International Journal of Computing and Optimization
Vol. 3, 2016, no. 1, 105 - 120
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ijco.2016.6410

An Improved Multi-Objective Particle Swarm Optimization

Hong-bin Bai

School of Science, Sichuan University of Science and Engineering, Zigong
Sichuan, 643000, P. R. China

Abstract

In this paper, a multi-objective particle swarm optimization based on extremal optimization with hybrid mutation and time-varied inertia (HM-TVWF-MOEPSO) method has been proposed in order to solve some of problems in the multi-objective particle swarm optimization and improve the performance of the algorithm.

Keywords: Multi-objective particle swarm optimization; Extremal optimization; Hybrid mutation; Time-varied inertia; Crowding distance

1. Introduction

Multi-objective Particle Swarm optimization (MOPSO) has been very popular tool in the multi-objective (MO) optimization field because its nature of the quick convergence and good diversity of optimal solutions in the Pareto front. Since PSO was proposed there have been numerous versions of MOPSO for solving the MO problems in the real life, e.g. MOPSO [11], Sigma-MPSO[12], Cluster-MPSO[39], Crowding-MOPSO[20], solving off-line two-dimensional flight path optimization [1], the flexible job shop scheduling problem [2], molecular docking problem [3], electrochemical machining process [4], a fuzzed MOPSO in electrical engineer [5], a combination of neural network modeling with MOPSO [6], MO pixel level image fusion [7], etc. Although PSO-based MO optimization technique has been extensive application, it has some drawbacks as follows; (1) the algorithm is easy to fall into the local optimal and lead to the premature convergence, and (2) the diversity of optimal solutions is lost. In order to overcome the above problems and improve the efficiency of the algorithm, a number of modified algorithms were increasingly produced, for example, avoiding premature convergence [8, 9, 10], guaranteeing the diversity [11, 12, 13], etc.

In the present article, a MOPSO, based on extremal optimization seen rarely, which is improved using hybrid mutation, time-varied inertia and crowding factor, is called Hybrid Mutation and Time-Varied inertia Multi-Objective Extremal Particle Swarm Optimization (HM-TVWF-MOEPSO), where extremal optimization, which has been connected with PSO for MO, is called a general-purpose local-search heuristic algorithm[34,35] . Additionally, similar to the ones suggested in [14, 15], in HM-TVWF-MOEPSO hybrid mutation operator is applied at the different search stage to increase the search efficiency. And the important parameter of PSO, i.e., the inertia weight ω , is also modified and applied. Its value will adaptively change with iterations. In the previous optimization methods genetic algorithm [16] has used the concept of time-varied inertia weight as well as PSO [17, 18, 19] etc, but most of them handled Single-Objective Optimizations. In the present paper this concept has been incorporated into MOPSO to achieve the trade-off between the global search and the local search in the problem space. Lastly, in order to improve the diversity in the Pareto-optimal solutions, a crowding-distance method [20] has also been adopted in HM-TV-MPEPSO.

In the next section the basic concepts of MO optimization are briefly presented. In Section 3 the PSO method and Extremal Optimization are described and briefly analyzed. In Section 4 the proposed method has been analyzed in detail. The obtained results and comparison to the other two state-of-the-art algorithms are presented in Section 5. In Section 6 the conclusions and future works of the paper are presented.

2. Multi-objective optimization

In general, a MO optimization problem can be mathematically stated by Eq.(1)[1]:

[1] Without loss of generality, only minimization problems will be assumed in this paper.

$$\text{Min } \vec{f}(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), ..., f_m(\vec{x})]^T \qquad (1)$$

where $\vec{x} = [x_1, x_{2,...,} x_n]^T \in X$ is an n -dimensional decision variable vectors and X is the decision variable space. $f_i : R^n \to R,\quad i = 1, 2, ..., m$ are the objective functions.

A few definitions about the concept of optimality will be described.

Definition 1 (Pareto Dominance): Give two vectors $\vec{x} = [x_1, x_2, ..., x_n]^T$ and $\vec{y} = [y_1, y_2, ..., y_n]^T$. Then, $\vec{x}$ *dominates* $\vec{y}$ *iff* $x_i \le y_i$, $i = 1, ..., n$, and $x_i < y_i$ for at least one component (denoted by $\vec{x} \prec \vec{y}$).

