonlinear, dynamical system, which is non-period, non-converging and bounded. Moreover, it depends on its initial condition and parameters [19]. Applications of chaos in several disciplines including operations research, physics, engineering, economics, biology, philosophy and computer science [20].

The nature of chaos is apparently random and changeable and it also possesses an element of regularity. Mathematically, chaos is a randomness of a simple deterministic dynamical system and chaotic system may be considered as sources of randomness [20]. A chaotic map is a discrete-time dynamical system $x_{k+1} = f(x_k)$, $0 < x_k < 1$, $k = 0, 1, 2, \dots$ running in the chaotic state. The chaotic sequence $\{x_k, k = 0, 1, 2, \dots\}$ can be used as a spread spectrum sequence and as a random number sequence.

Newly chaos is extended to various optimization areas, for example combinatorial optimization, because it can more easily escape from local minima in comparison with other stochastic optimization algorithms [21-23]. Using chaotic sequences in HS can be helpful to improve the global convergence, and to prevent getting stuck in local solutions than the classical HS algorithm which uses fixed values for HMCR, PAR and BW.

In the proposed chaotic HS algorithm, HMCR, PAR and BW values have been not fixed they were modified by selected chaotic maps.

The Chaotic Harmony Search algorithm for solving linear assignment problem, which is shown in figure 1 consists of the following four steps.

Step 1: set the parameters and initialize the HM by iterating the selected chaotic maps until it reaches the HMS, determining the fitness of each harmony vector as in equation 4 is done after.

Step 2: adjust PAR, BW and HMCR by the selected chaotic maps.

Step 3: updating harmony memory after evaluating all harmony vectors in equation 4.

Step 4: if the termination criterion is reached, return the best harmony vector; otherwise go to step 2.

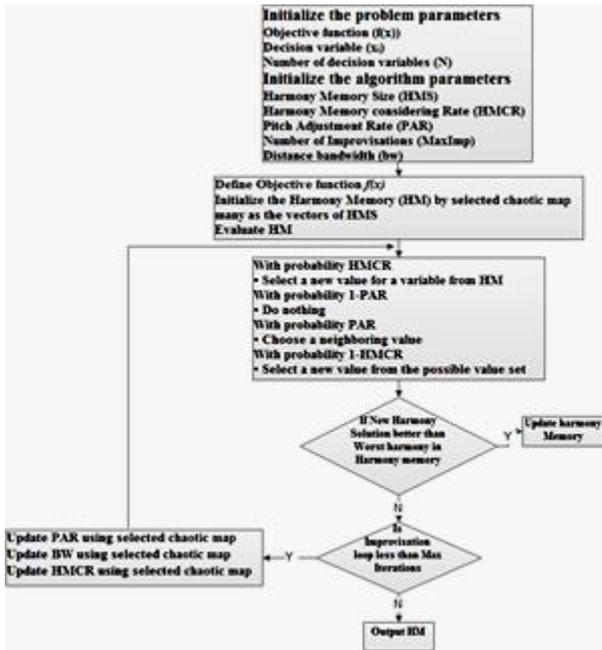


Fig. 1: Block Diagram of IHSCH

4.1 Handling Constraints

One of the well-known techniques of handling constraints is using penalty function, which transforms constrained problem into unconstrained ones, consisting of a sum of the objective and the constraints weighted by penalties. By using penalty function methods, the objectives are inclined to guide the search toward the feasible solutions. Hence, in this paper the corresponding objective function used in is defined and described as:

$$\min F(x) = f(x) + \lambda \sum_{n=1}^K \max(0, g_n) \quad (5)$$

where $f(x)$ is the objective function for assignment problem, λ is the penalty coefficient and it is set to a value of 10^7 in this paper, K is the number of constraints and g_n the constraints of the problem.

V. Computational Results

Numerous examples have been done to verify the weight of the planned algorithm. The initial parameters are set at $HMS = 50$ and $MaxImp = 10000$. The selected chaotic map for all examples is the logistic map, according to the following equation:

$$Y_{n+1} = \mu Y_n(1 - Y_n) \quad (5)$$

Clearly, $Y_n \in [0,1]$ under the conditions that the initial $Y_0 \in [0,1]$, where n is the iteration number and $\mu = 4$. All the experiments were performed on a Windows 7 Ultimate 64-bit operating system; processor Intel Core to Duo CPU 2.20 GHZ; 4 GB of RAM and codes were implemented in Visual c#.

