The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-1/W5, 2015
International Conference on Sensors & Models in Remote Sensing & Photogrammetry, 23–25 Nov 2015, Kish Island, Iran

OBCHS: AN EFFECTIVE HARMONY SEARCH ALGORITHM WITH OPPOSITION-BASED CHAOS-ENHANCED INITIALIZATION FOR SOLVING UNCAPACITATED FACILITY LOCATION PROBLEMS

Ali Asghar Heidari*, Omid Kazemizade, Rahim A. Abbaspour

School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Iran
(as_heidari, kazemizade.omid, abaspour)@ut.ac.ir

KEY WORDS: Uncapacitated Facility Location Problem; Optimization; Harmony Search Algorithm; Chaos; Opposition-based Learning

ABSTRACT:

In this paper, a continuous harmony search (HS) approach is investigated for tackling the Uncapacitated Facility Location (UFL) task. This article proposes an efficient modified HS-based optimizer to improve the performance of HS on complex spatial tasks like UFL problems. For this aim, opposition-based learning (OBL) and chaotic patterns are utilized. The proposed technique is examined against several UFL benchmark challenges in specialized literature. Then, the modified HS is substantiated in detail and compared to the basic HS and some other methods. The results showed that new opposition-based chaotic HS (OBCHS) algorithm not only can exploit better solutions competently but it is able to outperform HS in solving UFL problems.

1. INTRODUCTION

The HS optimizer thought up by (Geem et al., 2001) is one of the latest robust heuristics approaches that inspire the creativeness of musicians. Because of HS simplicity and easy carrying out, it has involved considerable attention and has been effectively employed for tackling problematic tasks (Degertekin, 2008). This algorithm is incapable to create a blameless sense of balance amongst exploration and exploitation (Saka, 2009). To relieve this problem, this article proposes an efficient modified HS-based optimizer to improve the performance of HS on complex spatial tasks in this research. For this aim, opposition-based learning (OBL) (Wang et al., 2011) and chaotic patterns (Tokuda et al., 1998) are utilized.

The key thought in OBL is to simultaneously consider an approximation and its matching opposite to recognize a better value for the existing solution (Wang et al., 2011). In many works, the scheme of opposite numbers is implemented to increase the convergence ratio of optimizers (Xu et al., 2014). Population initialization can be considered as a significant phase in HS since it can affect the fineness of the outcomes. Hence, chaos theory can also enhance the efficiency of the HS by improving the diversity of the members during iterations (Hong, 2009).

For this purpose, the proposed technique is examined against several UFL benchmark challenges in specialized literature. Several exact procedures developed before to handle the UFL task such as Lagrangean relaxation (Barcelo et al., 1990), branch and bound (Klose, 1998), dual approach (DUALLOC) (Erlenkotter, 1978), the primal-dual approaches (Korkel, 1989) and linear programming. It is recognizable that as this problem remains NP-hard (Sevkli et al., 2006); precise processes are not capable to tackle large applied problems competently. In the other hand, several attempts are offered to crack UFL task by utilizing metaheuristics. Some of main works are simulated annealing (SA) (Aydin et al., 2004), tabu search (TS) (Laurent et al., 2004) and genetic algorithms (GA) (Jaramillo et al., 2002). In this paper, the modified HS has been utilized and substantiated in detail and compared to the basic HS optimizer, GA and particle swarm optimization (PSO).

2. HARMONY SEARCH ALGORITHM

Harmony search (HS) algorithm can be regarded as a relatively well-established, robust meta-heuristic that has been implemented to tackle several complex tasks (Pan et al., 2011). This algorithm stimulates the improvisation of melody performers looking for an appropriate state of harmony. The HS optimizer reveals several benefits over the old-fashioned calculus-based problem solvers. The latter usually require some mathematical characteristics like differentiability and/or convexity, but HS is founded on a stochastic random exploration. It generates an updated candidate solution by examining all obtainable solutions.

