Desalination and Water Treatment
www.deswater.com

doi: 10.5004/dwt.2017.21048

86 (2017) 305–310
August

Aquifer management through a new multi-objective hybrid algorithm

A. Manolis, E. Sidiropoulos*, Ch. Evangelides

Department of Rural and Surveying Engineering, Faculty of Engineering, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece, Tel. +30 2310 996143; email: nontas@topo.auth.gr

Received 7 January 2017; Accepted 17 June 2017

ABSTRACT

A multi-objective hybrid optimization algorithm is presented combining harmony search and the Nelder–Mead simplex algorithm, applied to the solution of a bi-objective groundwater management problem. The objective functions of the corresponding optimization problem include pumping and installation costs to be minimized, along with the sum of supply discharges that is to be maximized in an aquifer. The number and position of production wells is addressed as a special feature of the problem. Optimal trade-offs among these conflicting objectives are found by determining the corresponding Pareto front. The present multi-objective approach is compared with a harmony-based method of the literature. The comparison is performed both on the aquifer problem and two standard benchmark functions, with results that favor the present approach.

Keywords: Aquifer management; Harmony search; Optimization; Multi-objective; Nelder–Mead

1. Introduction

Evolutionary methods and algorithms are becoming increasingly popular in scientific and engineering applications involving optimization. Furthermore, in the past 15 years, the interest of the scientific community is focused on multi-objective applications. These are applications with several objectives to be satisfied. In such a case, a single solution that is optimal with respect to all objectives cannot be found, especially when these objectives are conflicting with each other. Therefore, in a multi-objective problem the term "optimal" cannot be used for a solution and the term "domination" is introduced instead. By definition, a solution dominates another one when the first solution is no worse than the second solution in all objectives and it is better in at least one objective [1]. Thus, for a problem that requires all objective functions to be minimized, a solution x_1 dominates another solution x_2 if:

$$\forall i \in \{1,\cdots,m\} : f_i(x_i) \le f_i(x_2) \wedge \exists j \in \{1,\cdots,m\} : f_j(x_1) \le f_j(x_2) \quad (1)$$

The solutions that are not dominated by any other solution are known as Pareto – optimal solutions or non-dominated solutions [2]. Any non-dominated solution is accep le since none of the objective functions can be improved in value without degrading some of the other objective values.

A popular algorithm for solving single-objective optimization problems is the Harmony Search algorithm (HSA) [3]. The algorithm has global characteristics, it has been studied intensively during the recent years and it has been applied to a great variety of optimization problems [4–6]. Several single-objective hybridizations of HSA have been proposed, especially with emphasis on specific mathematical or engineering problems [7–11].

The Nelder–Mead (NM) method [12] is a simple, direct search, local optimization technique. The method is easy to implement in practice, it does not require the derivatives of the objective function and it involves the concept of an evolving population. These features make it particularly sui le for combination with evolutionary algorithms, such as

* Corresponding author.

Presented at the 13th International Conference on Protection and Restoration of the Environment (PRE-XIII), 3–8 July 2016, Mykonos, Greece.

genetic algorithms and simulated annealing, as in [13] and [14], which deal with single-objective optimization of general benchmark functions. The only single-objective hybridizations of Nelder–Mead with Harmony Search appear to be one by Jang et al. [15] and another one by the present authors [16]. The question of comparison to other single-objective approaches has been addressed in those references. The present treatment is, therefore, devoted exclusively to a multi-objective version of the water resources problem.

Multi-objective hybridizations of HSA, such as [17] and [18], are a lot rarer than single-objective ones. No multi-objective hybridization of HSA with the Nelder–Mead method is to be found in the literature, to the best of the present authors' knowledge. In this paper, such hybridization is presented. The resulting hybrid algorithm combines the typical background ideas contained in [2] and [19] and especially in their HSA counterpart (MOHS) [20], with a sui ly applied and presented here Nelder–Mead local search. The new algorithm is named multi-objective harmony simplex hybrid (MOHSH) and yields improved performance. It is geared to the solution of an aquifer management problem. The algorithm is also tested on two standard characteristic benchmark functions.