5.1 Example 1

An electronics firm produces electronic components, which supplies various electrical manufacturers [2]. Quality control records indicate that different employees produce different numbers of defective items. The average number of defects produced by each employee for each of the five components is given in the table (1). It is required to determine the optimal assignment that will minimize the total average number of defects produced by the firm per month.

Table 1: The average number of defects produced by each employee

Employee	Component				
	A	B	C	D	E
1	80	75	90	85	0
2	95	90	90	97	0
3	85	95	88	91	0
4	93	91	80	84	0
5	91	92	93	88	0

5.2 Example 2

The Omega pharmaceutical firm has six salespersons, which the firm wants to assign six sales regions. Given their various previous contacts, the salespersons are able to cover the regions in different amounts of time [2]. The amount of time (days) required by each salesperson to cover each city is shown in the table (2). It is required to determine which salesperson should be assigned to each region to minimize the total round trip time.

Table 2: The amount of time (days) required by each salesperson to cover each city

Salesperson	Region					
	A	B	C	D	E	F
1	12	7	20	14	8	10
2	10	14	13	20	9	11
3	5	3	6	9	7	10
4	9	11	7	16	9	10
5	10	6	14	8	10	12
6	0	0	0	0	0	0

5.3 Example 3

A small commuter airline in North Carolina has twenty flight attendants whom it wants to assign to twelve monthly flight schedules in a way that will minimize the number of nights they will be away from their homes [2]. The numbers of nights each attendant must be away from home with each schedule are given in table (3). We want to identify the optimal assignments that will minimize the total number of nights the attendants will be away from home.

Table 3: The numbers of nights each attendant must be away from home with each schedule

Attendant	Schedule																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	11	11	6	10	5	8	6	11	8	15	8	12	7	11	12	10	17	6	9	10
2	14	14	5	12	17	2	10	9	12	6	9	8	9	12	12	13	10	12	7	11
3	9	19	15	7	10	8	11	12	10	9	17	9	4	14	14	8	9	13	11	9
4	11	6	8	14	9	10	18	13	15	5	12	10	7	15	13	11	10	17	15	15
5	5	8	6	10	12	6	13	9	6	14	6	17	15	16	16	7	5	13	16	16
6	10	12	11	9	19	13	4	6	7	13	2	8	6	17	17	2	8	6	17	7
7	9	9	10	7	16	6	11	7	5	7	4	9	9	17	17	4	9	9	7	17
8	19	6	5	6	13	9	6	10	6	6	10	8	14	18	18	12	8	5	18	8
9	8	8	9	11	13	10	7	6	12	10	10	12	8	19	10	5	10	7	15	15
10	14	9	8	4	12	5	10	11	5	13	11	8	9	10	11	6	7	5	13	16
11	10	11	7	7	14	8	9	10	11	9	14	12	19	10	12	11	8	6	13	17
12	9	10	13	8	15	7	8	7	10	12	8	17	11	15	14	5	10	7	15	15
13	14	6	10	15	8	6	11	8	13	8	12	6	13	14	15	6	13	5	16	16
14	11	5	12	7	6	10	15	12	6	9	8	13	14	14	16	13	8	6	10	17
15	14	11	7	10	8	11	12	10	9	17	9	11	15	12	13	4	9	9	6	17
16	18	6	10	14	8	6	11	8	17	8	12	6	9	10	15	17	5	13	9	10
17	19	9	10	15	7	13	5	9	10	5	7	17	7	6	8	9	17	14	7	6
18	10	7	6	8	9	6	13	17	16	8	9	6	9	10	4	6	7	15	19	10
19	11	9	10	14	6	7	7	9	10	4	6	7	6	6	9	7	5	3	12	16
20	12	10	7	13	9	10	10	12	19	7	16	9	10	10	12	14	7	6	9	11