The HS technique is implemented by the subsequent phases:

Step1: Initialize the target problem and required parameters.
Step2: Arrange the harmony memory. The opening harmony memory (HM) is randomly created as

$$HM = \left[\begin{array}{ccccc|c} D_1^1 & D_2^1 & \dots & D_{n-1}^1 & D_n^1 & f\left(D^1\right) \\ D_1^2 & D_2^2 & \dots & D_{n-1}^2 & D_n^2 & f\left(D^2\right) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ D_1^{HMS} & D_2^{HMS} & \dots & D_{n-1}^{HMS} & D_n^{HMS} & f\left(D^{HMS}\right) \end{array}\right] \quad (1)$$

where D_i^j indicates the *i*th variable in the *j*th solution matrix and HMS denotes the size of harmony memory.

Step 3: Improvisation of a firsthand harmony based on the HM. Improvisation can be defined as the creation of a novel harmony. Then, the reorganized vector x_{new} is generated

* Corresponding author

through these procedures: First, random selection, second, HM consideration, and then pitch adjustment.

$$D_i^{new} \leftarrow D_i(k),\ D_i(k) \in \{D_i(1),\ D_i(2),\ D_i(K_i)\} \quad (2)$$

$$D_i^{new} \leftarrow D_i(k),\ D_i(l) \in \{D_i^1,\ D_i^2,...,\ D_i^{HMS}\} \quad (3)$$

$$D_i^{new} \leftarrow D_i(l \pm 1),\ D_i(l) \in \{D_i^1,\ D_i^2,...,\ D_i^{HMS}\} \quad (4)$$

Generally, these procedures are observed as the principal terms of HS that can be represented via

$$\left.\frac{\partial f}{\partial D_i}\right|_{D_i = D_i(l)} = \frac{1}{K_i}.(1-HMCR) + \frac{n(D_i(l))}{HMS}.HMCR.(1-PAR) + \frac{n(D_i(l \pm 1))}{HMS}.HMCR.PAR, \quad (5)$$

where $(1/K_i)\times(1-HMCR)$ demonstrates the rate to pick out a appropriate value for D_i by using random selection process, $(n(D_i(l))/HMS)\times HMCR.(1-PAR)$ elects the ratio through HM consideration, and $(n(D_i(l \pm 1))/HMS)\times HMCR.PAR$ selects the rate via pitch adjustment. This process performs as follows

for j = 1 to D do
  if rand() ≤ HMCR
    $x_{new,j} = x_{r,j}$, r ∈ (1,2,...,HMS) % memory consideration step
    if rand() ≤ PAR % pitch adjustment process
      if rand() ≤ PAR
        $x_{new,j} = x_{new,j} - rand() \times bw$
      else
        $x_{new,j} = x_{new,j} + rand() \times bw$
      end if
    end if

else
    $x_{new,j} = x_{j,L} + rand() \times (x_{j,U} - x_{j,L})$ % random selection
  end if
end for

Step 4: Update of HM: When the fresh vector $D_i^{new} = (D_1^{new}, D_2^{new}, ..., D_n^{new})$ is entirely produced, it should be compared to the rest of the saved vectors in HM. If it is superior to the worst value in HM, it shall be reorganized.

Figure 1 illustrates the flowchart of HS in detail.

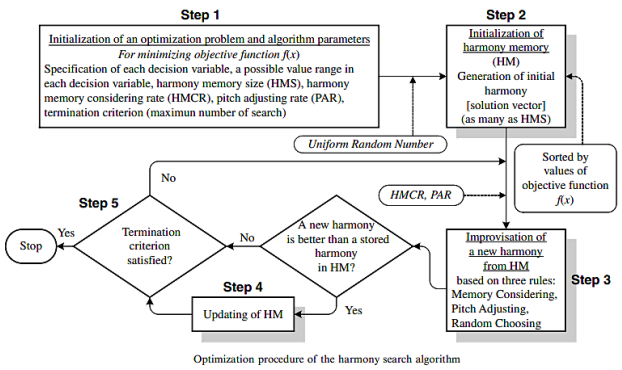

Figure 1. The HS flowchart (Lee et al., 2005)

The conventional HS is talented to reveal a superior performance compared to former optimizers in tackling constrained tasks. Nonetheless, premature convergence to regional bests can still be perceived in handling spatial optimization problems. In solving location problems, HS is usually incompetent to attain a correct balance among exploration and exploitation capacities. To relieve this problem, this paper suggests an effective improved HS-based technique to boost the performance of HS on complicated geospatial tasks like this research. For this goal, opposition-based learning (OBL) and chaotic motions can be used. In the next sections, the OBL is introduced to be used in HS initialization.