Groundwater management leads to challenging non-linear and non-convex mathematical programming problems, for which the choice of heuristic evolutionary methods has been es lished, in order to avoid computation of derivatives and trapping in local optima [6]. Groundwater management via single-objective harmony Search has been treated in [6]. The characteristic problem presented here is bi-objective and the HSA is enhanced by the hybridization. The problem involves pumping cost minimization with both pumping rates and well locations as optimization parameters. Simultaneously, the total discharge of the wells is to be maximized leading to a multi-objective problem with conflicting objectives. Well locations are considered in the literature either as two additional continuous decision variables, namely the *X* and *Y* coordinates of the well [21] or as grid point coordinates in a discretized region of the study domain [22]. In this paper well locations are, more realistically, selected from a set of discrete candidate locations as in [23], but with a different selection scheme.

2. MOHSH algorithm

The MOHSH algorithm is the result of the hybridization of HSA and Nelder–Mead method and its application in multi-objective problems. The incorporation of local and global search strategies in a unified algorithm provides the necessary flexibility for dealing with an overall non-convex function landscape, while at the same time performing explorations in its locally convex regions. The local search by Nelder–Mead algorithm is not applied in every improvisation, giving time to HSA in order to alter the harmonies. The number of improvisations performed by HSA between the local searches is specified by the user through a parameter called NM_ST. In order to sort the obtained solutions, the fast non-dominated sorting and the crowded-comparison operator are used as they are described in [2]. Thus, every harmony is assigned two factors. The first one, *np*, is used to determine by how many other harmonies it is dominated. The second one, *cd*, indicates its relative distance from the next and previous harmonies according to all objective functions. Between two solutions, we preferred the one that has lower *np*. If the two solutions have the same *np* the one is preferred that has higher *cd*, meaning that it is located in a less crowded region.

In addition to the harmony memory (HM), two new slots are kept in memory: the archive [20] and the vault. The size of the archive (AS) is also user-defined and it is used in order to store the non-dominated harmonies. The vault is used to store the solutions of the archive before they are altered by the algorithm, in order to avoid applications of Nelder–Mead to the same harmonies. The steps of the proposed algorithm are as follows:

Step 1: Parameter setting

Define the objective functions, the decision parameters with their boundaries and the parameters of the algorithm.

Step 2: Initialization

Fill the HM matrix with random decision variables in the feasible space. The HM matrix can be described as follows:

$$\mathrm{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{\mathrm{HMS}} & x_2^{\mathrm{HMS}} & \cdots & x_n^{\mathrm{HMS}} \end{bmatrix} \tag{2}$$

where n is the number of variables and HMS is the number of harmonies stored in HM.

Step 3: Archive generation

Evaluate the HMS harmonies and sort them with fast non-dominated sorting. The non-dominated harmonies move to archive. The empty slots of the HM are filled with randomly generated harmonies until the size of the HM is HMS.

Step 4: Generation of new harmonies

A new harmony is generated HMS times as follows:

for i in range (n):

Select a random harmony from archive x_i^{AR}

Select a random harmony from HM x_i^{HM}

$$\begin{aligned} &\mathit{if}\ \mathit{rand}(0,1) < \mathrm{HMCR}: \\ &x_i^{\mathrm{new}} = x_i^{\mathrm{AR}} + \mathit{rand}(-1,1) * (x_i^{\mathrm{AR}} - x_i^{\mathrm{HM}}) \\ &\mathit{if}\ \mathit{rand}\,(0,1) \geq \mathrm{HMCR}: \\ &x_i^{\mathrm{new}} = x_i^{\min} + \mathit{rand}(0,1) * (x_i^{\mathrm{mas}} - x_i^{\min}) \end{aligned}$$

Step 5: Update of the archive

The new harmonies are evaluated and the combined HM, archive and new harmonies are sorted with fast non-dominated sorting and the crowded-comparison operator. The first AS harmonies move to archive. The harmonies in positions [AS + 1: HMS + AS] move to HM and the rest are discarded.