Table 4: Optimal solution of selected examples

	Hungarian Method			IHSCCH		
	Optimal Cost	Optimal Assignment ($x \rightarrow y$, means assign x to y)	Time (sec)	Optimal Cost	Optimal Assignment ($x \rightarrow y$, means assign x to y)	Time (sec)
Example 1	328	1 \rightarrow B, 2 \rightarrow E, 3 \rightarrow A, 4 \rightarrow C, 5 \rightarrow D.	2	328	1 \rightarrow B, 2 \rightarrow E, 3 \rightarrow A, 4 \rightarrow C, 5 \rightarrow D.	1
Example 2	36	1 \rightarrow B, 2 \rightarrow E, 3 \rightarrow A, 4 \rightarrow C, 5 \rightarrow D, 6 \rightarrow F.	2	36	1 \rightarrow B, 2 \rightarrow E, 3 \rightarrow A, 4 \rightarrow C, 5 \rightarrow D, 6 \rightarrow F.	1
Example 3	100	1 \rightarrow 5, 2 \rightarrow 6, 3 \rightarrow 13, 4 \rightarrow 10, 5 \rightarrow 1, 6 \rightarrow 11, 7 \rightarrow 9, 8 \rightarrow 7, 9 \rightarrow 8, 10 \rightarrow 4, 11 \rightarrow 18, 12 \rightarrow 16, 13 \rightarrow 12, 14 \rightarrow 2, 15 \rightarrow 19, 16 \rightarrow 17, 17 \rightarrow 20, 18 \rightarrow 15, 19 \rightarrow 14, 20 \rightarrow 3.	131	100	1 \rightarrow 5, 2 \rightarrow 6, 3 \rightarrow 13, 4 \rightarrow 10, 5 \rightarrow 1, 6 \rightarrow 11, 7 \rightarrow 9, 8 \rightarrow 7, 9 \rightarrow 8, 10 \rightarrow 4, 11 \rightarrow 18, 12 \rightarrow 16, 13 \rightarrow 12, 14 \rightarrow 2, 15 \rightarrow 19, 16 \rightarrow 17, 17 \rightarrow 20, 18 \rightarrow 15, 19 \rightarrow 14, 20 \rightarrow 3.	2

Table 4 shows the solution results of selected examples using IHSCCH algorithm and Hungarian exact method. The results clearly indicate a match between the optimized values obtained using the two methods. On the other hand a considerable saving in computation time is achieved. The time saving becomes more realized as the scale of the problem gets larger. Since the assignment problem is NP-complete, so there is no known algorithm for solving this problem in polynomial time.

However, the proposed IHSCCH algorithm meta-heuristic proved its capability to solve different distinct assignment problems growing large to a size of 10000 \times 10000 when classical methods failed. A problem with such a scale may take time measured in days to be

solved traditionally if it is possible from the space complexity over view. The IHSCCH algorithm proved a remarkable save in computational time as shown in table (5).

Table 5: Solution time (Sec.)

Problem size	Hungarian method	IHSCCH
30 \times 30	180	2
40 \times 40	190	2
50 \times 50	201	3
100 \times 100	298	4
1000 \times 1000	399	9
10000 \times 10000	-	13

The reason for getting better results than the other algorithms considered is that the search power of the HS. To add to this, using Chaos helps the algorithms to escape from local solutions.

VI. Conclusion and Future Work

This paper introduced an improved harmony search algorithm integrated with chaos to update HM. The HM members were fine-tuned by the chaos operator to improve their affinities. Several examples have been used to verify the effectiveness of the proposed methods. The results proved a complete match between the proposed technique and previously used exact methods with a remarkable improvement in computational time and mathematical working out. The proposed algorithm managed to solve large scale problems that traditional method couldn't due to exponential growth in time and space complexities. The solution procedure will not face the same time waste in going through non converging iterations as traditional methods do.

Future work includes different chaotic maps to solve the linear assignment problem also we can solve variations of the assignment problem such as the quadratic assignment problem, the bottleneck assignment problem and categorized assignment problem by the proposed algorithm. Another direction can be study of different mechanisms for adapting the *HMCR*; *PAR*, and *bw* parameters.

Acknowledgments

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful comments.