2.1 Mathematical definition of OBL

Let $Y = (y_1^1,...,y_1^d,...,y_1^n)$ denotes a real position in n-dimensional space; where $\{y_1^1,...,y_1^d,...,y_1^n\} \in R$ and $y_i \in [L,U] \forall_i \in \{1,...,d,...,n\}$. The opposite value $\breve{Y} = (\breve{Y}_1^1,...,\breve{Y}_1^d,...,\breve{Y}_1^n)$ may be stated via $\breve{y}_i = L + U - y_i$. Let f = (.) be a fitness value utilized to estimate the candidate's appropriateness. With this regard, if $f(\breve{Y}) \le f(Y)$, then the position Y can be substituted by $\breve{Y}$, if not, we will carry on with Y.

2.2 Opposition-based HS (OBHS)

The computational time of HS principally will be related to the spatial distribution of the early population. Hence, population initialization can be considered as a significant phase in HS since it can affect the fineness of the outcomes. The key thought in OBL is to simultaneously consider an approximation and its matching opposite to recognize a better value for the existing solution. In many works, the scheme of opposite numbers is implemented to increase the convergence ratio of optimizers.

2.3 Opposition-based Chaotic HS (OBCHS)

Analogous to other heuristics, two core stages are noticeable in HS optimizer: initialization and breeding new generations. In this work, the OBL scheme is employed in the initialization phase. Population initialization can be considered as a significant phase in HS since it can affect the fineness of the outcomes. Hence, chaos theory can also enhance the efficiency of the OHS by improving the diversity of the members during iterations (Heidari et al., 2015). In these experiments, the logistic map is utilized in the starting stage of OHS. The logistic map can be formulated as

$$x_{k+1} = \alpha x_k (1 - x_k) \quad (6)$$

where x_k signifies the kth number and k shows iteration number. Obviously, $x \in (0,\ 1)$ with opening condition $x_0 \in (0,1)$. In later employment, a = 4 is utilized (Heidari et al., 2015).

3. UFL PROBLEMS

Location problems have been classified as NP-hard problems (Sevkli et al., 2006). Hence, these problems have been treated broadly by various methodologies. All the qualified optimizers for tackling the UFL problems are either exact or heuristic procedures. Because the UFL is NP-hard, exact procedures are

incapable to tackle large scale location tasks efficiently. There are some investigations to treat UFL with metaheuristics such as SA and PSO algorithms. In the following, the improved algorithm is utilized to investigate the UFL problems.

In *UFL* problems with *m* sites and *n* customer, the cost of each site (f_{ci}) is fixed. There is a transference cost among both site and customer c_{ij}. There is no restriction for the location capacities, but the entire request of every customer should be allocated to one location. The intention is to determine a set of localities with a minimum cost. Here, the exact *UFL* is described as

$$Z = \min(\sum_{i=1}^{m}\sum_{i=1}^{n} c_{ij}.x_{ij} + \sum_{i=1}^{n} f_{ci}.y_i) \quad (7)$$

$$s.t. \quad \sum_{i=1}^{n} x_{ij} = 1, \quad (8)$$

$$0 \le x_{ij} \le y_i \in \{0;1\}, \quad (9)$$

where $i = 1,...,n;\ j = 1,...,m;$

x_{ij}: the quantity provided from facility i to client j;

y_i: whether facility i is verified $(y_i = 1)$ or not $(y_i = 0)$

Restriction (8) is intended to check that all requests have been encountered by the open locations, and limit (9) is considered to preserve integrity. As it is presumed that no facility has capacity restriction, the request size of every client is omitted, and then (8) verified without any request variable. The structure presented in (Sevkli et al., 2006) is also utilized here to handle the continuous UFL.