Definition 2 (Pareto Optimal): A decision vector $\vec{x} \in \chi \subseteq R^n$ is said to be *Pareto Optimal iff* there does not exist another decision vector $\vec{y} \in \chi$ such that $\vec{f}(\vec{y}) \prec \vec{f}(\vec{x})$,

and it is denoted by $\vec{x}^*$.

And the set of P^* is called *Pareto Optimal Set*, the mathematic form of which is

$$P^* = \{\vec{x} \in \chi \mid \vec{x} \textit{ is Pareto Optimal} \wedge \chi \textit{ is the feasible region}\}.$$

Definition 3 (Pareto Front): The *Pareto Front* is defined by:

$$PF^* = \left\{(f_1(\vec{x}),...,f_m(\vec{x}))|\vec{x} \in p^*\right\}.$$

by:

$$PF^* = \left\{(f_1(\vec{x}),...,f_m(\vec{x}))|\vec{x} \in p^*\right\}.$$

3. Extremal particle swarm optimization

3.1 *Particle Swarm Optimization (PSO)*

Kennedy firstly proposed PSO algorithm in 1995 [21], which is inspired by the "*birds flock*" and is also a population based heuristic search algorithm. The population of the potential solutions is called a *swarm*, where every individual is called a *particle* [22]. Originally PSO was applied to balance the neural networks, but it was soon proven to be very effective tool for solving MO optimization problems because of its quick speed convergence [23]. When PSO is used to a specific optimization problem, a population of random particles representing feasible solutions in the search space is initialized and the algorithm searches for optimal solutions by updating iterations.

Considering that there is a optimization problem with a D-dimensional search space, the i -th particle can be represented by the D-dimensional vector $X_i^{(t)} = (x_{i,1}^{(t)}, x_{i,2}^{(t)}, \ldots, x_{i,D}^{(t)})$, and its velocity is $V_i^{(t)} = (v_{i,1}^{(t)}, v_{i,2}^{(t)}, \ldots, v_{i,D}^{(t)})$ at t -th iteration; the best previous position of the i -th particle is recorded and denoted as $p_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,D})$, the global best position of the swarm found so far is represented by $Pg = (p_{g,1}, p_{g,2}, \ldots, p_{g,D})$,. The fitness value of each particle, which is usually evaluated according to putting its position into the given objective functions, is generally used to measure its performance. At the t iteration each particle can update its velocity and position by Eq.(2) and Eq.(3):

$$v_{i,j}^{(t+1)} = \omega^{(t)} v_{i,j}^{(t)} + c_1 r_1 (p_{i,j}^{(t)} - x_{i,j}^{(t)}) + c_2 r_2 (p_{g,j}^{(t)} - x_{i,j}^{(t)}), \tag{2}$$

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} + v_{i,j}^{(t+1)}, \tag{3}$$

where $i = 1, 2, \ldots, N_s$, N_s is the size of the population (which is set to 100 in this paper); $j = 1, 2, \ldots, D$, D is the dimensions of the search space; $t = 1, \ldots, T$, T is maximum number of iterations(which is set to 200 in this paper). $\omega^{(t)}$, c_1, $c_2 \geq 0$, $w^{(t)}$ is the inertia weight with time-varied. c_1 and c_2 are the cognitive and social factors respectively, which is set to $c_1 = c_2 = 2.0$ in this paper. r_1 and r_2 are two random numbers generated uniformly within the range [0, 1].

3.2 *Extremal Optimization*

PSO is effective for most of optimization problems, especial MO, but it is very sensitive to its parameters, i.e. ω, r_1 and r_2 , which affect its convergence behavior [24, 25] such that it usually suffers from being trapped into local optimum [41, 42], reducing its performance. However, extremal optimization method, which was proposed by Boettcher [34, 35], is inspired by equilibrium statistical physics [36].

EO was indicated to have strong local search ability [37]. Because it can update extremely undesirable variables of a single sub-optimal solution in the search space, replaced by new and random values. Additionally, the change of the fitness value of a variable can change the fitness values of its neighbors. Big and dynamic scale turbulence efficiently exploits many local optima [40]. The basic procedure of EO algorithm is seen as follows (in Fig.1)

Step 1. Set the index of the current particle $i = 1$.
Step 2. for $i = 1$ to $i = N_s$, for j =1 to D

·perform mutation for the position $x_{i,j} = (x_{i,1}, x_{i,2}, ..., x_{i,D})$.