Step 6: Nelder–Mead implementation

This step is executed every NM iterations. The vectors that participate in the local optimization by the Nelder–Mead method are one member from the archive as guide and n members randomly picked from the HM. Whenever a sorting procedure is required from the algorithm, the fast non-dominated sorting and the crowded comparison operator are applied. In order to evaluate the new solutions according to their performance in all the population, the harmonies from the archive and HM are added to the sorting. Thus, a more accurate ranking is given to the vectors with little computational cost, since these harmonies are already evaluated.

The procedure is repeated for every harmony in the archive that is not in the vault. The algorithm is terminated if one of the following criteria is met [19]:

- The simplex size is smaller than a predefined value. This value can be relative to the expected precision.
- A user-defined number of maximum iterations or function evaluations is reached.

Step 7: Update of the archive and the vault

The harmonies that were obtained by the Nelder–Mead implementation, the harmonies from HM and the archive are sorted. The archive and the HM are recreated with the same manner as in step 3. The vault is being replaced by the archive in order to avoid reevaluation of these harmonies from the Nelder–Mead method. If the maximum number of iterations or function evaluations is reached, then the members of the archive compose the calculated non-dominated set. If not, the process is repeated from step 4.

3. Performance measures

In order to estimate the efficacy of the proposed algorithm, two performance metrics are used. The first one is the *C* metric function and it is defined as follows [20]: let $X',X'' \subseteq X$ be two sets of decision vectors. The function *C* maps the ordered pair (X',X'') to the interval [0,1].

$$C(X',X'') = \frac{\left|a'' \in X''; \exists a' \in X' : a' \prec a''\right|}{\left|X''\right|} \tag{3}$$

The value $C(X',X'') = 1$ means that all solutions of X'' are dominated by solutions from X'. The opposite condition $C(X',X'') = 0$ means that there is no solution in X'' dominated by any solution in X'. Notably, both $C(X',X'')$ and $C(X'',X')$ have to be calculated since $C(X',X'')$ is not necessarily equal to $1 - C(X'',X')$. This function is chosen because it does not require any knowledge of the true Pareto front in order to evaluate the algorithms in test.

The second performance metric Δ is used to measure the extent of spread achieved among the obtained solutions [2]. It is calculated as follows:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1}\left|d_i - \bar{d}\right|}{d_f + d_l + (N-1)\bar{d}} \tag{4}$$

where d_i is the Euclidean distance between consecutive solutions in the obtained set of solutions, $\bar{d}$ is the average of these distances, d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained set and *N* is the number of the solutions. In an ideally distributed set, all d_i would be equal to $\bar{d}$ and $d_f = d_l = 0$. Therefore the Δ metric would be 0. The metric takes higher values for worse distributions. The application of the Δ metric does not require the knowledge of the true Pareto front but it requires its extreme solutions.

4. Application on test problems

Performance of MOHSH is tested on two well-known multi-objective problems: the Fonseca–Fleming and the Kursawe problem. Both problems are solved by the present MOHSH and by the MOHS [19] algorithm. The parameters used in the algorithms are shown in Table 1.

The Fonseca–Fleming problem [24] is defined as follows:

$$\text{Minimize} = \left\{ \begin{aligned} f_1(x) &= 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) \\ f_2(x) &= 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right) \end{aligned} \right\} \tag{5}$$

The search domain is $-4 \le x_i \le 4$, $i = 1,2,3$ and the optimal solutions are found for $x_1 = x_2 = x_3$, $x_i \in [-1/\sqrt{3}, 1/\sqrt{3}]$.

Notably, the iterations of HSA in MOHSH are about half of the iterations in MOHS due to the addition of the Nelder–Mead method and the function evaluations it requires. The results are shown in Figs. 1 and 2. Both algorithms have a rather good approach of the true Pareto front while the differences between the calculated solutions are small. When the *C* metric is calculated, it shows that 28 solutions of MOHS are dominated by solutions from MOHSH while none of the solutions from MOHSH is dominated by solutions from MOHS. The Δ metrics are $\Delta_{\text{MOHSH}} = 0.57$ and $\Delta_{\text{MOHS}} = 0.71$.