4. EXPERIMENTAL RESULTS

Here, the performance of new version of HS optimizer is investigated in detail. For experimentations of this paper, each algorithm is implemented by using MATLAB R2012a (7.14) on a T6400@4 GHz Intel Core (TM) 2 Duo processor PC with 4 GB RAM. The OCHS algorithm is employed to tackle 15 UFL benchmarks obtained from OR Library (Beasley, 2005). For each benchmark collection HS and OCHS procedures are experienced for 30 trials with 1.00E+03 iterations. The size and optimum solutions of benchmark cases are presented in Table 1. The outcomes are described with respect to quality indexes: Mean Relative Percent Error (*MRPE*) which is described as:

$$MRPE = \sum_{i=1}^{R} \left(\frac{(H_i - U) \times 100}{U_i} \right) / R \quad (10)$$

where H_i symbolizes the *ith* replication value and *U* is the finest value explored in past attempts and *R* shows the replications number. Other indexes are "Optimum Rate" denoted by (*HR*) and running time showed by (*RT*). In addition, *HR* represents the ratio of the total runs returned the best to the total trials. The greater *HR* and *MRPE* show the better quality of the solutions.

From the results, it can be detected that the HS outperforms conventional PSO in Cap 73, Cap 101, Cap 103, Cap 104, Cap 131, Cap 132, Cap 133 and Cap 134 with respect to the consumed time. With respect to HR values, HS can explore the locations with better quality in all cases exempt Cap 101, Cap 103 and Cap 131. In addition, HS outperforms PSO with regard to MRPE values. The MRPE of PSO is very high in tackling Cap-A, Cap-B and Cap-C problems and none of the efforts find the best answer. In general, the performance of PSO is not remarkable in comparison with HS. From the OBCHS results, it can be recognized that the quality is better than HS and PSO. It also is clear that OBCHS can outperform the PSO and HS. OBCHS can perform quicker with respect to the time. For instance, OBCHS can find the locations in 08.12 (s) in Cap-A test case, but the time of HS is not better than 14.72 (s). The efficiency of OBCHS is impressive compared to PSO and HS with regard to the all indexes.

		PSO (Sevkli et al.,2006)			
Problem	**Size**	**Optimum**	**MRPE**	**HR**	**RT**
***Cap*71**	16 × 50	932615.75	0.05	0.87	0.12
***Cap*72**	16 × 50	977799.40	0.07	0.80	0.16
***Cap*73**	16 × 50	1010641.45	0.06	0.63	0.27
***Cap*74**	16 × 50	1034976.98	0.07	0.73	0.21
***Cap*101**	25 × 50	796648.44	0.14	0.53	0.67
***Cap*102**	25 × 50	854704.20	0.15	0.40	0.85
***Cap*103**	25 × 50	893782.11	0.16	0.20	1.08
***Cap*104**	25 × 50	928941.75	0.18	0.70	0.47
***Cap*131**	50 × 50	793439.56	0.75	0.07	4.26
***Cap*132**	50 × 50	851495.33	0.78	0.00	4.56
***Cap*133**	50 × 50	893076.71	0.73	0.00	4.58
***Cap*134**	50 × 50	928941.75	0.89	0.10	4.15
Cap-A	100 × 1000	17156454.48	22.01	0.00	13.64
Cap-B	100 × 1000	12979071.58	10.75	0.00	16.58
Cap-C	100 × 1000	11505594.33	9.72	0.00	24.18

Table 1. Simulation outcomes obtained by PSO

		HS			
Problem	**Size**	**Optimum**	**MRP**	**HR**	**RT**
***Cap*71**	16 × 50	932615.75	0.04	0.90	0.12
***Cap*72**	16 × 50	977799.40	0.09	0.84	0.19
***Cap*73**	16 × 50	1010641.45	0.07	0.68	0.23
***Cap*74**	16 × 50	1034976.98	0.08	0.78	1.05
***Cap*101**	25 × 50	796648.44	0.13	0.42	0.65
***Cap*102**	25 × 50	854704.20	0.10	0.54	0.98
***Cap*103**	25 × 50	893782.11	0.17	0.12	0.99
***Cap*104**	25 × 50	928941.75	0.19	0.72	0.45
***Cap*131**	50 × 50	793439.56	0.76	0.07	4.13
***Cap*132**	50 × 50	851495.33	0.77	0.01	4.14
***Cap*133**	50 × 50	893076.71	0.72	0.00	4.13
***Cap*134**	50 × 50	928941.75	0.94	0.14	3.98
Cap-A	100 × 1000	17156454.48	12.74	0.00	14.72
Cap-B	100 × 1000	12979071.58	4.12	0.00	16.01
Cap-C	100 × 1000	11505594.33	2.41	0.00	22.41