·store the j-th new positions obtained, $x_{i,j}$ ($j = 1, ..., D$).

●evaluate the fitness values of the D new position according to $\Gamma_{i,j} = OBJ(x_{i,j}) - OBJ(p_g)$.

●find out the worst fitness value according to their fitness value, and the new position is set by the worst fitness value corresponding to position, $x_{i,j}$.

●If $OBJ(x_{i,j}) < OBJ(x_i)$, set $x_i = x_{i,j}$ and $OBJ(x_i) = OBJ(x_{i,j})$,otherwise, x_i unchanged.

Step 3. update p_i and p_g.
Step 4. return the results.

Fig.1. the basic procedure of EO

4. The proposed approach

4.1 hybrid HM-TVWF-MOEPSO algorithm

It's well known that PSO has great global search capability and CLS has strong local search ability. In this paper, a novel hybrid HM-TVWF-MOEPSO algorithm is proposed, which combines the merits of PSO, CLS, hybrid mutation method and time-varied inertia weight factor method. It makes full uses of the exploration capability of PSO and the exploitation ability of CLS. At the same time, similar to other MOPSO algorithms, it also employs Pareto-dominance for MO problems, crowding-distance mechanism for the selection of global best particles, and external archive storing the previous non-dominated solutions found for the convergence toward globally non-dominated solutions. So this hybrid algorithm can overcome the shortcoming of PSO and maintain the diversity of optimal solutions in the Pareto Front, enhancing the algorithm performance. The basic structure of HM-TVWF-MOEPSO is described in Fig.2.

4.2 Hybrid Mutation Strategy

PSO has good convergence properties in solving the single objective optimization problems, while its situation is contrary to MO cases; the convergence is often implemented at the cost of the diversity [11]. To let the MOPSO extensively explore the search space, while receiving better diversity, a hybrid mutation operator has been employed in HM-TVWF-MOEPSO derived from [34,39]. It combines Gaussian mutation with Cauchy mutation. X. Yao had been studied the mechanisms of Gaussian and Cauchy mutation operations [15].

And Cauchy mutation is better at coarse-grained search, while at fine-grained search Gaussian mutation can obtain good performance. Consequently, in the present algorithm Cauchy mutation is fist performed at the global search, and Gaussian mutation used at the local search. In this paper, the Gaussian mutation employs Eq. (4).

$$x_i' = x_i + N_i(0,1) \qquad (4)$$

where x_i' : the position coordinate of i -th particle after Gaussian mutation; x_i : the position coordinate of the i -th particle before mutation; $N_i(0,1)$ represents the Gaussian random number with means 0 and standard deviation 1, generated again for i -th particle, $i = 1, 2, .., N_s$.

The Cauchy mutation performs according to Eq. (5).

$$x_i' = x_i + \delta_i \qquad (5)$$

where δ_i represents the Cauchy random variable with the scale parameter equal to one, generated again for the i -th particle.

Algorithm HM-TVWF-MOEPSO: $O_f = \mathrm{HM-TVWF-MOEPSO}(N_s, N_a, T, D)$

/* O_f : the final output of the algorithm, N_s : size of the swarm, N_a :size of the external archive */

1. $t = 0$, initialize a swarm s_0 of N_s particles with random positions $x_{i,j}^{(0)}$ and velocities $v_{i,j}^{(0)}$ on D dimensions.

/* s_0 : swarm at iteration 0; $x_{i,j}^{(0)}$: the j -th coordinate of the i -th particle at the $t = 0$ iteration; $v_{i,j}^{(0)}$: the velocity of the i -th particle in the j -th dimension at the $t = 0$ iteration */

•Update $p_{i,j}^{(0)}$ using $p_{i,j}^{(0)} \leftarrow x_{i,j}^{(0)}$ and $A^{(0)}$ using $A^{(0)} \leftarrow non\_do\min ated(S_0)$ and $AS^{(0)} = \left|A^{(0)}\right|$.