References

- [1] M. Balinski, "Signature methods for the assignment problem," *Operations research*, vol. 33, pp. 527-536, 1985.
- [2] R. E. Burkard, M. Dell'Amico, and S. Martello, *Assignment problems*: Siam, 2009.
- [3] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, pp. 83-97, 1955.
- [4] M. Aigner, *Combinatorial theory* vol. 234: Springer-Verlag New York, 1979.
- [5] O. Abdel-raouf and M. A.-b. Metwally, "A Survey of Harmony Search Algorithm," *International Journal of Computer Applications*, vol. 70, 2013.
- [6] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, pp. 60-68, 2001.
- [7] Z. Geem, J. Kim, and G. Loganathan, "Harmony search optimization: application to pipe network design," *International journal of modelling & simulation*, vol. 22, pp. 125-133, 2002.
- [8] Z. W. Geem, C.-L. Tseng, and Y. Park, "Harmony search for generalized orienteering problem: best touring in China," in *Advances in natural computation*, ed: Springer, 2005, pp. 741-750.
- [9] J. H. Kim, Z. W. Geem, and E. S. Kim, "Parameter estimation of the nonlinear muskingum model using Harmony search," *JAWRA Journal of the American Water Resources Association*, vol. 37, pp. 1131-1138, 2001.
- [10] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Computers & Structures*, vol. 82, pp. 781-798, 2004.
- [11] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied mathematics and computation*, vol. 188, pp. 1567-1579, 2007.
- [12] M. G. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, pp. 643-656, 2008.
- [13] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems," *Applied Mathematics and Computation*, vol. 216, pp. 830-848, 2010.
- [14] Q.-K. Pan, P. Suganthan, J. Liang, and M. F. Tasgetiren, "A local-best harmony search algorithm with dynamic subpopulations," *Engineering Optimization*, vol. 42, pp. 101-117, 2010.
- [15] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 41, pp. 89-106, 2011.
- [16] D. Zou, L. Gao, J. Wu, S. Li, and Y. Li, "A novel global harmony search algorithm for reliability problems," *Computers & Industrial Engineering*, vol. 58, pp. 307-316, 2010.
- [17] Z. W. Geem, K. S. Lee, and Y. Park, "Application of harmony search to vehicle routing," *American Journal of Applied Sciences*, vol. 2, p. 1552, 2005.
- [18] R. Barton, "Chaos and fractals," *The Mathematics Teacher*, vol. 83, pp. 524-529, 1990.
- [19] G. Heidari-Bateni and C. D. McGillem, "A chaotic direct-sequence spread-spectrum communication

system," *Communications, IEEE Transactions on*, vol. 42, pp. 1524-1527, 1994.

- [20] Y. Ke, J. Cheng, and W. Ng, "MIC framework: an information-theoretic approach to quantitative association rule mining," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, 2006, pp. 112-112.
- [21] B. Alatas, "Chaotic harmony search algorithms," *Applied Mathematics and Computation*, vol. 216, pp. 2687-2699, 2010.
- [22] P. Gaspard, *Chaos, scattering and statistical mechanics* vol. 9: Cambridge University Press, 2005.
- [23] C. Letellier, *Chaos in nature* vol. 81: World Scientific Publishing Company, 2013.
- [24] D. Yang, G. Li, and G. Cheng, "On the efficiency of chaos optimization algorithms for global optimization," *Chaos, Solitons & Fractals*, vol. 34, pp. 1366-1375, 2007.

University. His current research interests are characterization of probability distribution, optimization, Intelligent Computing, Evolutionary Computation and decision support systems.

How to cite this paper: Osama Abdel-Raouf, Mohamed Abdel-Baset, Ibrahim El-henawy, "Improved Harmony Search with Chaos for Solving Linear Assignment Problems", *International Journal of Intelligent Systems and Applications (IJISA)*, vol.6, no.5, pp.55-61, 2014. DOI: 10.5815/ijisa.2014.05.05

Authors' Profiles



Ibrahim El-henawy received the M.S. and Ph.D. degrees in computer science from State University of New York, USA in 1980 and 1983, respectively. Currently, he is a professor in computer science and mathematics department, Zagazig University. His current research

interests are mathematics, operations research, statistics, networks, optimization, Intelligent Computing, Computer Theory, digital image processing, and pattern recognition.



Osama Abdel-Raouf received the M.S. and Ph.D. degrees in operations research and decision support systems from Monofia University. Currently, he is an associate professor in operations research department, Monofia University. His current research interests are evolutionary

algorithms, artificial intelligence, and decision support systems.



Mohamed Abd El-Baset received the B.Sc. degree in information system and technology from Zagazig University in 2006, and he obtained his M.S. degree in operations research and decision support systems from Zagazig University, in 2011. Currently, he is a teaching

assistant in operations research department, Zagazig