Decomposition-Based Multiobjective Evolutionary Algorithm With an Ensemble of Neighborhood Sizes

Article *in* IEEE Transactions on Evolutionary Computation · June 2012
DOI: 10.1109/TEVC.2011.2166159

Shi-Zheng Zhao
Nanyang Technological University
20 PUBLICATIONS **1,386** CITATIONS
SEE PROFILE

Ponnuthurai N. Suganthan
Nanyang Technological University
434 PUBLICATIONS **23,008** CITATIONS
SEE PROFILE

Qingfu Zhang
City University of Hong Kong
217 PUBLICATIONS **10,776** CITATIONS
SEE PROFILE

Letters

Decomposition-Based Multiobjective Evolutionary Algorithm with an Ensemble of Neighborhood Sizes

Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, *Senior Member, IEEE,* and Qingfu Zhang, *Senior Member, IEEE*

***Abstract*—The multiobjective evolutionary algorithm based on decomposition (MOEA/D) has demonstrated superior performance by winning the multiobjective optimization algorithm competition at the CEC 2009. For effective performance of MOEA/D, neighborhood size (NS) parameter has to be tuned. In this letter, an ensemble of different NSs with online self-adaptation is proposed (ENS-MOEA/D) to overcome this shortcoming. Our experimental results on the CEC 2009 competition test instances show that an ensemble of different NSs with online self-adaptation yields superior performance over implementations with only one fixed NS.**

***Index Terms*—Decomposition, multiobjective optimization, self-adaptation.**

I. Introduction

A MULTIOBJECTIVE optimization problem (MOP) can be defined mathematically as follows [1]:

$$\begin{array}{ll}\text{minimize :} & F(x) = (f_1(x), \ldots, f_m(x))^T \\ \text{subject to :} & x \in \Omega\end{array} \tag{1}$$

where Ω is the decision (variable) space, $F : \Omega \to R^m$ consists of m real-valued objective functions, and R^m is called the objective space. In many real-world applications, since the objectives in (1) conflict with one another, no point in Ω can minimize all the objectives at the same time.

Let $u, v \in R^m$, u is stated to *dominate* v if and only if $u_i \leq v_i$ for every $i \in \{1, \ldots, m\}$ and $u_j < v_j$ for at least one index $j \in \{1, \ldots, m\}$. A point $x^* \in \Omega$ is Pareto optimal if there is no any other point $x \in \Omega$ such that $F(x)$ dominates $F(x^*)$. $F(x^*)$ is then called a Pareto optimal (objective) vector. In other words, any improvement in a Pareto optimal point in one objective must lead to deterioration to at least one other objective. The set of all the Pareto optimal objective vectors is the Pareto front (PF) [1].

Many multiobjective evolutionary algorithms (MOEAs) have been developed to find a set of representative Pareto optimal solutions in a single run. Most of them are Pareto dominance based. Guided mainly by dominance-based fitness measures of individual solutions, these algorithms push the whole population toward the PF. NSGA-II, SPEA-II, and PAES [1] have been among the most popular Pareto-dominance-based MOEAs in the past.

Multiobjective evolutionary algorithm based on decomposition (MOEA/D) [3] is a recent MOEA. Using conventional aggregation approaches, MOEA/D decomposes the approximation of the PF into a number of single objective optimization subproblems. The objective of each subproblem is a (linear or nonlinear) weighted aggregation of all the objectives in the MOP under consideration. Neighborhood relations among these subproblems are defined based on the distances among their aggregation weight vectors. Each subproblem is optimized by using information mainly from its neighboring subproblems. The neighborhood size (NS) plays a crucial role in MOEA/D [5]. Arguably, different multiobjective problems need different NSs, and even for a particular problem, using different NSs at different search stages could improve the algorithm performance. When some solutions are trapped in a locally optimal region, a large NS is required to increase diversity for helping these solutions escape from the trapped region. However, if the globally optimal area has been found, a small NS will be favorable for local exploitation.