/* $p_{i,j}^{(0)}$: the j -th coordinate of the best individual of the i -th particle; $A^{(0)}$: external archive at the iteration 0.*/

2. for $t = 1$ to $t = T$

- for $i = 1$ to $i = N_s$
  - . $P_g \leftarrow get\_gbest()$ /* return the global best*/
  - . $P_i \leftarrow get\_pbest()$ /*return the personal best*/
  - . adjust parameter(w_t) /*adjust the inertia weight w_t */
  - . update the velocities and positions of each particle /* according to Eq.(2) and(3) */

. Gaussian mutate(S_t).
. perform EO /"perform extreme local search"/
. $A_{t+1} \leftarrow non\_dominated(S_t \cup A_t)$ /* update the external archive*/
. Cauchy mutate (S_{t+1}) /* perform hybrid mutation*/

3. $O_f \leftarrow A_{t+1}$ and stop. /*return final optimal solutions*/

Fig. 2. The basic structure of HM-TVWF-MOEPSO

4.3 *Time-varied Inertia Factor*

Y. Shi [26] studied the importance of integration inertia weight w into traditional PSO in improving the convergence of PSO, it is indicated that w can balance between the global search and the local search, i.e. the higher values of w ensure in the global search at the initial iterations while the lower values provide in the local search around the current search zone at the later iterations. PSO with w may obtain good performance for optimization problems. Because of the complexity of search space of MO optimization problems, the parameter w can play the very vital effect in MOPSO algorithm.

In this paper, time-varied inertia weight factor [27] is introduced in HM-TVWF-MOEPSO, the value of $w^{(t)}$ will decrease linearly with the iteration from w_{max} to w_{min}. It changes according to Eq. (6).

$$w^{(t)} = (w_{max} - w_{min})\frac{T-t}{T} + w_{min}, \qquad (6)$$

where w_{max} and w_{min} are the maximum value and the minimum one of inertia weight, respectively (in this paper $w_{max} = 07, w_{min} = 0.4$ as suggested [33]). T is the maximum iteration number. The function of adjust parameter ($w^{(t)}$) in Fig.2 accomplishes the process.

4.4 *Update external archive*

The external archive is used to store the non-dominated solutions, and the selection of the p_g solution is performed from it in HM-TVWF-MOEPSO, so it has very important role. In every iteration, a candidate solution may be added to the archive only if one of the following conditions is satisfied [28].

(a) If the archive is full, the candidate solution is non-dominated and it is in a less crowed region than at least one solution.

(b) If he archive is not full, the candidate solution is not dominated by any solution in the archive.

(c) The archive is empty.

4.5 *Global best selection*

It is difficult for MO optimization problems with the conflicting character among multiple objectives to select a single optimum solution as the global best p_g , because the selection need consider the two aspects of the convergence ability of algorithm and the diversity of solutions. To solve this problem, the idea of non-dominance is applied and the above archive is maintained, from which a candidate solution is selected as p_g , and the selection must satisfy some diversity measure. In this paper the diversity measure is similar to the crowding-distance [20].

5. Experimental results and discussion

5.1 *Benchmark Functions and Metrics*

To evaluate the performance of HM-TVWF-MOEPSO algorithm, its results are compared with the state-of-the-art algorithms in the area: NSGA-II [38] and MOPSO [11].The benchmark functions employed in this study are ZDT1, ZDT2 and ZDT4 [32]. They are two dimensional objective functions, which take the form:

And the following metrics are employed for the purpose of providing a quantitative comparison of results: success counting (which is a variation of the metric called "error ratio" [29]), inverted generational distance [29], two set coverage [30, 31], two set difference Hypervolume [30].

Definition 4 (Success Counting (SCC)): this measure counts the number of vectors (in the current set of non-dominated vectors available) that are members of the Pareto optimal set: $SC = \sum_{i=1}^{n} s_i$, where n is the number of vectors in the current set of non-dominated vectors available; s_i =1 if vector i is a member of the Pareto optimal set, otherwise s_i =0; it should then be clear that $SC = n$ indicates an ideal behavior.

Definition 5 (Inverted Generational Distance (IGD)): this measure estimates how far are each of its elements in the Pareto front produced by an algorithm with respect to those in the true Pareto front. IGD is defined as: $IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}$, where n is the number of elements in the true Pareto front and d_i is the Euclidean distance (measured in objective space) between each of these and the nearest member of the set of non-dominated vectors found by an algorithm. It should be clear that a value of IGD = 0 indicates that all the elements generated are in the true Pareto front of the problem.