The Kursawe problem [25] is defined as follows:

$$\text{Minimize} = \left\{ \begin{aligned} f_1(x) &= \sum_{i=1}^{2}\left[-10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right] \\ f_2(x) &= \sum_{i=1}^{3}\left[\left|x_i\right|^{0.8} + 5\sin\left(x_i^3\right)\right] \end{aligned} \right. \tag{6}$$

Table 1
Parameters setting

Parameter	Abbreviation	Value
Harmony memory size	HMS	200
Archive size	AS	100
Maximum functions evaluation	MFE	25,000
Harmony memory consideration rate	HMCR	0.9
Maximum iterations of Nelder–Mead	MINM	20
Number of iterations between Nelder–Mead	NM_ST	10

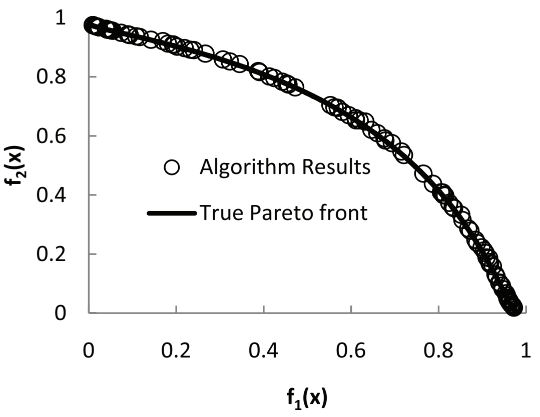


Fig. 1. MOHS application on Fonseca–Fleming test problem.

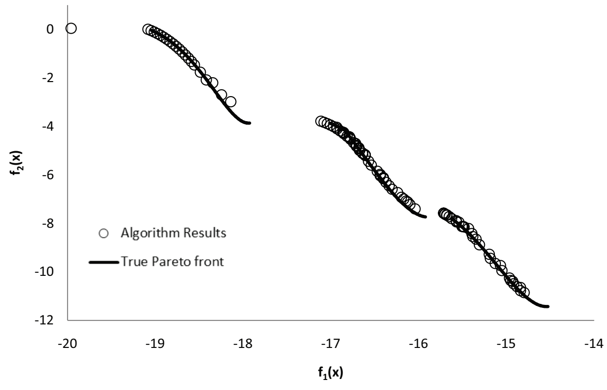


Fig. 3. MOHS application on Kursawe test problem.

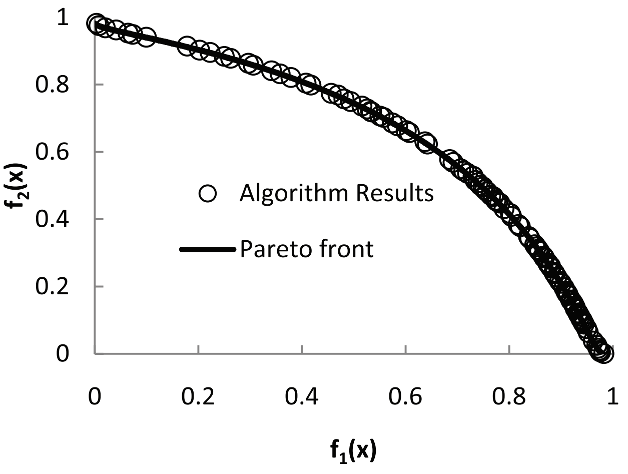


Fig. 2. MOHSH application on Fonseca–Fleming test problem.

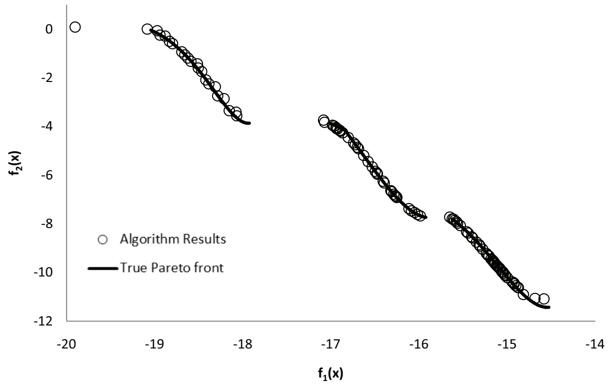


Fig. 4. MOHSH application on Kursawe test problem.