Table 2. Simulation outcomes obtained by HS

		OBCHS			
Problem	**Size**	**Optimum**	**MRP**	**HR**	**RT**
***Cap*71**	16 × 50	932615.75	0.00	1.00	0.07
***Cap*72**	16 × 50	977799.40	0.00	1.00	0.06
***Cap*73**	16 × 50	1010641.45	0.01	1.00	0.14
***Cap*74**	16 × 50	1034976.98	0.00	1.00	0.45
***Cap*101**	25 × 50	796648.44	0.00	1.00	0.61
***Cap*102**	25 × 50	854704.20	0.00	1.00	0.85
***Cap*103**	25 × 50	893782.11	0.02	1.00	0.78
***Cap*104**	25 × 50	928941.75	0.00	0.99	0.12
***Cap*131**	50 × 50	793439.56	0.00	1.00	2.41
***Cap*132**	50 × 50	851495.33	0.01	1.00	3.98
***Cap*133**	50 × 50	893076.71	0.00	1.00	3.41
***Cap*134**	50 × 50	928941.75	0.00	1.00	2.12
Cap-A	100 × 1000	17156454.48	0.01	0.98	08.12
Cap-B	100 × 1000	12979071.58	0.01	0.97	13.02
Cap-C	100 × 1000	11505594.33	0.03	0.50	19.01

Table 3. Simulation outcomes obtained by OBCHS

For more investigation, OBCHS is compared with a GA exposed by (Jaramillo et al., 2002) and ESA suggested by (Aydin et al., 2004). The associated results are tabulated in Table 4 and Table 5.

Problem	Standard Deviation			
	GA	ESA	HS	OBCHS
Cap71	0.00	0.00	0.00	0.00
Cap72	0.00	0.00	0.001	0.00
Cap73	0.00033	0.00	0.00021	0.0001
Cap74	0.00	0.00	0.00	0.00
Cap101	0.00020	0.00	0.0001	0.00
Cap102	0.00	0.00	0.0001	0.00
Cap103	0.00015	0.00	0.0023	0.00
Cap104	0.00	0.00	0.00	0.00
Cap131	0.00065	0.00008	0.0004	0.00
Cap132	0.00	0.00	0.00	0.00
Cap133	0.00037	0.00002	0.00075	0.00
Cap134	0.00	0.00	0.00	0.00
Cap-A	0.00	0.000	0.00	0.00
Cap-B	0.00172	0.00070	0.001	0.00002
Cap-C	0.00131	0.00119	0.00	0.00

Table 4. Comparison of outcomes attained by all optimizers (GA and ESA results are from (Sevkli et al., 2006))

Problem	CPU Time			
	GA	ESA	HS	OBCHS
Cap71	0.287	0.041	0.042	0.014
Cap72	0.322	0.028	0.210	0.021
Cap73	0.773	0.031	0.215	0.028
Cap74	0.200	0.018	0.041	0.014
Cap101	0.801	0.256	0.704	0.098
Cap102	0.896	0.098	0.085	0.019
Cap103	1.371	0.119	1.012	0.064
Cap104	0.514	0.026	0.201	0.014
Cap131	6.663	2.506	4.122	0.514
Cap132	5.274	0.446	0.492	0.197
Cap133	7.189	0.443	0.785	0.312
Cap134	2.573	0.079	1.415	0.142
Cap-A	184.422	17.930	47.412	15.741
Cap-B	510.445	91.937	147.52	84.112
Cap-C	591.516	131.345	110.430	101.922

Table 5. Comparison of CPU time of the all optimizers (GA and ESA results are from (Sevkli et al., 2006))

From the outcomes, it is notable that ESA outperforms HS and HS overtakes GA based on standard deviation. The suggested approach executes better compared to the other methods in both indexes. In detail, with respect to CPU time, proposed OBCHS is more efficient than GA, ESA and HS. Particularly for Cap-C, ESA performs with more time than OBCHS. From the results, it can be observed that OBCHS has no deviation in tackling Cap 71, Cap 72, Cap 74, and Cap 101-134. These outcomes affirm that the convergence issue can be alleviated considerably with the proposed mechanism.