Definition 6 (Two Set Coverage (SC)): SC is defined as the mapping of the order pair ($X^{'}, X^{''}$) to the interval [0, 1]: $SC(X^{'}, X^{''}) \triangleq \frac{|\{a^{''} \in X^{''}; \exists a^{'} \in X^{'} : a^{'} \prec a^{''}\}|}{|X^{''}|}$. If all points in $X^{'}$ dominate or are equal to all points in $X^{''}$, SC=1, and SC=0 otherwise. In general, $SC(X^{'}, X^{''})$ and $SC(X^{''}, X^{'})$ both have to be considered due to set intersections not being empty. If $SC(X^{'}, X^{''})$ = 0 and $SC(X^{''}, X^{'})$ = 1, $X^{''}$ is completely better than $X^{'}$; if $SC(X^{'}, X^{''}) > SC(X^{''}, X^{'})$, $X^{'}$ is relatively than $X^{''}$; if $SC(X^{'}, X^{''})$ >0.9 and $SC(X^{''}, X^{'})$=0 $X^{'}$ is nearly than $X^{''}$.

Definition 7 (Two Set Difference Hypervolume (HV)): this measure defines the area (for the two dimensional case) of objective value space covered by the solution of an algorithm(P_a^*). For example, a vector in P_a^* for a two-objective MO problem defines a rectangle bounded by an origin and ($f_1(\vec{x}), f_2(\vec{x})$). The union of all such rectangles' area defined by each vector in P_a^* is then the comparative measure and is defined as: $HV = \{\bigcup_i a_i \mid x_i \in P_a^*\}$, where x_i is a non-dominated vector in P_a^* and a_i is the hypervolume determined by the components of x_i and the origin. If $HV(X^{'}, X^{''})$ =0 and $HV(X^{''}, X^{'})$ <0, $X^{''}$ is better than $X^{'}$.

5.2 *Experimental Results*

For each benchmark function in Table 1 they are done 20runs per algorithm and 2000 function evaluations; and the parameters of each method (NSGA-II[2] denoted by A, MOPSO[3] denoted by B, and HM-TVWF-MOEPSO denoted by C in the below tables) are set as follows. NSGA-II: *popsize*=100, T=200 generations, crossover rate p_c =1.0, mutation probability $p_m = 1/codesize$ .MOPSO: $N_s = 100$, $T = 200$, mutation rate p_m =0.05.The parameters HM-TVWF-MOEPSO have been mentioned in the above contents. The results compared are indicated in the below tables.

[2] The code of NSGA-II is from http:// www.tik.ee.ethz.ch/pisa
[3] The code of MOPSO is from http://delta.cs.cinvestav.mx/~ccoello/EMOO

5.3 *Discussion*

From Table 1, it' known that the results generated by HM-TVWF-MOEPSO is the best (on average) with respect to the three benchmark functions. Obviously, the Pareto front of HM-TVWF-MOEPSO is the closest to the true Pareto front.

Metrics		Functions								
		ZDT1			ZDT2			ZDT4		
		A	B	C	A	B	C	A	B	C
SCC	Best	37	0	71	30	0	93	0	0	100
	median	20	0	32	0	0	40	0	0	95
	Worst	9	0	5	0	0	1	0	0	2
	average	21	0	**35**	7	0	**42**	0	0	**77**
	Std. dev.	7.5	0	18.3	10.2	0	35.7	0	0	25.1
IGD	Best	0.0008	0.0238	0.0009	0.0009	0.0291	0.0008	0.0125	4.6409	0.0013
	median	0.0008	0.0274	0.0009	0.0720	0.1087	0.0009	0.1306	11.780	0.0009
	Worst	0.0010	0.0383	0.0013	0.0744	0.3492	0.0316	0.3217	14.837	0.0773
	average	0.0009	0.0279	**0.0008**	0.0507	0.1637	**0.0044**	0.1503	9.8905	**0.0032**
	Std. dev.	0.0009	0.0041	0.0008	0.0351	0.0931	0.0093	0.0958	3.891	0.0081

Table 1.Comparison of results based on metrics of SCC and IGD for ZDT1, ZDT2 and ZDT4. (Where A denotes NSGA-II, B denotes MOPSO, and C denotes HM-TVWF-MOEPSO)