The search domain is $-5 \le x_i \le 5$, $i = 1,2,3$. The Pareto-optimal set is disconnected and the optimal solutions cannot be found with the use of a single equation. More information on the calculation of the optimal solutions can be found in [26].

The *C* metric shows that 47 solutions of MOHS are dominated by solutions from MOHSH while 16 solutions from MOHSH are dominated by solutions from MOHS. The Δ metrics are $\Delta_{\text{MOHSH}} = 0.60$ and $\Delta_{\text{MOHS}} = 0.72$. It can also be seen from Figs. 3 and 4 that the MOHSH describe the true Pareto front in a slightly more accurate way.

5. Aquifer management

The basic objectives of groundwater management are usually the minimization of pumping cost and the maximization of total pumping rate. In order to estimate the optimal trade-offs between these conflicting targets the MOHSH and the MOHS algorithms are used.

A semi-infinite unconfined aquifer is considered. The boundary condition of the aquifer is a Dirichlet boundary on the south (lake level is set to 100 m).

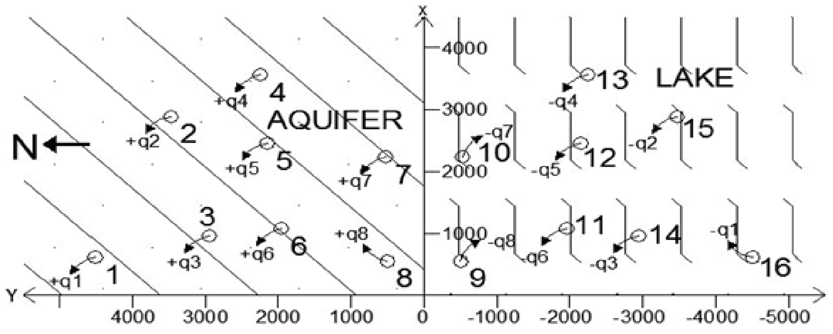
The management period is set to 1 year, and the flow is considered unsteady. The initial saturated thickness of the aquifer is set to D_0 = 100 m.The pumping wells are in the rectangle with coordinates (0; 0), (0; 6,000), (4,500; 6,000), (4,500; 0). The number of wells is considered unknown and is to be optimized as well. In order to simulate the Dirichlet boundary on the south of the aquifer, eight imaginary wells are placed, symmetrical to the boundary, of which the pumping rate will be added. The candidate locations of the wells are shown in Fig. 5. The first objective function is the sum of the pumping rates of the wells which has to be maximized (Eq. (7)). The second objective function (Eq. (8)) is the cost function of pumping, calculated in an arbitrary unit, which has to be minimized. The first term of the cost function is the fixed cost of installing a well at location *i* and the second represents the operating expense of the well:

$$\max f_1(q) = \sum_{i=1}^{n} q_i \tag{7}$$

$$\min f_2(q) = a_1 \sum_{i=1}^{n} u_i + a_2 \sum_{i=1}^{n} q_i u_i (H_i - h_i) \tag{8}$$

where

$$u_i = G(q_i - 0.01), \tag{8a}$$



Well no	X	Y
1	612	4506
2	2889	3477
3	956	2949
4	3560	2246
5	2458	2153
6	1074	1954
7	2233	526
8	554	502

Fig. 5. Plan view of the aquifer and locations of candidate wells.

with G being a step function of the type:

$$G(x) = \begin{cases} 1 \; if \; x \geq 0 \\ 0 \; if \; x < 0 \end{cases}$$

with the constraints:

$$h_i \geq h_0 \tag{9}$$

$$q_{\min} \leq q_i \leq q_{\max} \tag{10}$$

where $a_1 = 20{,}000$ is the installation coefficient and $a_2 = 0.256$ is the operating coefficient, n is the number of wells that will be installed, q_i is the pumping rate of the well i, u_i coefficient that is equal to 1 when the well i is used and equal to 0 when it is not used, H_i is the surface altitude at the well i, h_i is the water level at the well i, $h_0 = 90$ m is the minimum desired water level at the aquifer, $q_{\min} = 0$ m^3/d the minimum pumping rate of the well i and $q_{\max} = 30{,}000$ m^3/d the maximum pumping rate of the well i. It must be noted that, when $q_i < 0.01$ the well i is considered inactive and then u_i is set to 0, by virtue of its definition (8a).