5. CONCLUSIONS

In this article, a new HS-based technique is utilized to tackle UFL tasks. The OBCHS has been developed by using chaos and OBL theme. OBCHS has been examined on some benchmark cases and best solutions are attained after a reasonable time. The comparative outcomes of novel procedure vividly affirm that early convergence issue is mitigated significantly. New HS algorithm not only can exploit better solutions competently but it is able to outperform HS in solving UFL problems. To the best of our knowledge, this research is the first implementation of chaos based HS stated for UFL in the expert literature. For future researches, OBCHS can be employed to realize other spatial optimization tasks.

REFERENCES

Aydin, M.E., Fogarty, T.C., 2004. A Distributed Evolutionary Simulated Annealing Algorithm for Combinatorial Optimization Problems. *J. of Heuristics* 10, pp.269-292.

Barcelo, J., Hallefjord, A., Fernandez, E. and Jrnsten, K., 1990. Lagrengean Relaxation and Constraint Generation Procedures for Capacitated Plant Location Problems with Single Sourcing. *O R Spektrum* 12, pp.79-78.

Beasley J.E., 2005. OR-Library: http://people.brunel.ac.uk/ ~mastjjb/jeb/info.html

Degertekin, S. O., 2008. Optimum design of steel frames using harmony search algorithm. *Structural and Multidisciplinary Optimization*, *36*(4), pp.393-401.

Erlenkotter, D., 1978. A dual-based procedure for uncapacitated facility location. *Operations Research* 26, pp.992-1009.

Geem, Z. W., Kim, J. H., & Loganathan, G., 2001. A new heuristic optimization algorithm: Harmony search. *Simulation* 76, pp.60–68.

Heidari, A. A., Abbaspour, R. A. and Jordehi, A. R., 2015. An efficient chaotic water cycle algorithm for optimization tasks. *Neural Computing and Applications*, doi: 10.1007/s00521-015-2037-2.

Hong, W. C., 2009. Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. *Energy Conversion and Management 50*(1), pp. 105-117.

Jaramillo, J.H., Bhadury, J., Batta, R., 2002. On the Use of Genetic Algorithms to Solve Location Problems. *Computers & Operations Research* 29, pp.761-779.

Klose, A., 1998, A Branch and Bound Algorithm for an UFLP with a Side Constraint. *Int. Trans. Opl. Res.* 5, pp. 155-168.

Korkel, M., 1989. On the Exact Solution of Large-Scale Simple Plant Location Problems. *European J. of Operational Research* 39(1), pp.157-173.

Laurent, M. and Hentenryck, P.V., 2004. A Simple Tabu Search for Warehouse Location. *European J. of Operational Research* 157, pp.576-591.

Lee K.S., Geem Z.W., 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering* 194, pp. 3902-3933.

Pan, Q. K., Suganthan, P. N., Liang, J. J., and Tasgetiren, M. F., 2011. A local-best harmony search algorithm with dynamic sub-harmony memories for lot-streaming flow shop scheduling problem. *Expert Systems with Applications 38*(4), pp.3252-3259.

Saka, M. P., 2009. Optimum design of steel sway frames to BS5950 using harmony search algorithm. *Journal of Constructional Steel Research* 65(1), pp.36-43.

Sevkli, M., & Guner, A. R., 2006. A continuous particle swarm optimization algorithm for uncapacitated facility location problem. In *Ant colony optimization and swarm intelligence*. Springer Berlin Heidelberg, pp. 316-323

Tokuda, I., Aihara, K., & Nagashima, T., 1998. Adaptive annealing for chaotic optimization. *Physical Review E*, *58*(4), pp.51-57.

Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., & Ventresca, M., 2011. Enhancing particle swarm optimization using generalized opposition-based learning. *Information Sciences 181*(20), pp. 4699-4714.

Xu, Q., Wang, L., Wang, N., Hei, X., & Zhao, L., 2014. A review of opposition-based learning from 2005 to 2012. *Engineering Applications of Artificial Intelligence* 29, pp.1-12.