Functions	Metric **SC**	Methods		
	SC(X)	A	B	C
ZDT1	A	0.00	1.00	0.03
	B	0.00	0.00	0.00
	C	0.72	0.99	0.00
ZDT2	A	0.00	1.00	0.01
	B	0.00	0.00	0.00
	C	0.90	1.00	0.00
ZDT4	A	0.00	1.00	0.00
	B	0.00	0.00	0.00
	C	0.93	0.00	0.00

Table 2.Comparison of results based on metrics of SC for ZDT1, ZDT2 and ZDT4 (Where A denotes NSGA-II, B denotes MOPSO, and C denotes HM-TVWF-MOEPSO)

The comparison from the Table 2 indicates that HM-TVWF-MOEPSO is relatively better than NSGA-II on ZDT1 and ZDT2, moreover it is nearly better than NSGA-II on ZDT4; additionally, on the three benchmark functions NSGA-II is all completely better than MOPSO.

Functions	Metric **HV**	Methods		
	HV(X)	A	B	C
ZDT1	A	0.000000	-0.321268	0.001038
	B	0.000000	0.000000	0.002701
	C	-0.000834	-0.410913	0.000000
ZDT2	A	0.000000	-0.304775	0.000397
	B	0.000000	0.000000	0.000000
	C	-0.004926	-0.591763	0.000000
ZDT4	A	0.000000	-0.503104	0.000279
	B	0.000000	0.000000	0.000000
	C	-0.100313	-0.577481	0.000000

Table 3.Comparison of results based on metrics of HV for ZDT1, ZDT2 and ZDT4 (Where A denotes NSGA-II, B denotes MOPSO, and C denotes HM-TVWF-MOEPSO)

From the Table 3 we can conclude that the hypervolume corresponding to the front obtained from the union of HM-TVWF-MOEPSO and MOPSO fronts is bigger than the one corresponding to the front of HM-TVWF-MOEPSO.

6. Conclusions and discussion

In this paper a modified MOPSO algorithm which employs extremal optimization, hybrid mutation, time-varied inertia weight and crowding distance based selection mechanism is presented. Extremal optimization used enhances effectively the local search ability of the algorithm, hybrid mutation introduced can realize to search at different stage for the solution space, and time-varied inertia weight can also adaptively balance between the global search and the local search, at the same time crowding distance mechanism helps to select the leaders from the archive storing non-dominated solutions (which is obtained at each iteration). Through comparison with two other algorithms, it is clearly found that HM-TVWF-MOEPSO is highly competitive, it' good in approximating Pareto optimal front as well as maintaining the diversity of the optimal solutions on the front. In future, the algorithm termination conditions and the system stability of stochastic search algorithm should be studied. In addition, it is also important to solve MO problems of the specific application with the improved PSO.

References

[1] L. Blasi, S. Barbato and M. Mattei, A particle swarm approach for flight path optimization in a constrained environment, *Aerosp. Sci. Technol.*, **26** (2013), no. 1, 128-137. http://dx.doi.org/10.1016/j.ast.2012.02.021

[2] G. Moslechi and M. Mahanam, A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search, *International Journal of Production Economics*, **129** (2010), no. 1, 14-22. http://dx.doi.org/10.1016/j.ijpe.2010.08.004

[3] S. Janson, D. Merle and M. Middendorf Molecular docking with multi-objective Particle Swarm Optimization, *Applied Soft Computing*, **8** (2008), no. 1, 666-675. http://dx.doi.org/10.1016/j.asoc.2007.05.005

[4] R. V. Rao, P. J. Pawar and R. Shankar, Multi-objective optimization of electrochemical machining process parameters using a particle swarm optimization algorithm, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, **222** (2008), 949-958.