The constraint (9) is referred to the maximum allowed drawdown $s = H - h_i = 10$ m. The drawdown is calculated by the simplified Theis equation.

$$s(r,t) = \frac{Q}{2\pi T} \ln\left(\frac{1.5\sqrt{tT/S}}{r}\right) \tag{11}$$

Since the aquifer is unconfined, the drawdown s is replaced with the corrected drawdown s' where s' is given as $s - s^2/2D_0$ with s given by Eq. (11) and D_0 the initial saturated thickness of the unconfined aquifer. Every time the algorithm calculates the objective function, the corrected drawdown in every well that is being used is calculated as well. In order to handle the maximum drawdown constraint, a penalty term z (Eq. (12)) is added to the objective function that takes an infinite value when $s_i^{'} > 10m$.

$$z = \begin{Bmatrix} inf & if \; s_i^{'} > 10m \\ 0 & if \; s_i^{'} \leq 10m \end{Bmatrix} \; \forall i = 1,2,\ldots,n \tag{12}$$

Thus the objective function takes its final form of Eq. (13).

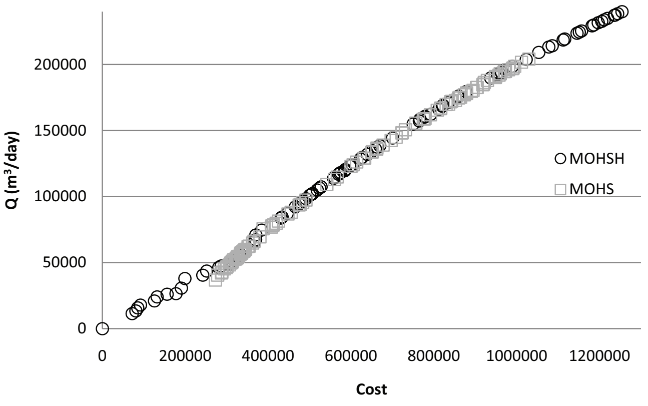


Fig. 6. Application of MOHSH and MOHS in aquifer management problem.

$$\min f(q) = a_1 \sum_{i=1}^{n} u_i + a_2 \sum_{i=1}^{n} q_i u_i (H_i - h_i) + z \tag{13}$$

In this manner, the algorithm rejects the solutions that violate the maximum drawdown constraint. The application of the constraint described in Eq. (10) is very simple for the HSA part of the algorithm, since HSA does not allow the decision variables to move outside of specified boundaries. In the NM part of the algorithm a barrier was implemented during the reflection and expansion steps of the method, forcing the variables to take values only in the feasible space.

The problem is solved by MOHSH and MOHS algorithms for comparison purposes and the solutions obtained are presented in Fig. 6. The C metric function shows that 32 solutions of MOHS are dominated by solutions from MOHSH while one solution from MOHSH is dominated by a solution from MOHS. Also as it is shown in Fig. 6, the solutions from MOHSH algorithm succeed on describing the whole extent of the Pareto front while the solutions from MOHS do not. The Δ metrics are $\Delta_{\text{MOHSH}} = 0.75$ and $\Delta_{\text{MOHS}} = 0.83$ meaning that MOHSH has a slightly better distribution of the solutions.

6. Conclusions and discussion

A multi-objective optimization approach is presented with a view to deal with a groundwater management problem. The problem considered involves both continuous and discrete variables. The HSA is combined with the Nelder–Mead method, thus producing a new multi-objective hybrid, for the aquifer management problem that constitutes the main theme of the present paper. Additionally, the proposed algorithm is also tested on two benchmark bi-objective problems of the literature. The comparisons are performed with a reference to an es lished harmony-based multi-objective algorithm that does not involve hybridization, in contrast to the present approach. The results are in favor of the proposed method, on the basis of es lished performance criteria for multi-objective algorithms.