Ensemble learning has proven to be very efficient and effective for adjusting algorithmic control parameters and operators in an online manner [7]–[9]. In this letter, we propose to use an ensemble of different NSs in MOEA/D and dynamically adjust their selection probabilities based on their previous performances. We compare the resultant algorithm, called ENS-MOEA/D, with MOEA/D proposed in [3] on CEC 2009 MOEA Contest benchmark problems [4]. Our results indicate that ensemble learning improves the performance of MOEA/D significantly.

II. Review on the MOEA/D

There are several variants of MOEA/D. In this letter, we use MOEA/D with dynamical resource allocation [3], which won the CEC2009 multiobjective algorithm contest. To decompose (1), MOEA/D needs N evenly spread weight vectors $\lambda^1, ..., \lambda^N$. Each $\lambda^j = (\lambda^j_1, ..., \lambda^j_m)^T$ satisfies $\sum_{k=1}^{m} \lambda^j_k = 1$ and $\lambda^j_k \geq 0$ for all k and m. Let $z^* = \left(z^*_1, ..., z^*_m\right)^T$, where $z^*_i = \min\{f_i(x)|x \in \Omega\}$. Then, the problem of approximation of the PF of (1) can be decomposed into N scalar optimization subproblems and the objective function of the jth minimization subproblem is

$$g^{te}(x|\lambda, z^*) = \max_{1\leq i\leq m} \left\{\lambda_i |f_i(x) - z^*_i|\right\}. \tag{2}$$

z^* is often unknown before the search, the algorithm uses the lowest f_i-value found during the search to substitute z^*_i[2]. During the search, MOEA/D maintains:

1) a population of N points $x^1, ..., x^N \in \Omega$, where x^i is the current solution to ith subproblem;

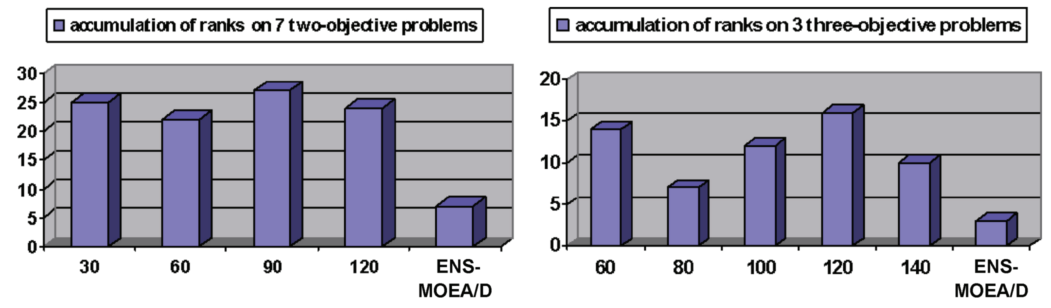
Manuscript received December 24, 2010; revised May 31, 2011; accepted August 4, 2011.

S.-Z. Zhao and P. N. Suganthan are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore (e-mail: zh0032ng@ntu.edu.sg; epnsugan@ntu.edu.sg).

Q. Zhang is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: qzhang@essex.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2011.2166159



AQ:3 Fig. 1. Performance of all the variants of MOEA/D with different fixed NS and ENS-MOEA/D.