[5] L. Wang and C. Singh Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm, *Electric Power Systems Research*, **77** (2006), no. 12, 1654-1664. http://dx.doi.org/10.1016/j.epsr.2006.11.012

[6] Y. Karpat and T. Ozel, Multi-objective optimization for turning processes using neural network modeling and dynamic-neighborhood particle swarm optimization, *Int. J. Adv. Manufacturing. Technol.*, **35** (2007), 234-247. http://dx.doi.org/10.1007/s00170-006-0719-8

[7] Y. Niu, L. Shen, X. Huo and G. Liang, Multi-objective wavelet-based pixel-level image fussion using multi-objective constriction particle swarm optimization, Chapter in *Multi-Objective Swarm Intelligent System*, **261** 2010, 151-178. http://dx.doi.org/10.1007/978-3-642-05165-4_7

[8] C. K. Monson, K. D. Seppi, Adaptive diversity in PSO, *Proceedings of the 8th Annual Conference on Genetic and evolutionary computation GECCO'2006*, ACM Press, New York, NY,, USA, (2006), 59-66. http://dx.doi.org/10.1145/1143997.1144006

[9] S.C. Esquivel, C. A. C. Coello, On the use of particle swarm optimization with multimodal functions, *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.2*, (2003), 1130-1136. http://dx.doi.org/10.1109/cec.2003.1299795

[10] Z. Michalewicz, Genetic Algorithms + Data Structure=Evolution Programs, Springer-Verlag, 1992. http://dx.doi.org/10.1007/978-3-662-02830-8

[11] C. A.C. Coello, G. T. Pulido, M. S. Lechuga, Handing multiple objectives with particle swarm optimization, *IEEE Transactions on Evolutionary Computation*, **8** (2004), no. 3, 256-279.
http://dx.doi.org/10.1109/tevc.2004.826067

[12] S. Mostaghim, J. Teich, Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization (MOPSO), *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03*, (2003), 26-33.
http://dx.doi.org/10.1109/sis.2003.1202243

[13] X. Li, A non-dominated sorting particle swarm optimizer for multi-objective optimization, Chapter in *Genetic and Evolutionary Computation (GECCO'2003)*, Vol. 2723, 2003, 37-48. http://dx.doi.org/10.1007/3-540-45105-6_4

[14] T. Ying, Y.P. Yang, J.C. Zeng, An enhanced hybrid quadratic particle swarm optimization, *Sixth International Conference on Intelligent Systems Design and Applications*, Vol. 2, (2006), 980-985.
http://dx.doi.org/10.1109/isda.2006.253745

[15] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, *IEEE Transactions on Evolutionary Computation*, **3** (1999), 82-102.
http://dx.doi.org/10.1109/4235.771163

[16] S.K. Pal, S. Bandyopadhyay, C.A. Murthy, Genetic algorithms for generation of class boundaries, *IEEE Transaction on Systems Man Cybernetics – Part B: Cybernetics*, **28** (1998), 816-828. http://dx.doi.org/10.1109/3477.735391

[17] A. Carlisle, G. Dozier, Adapting particle swarm optimization to dynamic environments, *Proceedings of International Conference on Artificial Intelligence (ICAI 2000)*, Las Vegas, Nevada, USA, (2000), 429-434.

[18] A. Carlisle, G. Dozier, Tracking changing extrema with adaptive particle swarm optimizer, *Proceedings of the 5th Biannual World Automation Congress*, (2002), 265-270. http://dx.doi.org/10.1109/wac.2002.1049555

[19] M.-R. Chen, Y. –Z. Lu, A novel elitist multiobjective optimization algorithm: Multiobjective extremal optimization, *European Journal of Operational Research*, **188** (2008), 637-651. http://dx.doi.org/10.1016/j.ejor.2007.05.008

[20] Carlo R. Raquel, Prospero C. Naval Jr., An effective use of crowding distance in multiobjective particle swarm optimization, *Proceedings of the 2005 conference on genetic and evolutionary computation,* (2005), 257-264. http://dx.doi.org/10.1145/1068009.1068047

[21] James Kennedy and Russell Eberhart, Particle swarm optimization, *IEEE International Conference Neural Networks*, (1995), 1942-1948. http://dx.doi.org/10.1109/icnn.1995.488968

[22] Praveen Kumar Tripathi, Sanghamitra Bandyopadhyay, Sankar Kumar Pal, Multi-objective Particle Swarm Optimization with time variant inertia and acceleration coefficients, *Information Sciences*, **177** (2007), 5033-5049. http://dx.doi.org/10.1016/j.ins.2007.06.018

[23] Russell C. Eberhart, Roy Dobbins and Patrick K. Simpson, *Computational Intelligence PC Tools*, Morgan Kaufman Publisher, 1996.