References

[1] E. Zitzler, K. Deb, T. Lothar, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput., 8 (2000) 173–195.

[2] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput., 6 (2002) 182–197.

[3] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comput. Methods Appl. Mech. Eng., 194 (2004) 3902–3933.

[4] E. Sidiropoulos, Harmony search and cellular automata in spatial optimization, Appl. Math., 3 (2012) 1532–1537.

[5] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput., 188 (2007) 1567–1579.

[6] M.T. Ayvaz, Application of Harmony Search algorithm to the solution of groundwater management models, Adv. Water Res., 32 (2009) 916–924.

[7] G. Wang, L. Guo, 2013, A novel hybrid bat algorithm with Harmony search for global numerical optimization, J. Appl. Math., 2013 (2013) 696491.

[8] A. Kaveh, S. Talatahari, Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, Comput. Struct., 87 (2009) 267–283.

[9] S. Alaei, F. Khoshalhan, A hybrid cultural – harmony algorithm for multi-objective supply chain coordination, Sci. Iran. E, 22 (2015) 1227–1241.

[10] M. Fesanghary, M. Madhavi, M. Minary-Jolandan, Y. Alizadeh, Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems, Comput. Methods Appl. Mech. Eng., 197 (2008) 3018–3091.

[11] M.H. Mashinchi, M.A. Orgun, M. Mashinchi, W. Petrycz, Tabu-harmony search-based approach to fuzzy-linear regression, IEEE Trans. Fuzzy Syst., 19 (2001) 432–448.

[12] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J., 7 (1965) 308–313.

[13] S.S. Fan, Y. Liang, E. Zahara, A genetic algorithm and a particle swarm optimizer hybridized with Nelder-Mead simplex search, Comput. Ind. Eng., 50 (2004) 401–425.

[14] R. Chelouah, P. Siarry, Genetic and Nelder–Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions, Eur. J. Oper. Res., 148 (2003) 335–348.

[15] W.S. Jang, H.I. Kang, B.H. Lee, Hybrid Simplex-Harmony Search Method for Optimization Problems, IEEE Congress on Evolutionary Computation (CEC 2008) Proc., pp. 4157–4164.

[16] A. Manolis, E. Sidiropoulos, A harmony search hybrid for aquifer management, Proc. 12th International Conference on Protection and Restoration of the Environment, 2014, pp. 9–15.

[17] L.S. de Souza, R.B.C. Prudêncio, F.A. de Barros, A hybrid particle swarm optimization and harmony search algorithm approach for multi-objective test case selection, J. Braz. Comput. Soc., (2015) 21–29.

[18] Y. Li, X. Li, J.N.D. Gupta, Solving the multi-objective flow line manufacturing cell scheduling problem by hybrid harmony search, Expert Syst. Appl., 42 (2015) 1409–1417.

[19] H. Ghiasi, D. Pasini, L. Lessard, A non-dominated sorting hybrid algorithm for multi-objective optimization of engineering problems, Eng. Optim., 43 (2012), (39–59).

[20] W. Sheng, K. Liu, Y. Li, Y. Liu, X. Meng, Improved multiobjective harmony search algorithm with application to placement and sizing of distributed generation, Math. Prob. Eng., 2014 (2014) 871540.

[21] C. Huang, A.S. Mayer, Pump-and-treat optimization using well locations and pumping rates as decision variables, Water Resour. Res., 33 (1997) 1001–1012.

[22] S. Khalaf, M.I. Gad, Optimal well locations using genetic algorithm for Tushki Project, Western Desert, Egypt, Hydrol. Earth Syst. Sci., 11 (2014) 11395–11438.

[23] E. Sidiropoulos, P. Tolikas, Genetic algorithms and cellular automata in aquifer management. Appl. Math. Modell., 32 (2007) 617–640.

[24] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evol. Comput., 3 (1995) 1–16.

[25] F. Kursawe, A Variant of Evolution Strategies for Vector Optimization, Proc. 1st Workshop on Parallel Problem Solving from Nature, Vol. 496, 1991, pp. 193–197.

[26] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, West Sussex 2001, pp. 327–328.