[24] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, *Information Processing Letters*, **85** (2003), 317-325. http://dx.doi.org/10.1016/s0020-0190(02)00447-7

[25] S. Naka, T. Genji, T. Yura, Y. Fukuyama, A hybrid particle swarm optimization for distribution state estimation, *IEEE Transactions on Power Systems*, 18 (2003), 60-68. http://dx.doi.org/10.1109/tpwrs.2002.807051

[26] Y. Shi, R. C. Eberhart, A modified particle swarm optimizer, *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, (1998), 69-73. http://dx.doi.org/10.1109/icec.1998.699146

[27] Y. Shi, R. C. Eberhart, Empirical study of particle swarm optimization, *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, Vol. 3, (1999), 101-106. http://dx.doi.org/10.1109/cec.1999.785511

[28] L. Wang, C. Singh, Environmental/economic power dispatch using a fuzzification multi-objective particle swarm optimization algorithm, *Electric Power System Res.*, **77** (2007), no .12, 1654-1664. http://dx.doi.org/10.1016/j.epsr.2006.11.012

[29] David A. Van Veldhuizen, *Multiobjective Eovlutionary Algorithms: Classifications, Analyses, and New Innovations*, PhD Thesis, Department of Electrical and Computer Engineering. Air Force Institute of Technology, Wright-Patterson AFB,Ohio, May, 1999.

[30] Eckart Zitzler, *Evolutionary Algorithms for Multiobjective Optimization:*

Methods and Applications, PhD Thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November, 1999.

[31] Eckart Zitzler and Lothar Thiele, Multiobjective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation*, **3** (1999), no. 4, 257-271. http://dx.doi.org/10.1109/4235.797969

[32] E. Zitzler, K. Deb, L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: empirical Results, *Evolutionary Computation*, **8** (2000), 173-195. http://dx.doi.org/10.1162/106365600568202

[33] F. van den Bergh, *An Analysis of Particle Swarm Optimization*, PhD. Thesis, Faculty of Natural and Agricultural Science, University of Pretoria, Pretoria, November, 2001.

[34] S. Boettcher, A.G. Percus, Extremal optimization: methods derived from co-evolution, *Proceedings of the Genetic and Evolutionary Computation Conference*, (1999), 825-832.

[35] S. Boettcher, A.G. Percus, Nature's way of optimizing, *Artificial Intelligence*, **119** (2000), 275-286. http://dx.doi.org/10.1016/s0004-3702(00)00007-2

[36] S. Boettcher, A.G. Percus, Optimizing Glasses with Extremal Dynamics. Chapter in *Computer Simulation Studies in Condensed-Matter Physics XVI*, Part II, Springer, Berlin, Heidelberg, 2006, 74–79. http://dx.doi.org/10.1007/3-540-26565-1_7

[37] S. Boettcher, A.G. Percus, Optimization with extremal dynamics, *Physical Review Letters*, **86** (2001), 5211-5214. http://dx.doi.org/10.1103/physrevlett.86.5211

[38] K. Deb, A. Pratap, S. Agarval, T. Meyarivan, A Fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.*, **6** (2002), 182-197. http://dx.doi.org/10.1109/4235.996017

[39] G. Toscano Pulido, C.A. Coello Coello, Using Clustering Techniques to Improve the Performance of a Particle Swarm Optimizer, Chapter in *Genetic and Evolutionary Computation – GECCO*, 2004, 225-237. http://dx.doi.org/10.1007/978-3-540-24854-5_20

[40] Min-Rong Chen, Xia Li, Xi Zhang, Yong-Zai Lu, A novel particle swarm optimizer hybridized with extremal optimization, *Applied Soft Computing*, **10** (2010), 367-373. http://dx.doi.org/10.1016/j.asoc.2009.08.014

[41] P.J. Angeline, Evolutionary optimization versus particle swarm optimization: philosophy and performance differences, Chapter in *Evolutionary Programming VII*, 1998, 661-610. http://dx.doi.org/10.1007/bfb0040811

[42] B. Liu, L. Wang, Y.H. Jin, F. Tang, D.X. Huang, Improved particle swarm optimization combined with chaos, *Chaos Solitons Fractals*, **25** (2005), no. 5, 1261-1271. http://dx.doi.org/10.1016/j.chaos.2004.11.095

Received: May 10, 2016; Published: July 6, 2016