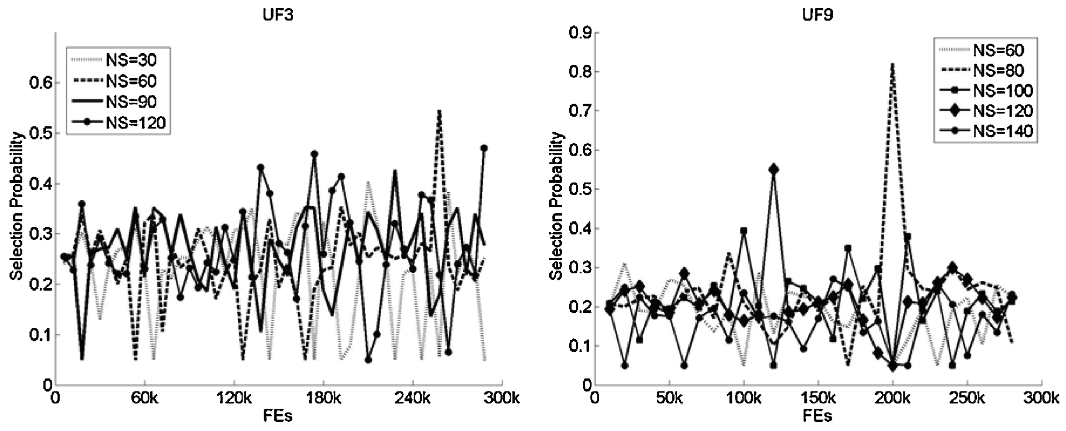


Fig. 2. Selection probabilities of different NSs in ENS-MOEA/D. (a) UF3. (b) UF9.

2) $FV^1, \ldots, FV^N$, where FV^i is the F-value of x^i, i.e., $FV^i = F(x^i)$ for each $i = 1, \ldots, N$;
3) $z = (z_1, \ldots, z_m)^T$, where z_i is the best value found so far for objective f_i.

For each weight vector, its NS-neighborhood is the set of NS-closest weight vectors to it. Correspondingly, each solution and each subproblem have their NS-neighborhoods, respectively.

At each generation, a set of the current solutions is selected. For each selected solution x^i, MOEA/D does the following.

1) Set the mating and update range P to be the T-neighborhood of x^i with a large probability, and the whole population otherwise.
2) Randomly select three current solutions from P.
3) Apply genetic operators on the above selected solutions to generate a new solution y, repair y if necessary. Compute $F(y)$.
4) Replace a small number of solutions in P by y if y is better than them for their subproblems.

No solution will be replaced in Step 4 if y is not better than any solution in P for its subproblem. When such a case happens, we say that the update fails, otherwise, it is successful.

III. Ensemble of NSs for MOEA/D

To evolve the solution of a subproblem, only the current solutions to its neighboring subproblems are exploited in MOEA/D. Usually, a larger neighborhood makes the search more globally, whereas a smaller NS encourages local search. Hence, by appropriately adjusting the NS for each subproblem, the performance of the MOEA/D can be enhanced. However, for diverse problems, a trial-and-error approach can be too demanding for tuning NSs. Motivated by these observations, we employ an ensemble of NSs which are selected according to their historical performances of generating promising solutions.

In ENS-MOEA/D, K-fixed NSs are used as a pool of candidates. During the evolution, a NS will be chosen for each subproblem from the pool based on the candidates' previous performances of generating improved solutions. In ENS-MOEA/D, the certain fixed number of previous generations used to store the success probability is defined as the learning period (LP). At the generation $G > \text{LP} - 1$, the probability of choosing the kth ($k = 1, 2, \ldots, K$)NS is updated by

$$p_{k,G} = \frac{R_{k,G}}{\sum_{k=1}^{K} R_{k,G}} \tag{3}$$

where

$$R_{k,G} = \frac{\sum\limits_{g=G-\text{LP}}^{G-1} FEs\_success_{k,g}}{\sum_{g=G-\text{LP}}^{G-1} FEs_{k,g}} + \varepsilon, (k = 1, 2, .., K; G > \text{LP}). \tag{4}$$

$R_{k,G}$ represents the proportion of improved solutions generated with the kth NS within the previous LP generations. By improved solutions, we mean the solutions which successfully entered the next generation. $FEs_{k,g}$ is the total number of solutions generated with the kth NS within the previous LP generations, $FEs\_\text{success}_{k,g}$ is the number of improved solutions generated with the kth NS within the previous LP generations. The small constant value $\varepsilon = 0.05$ is used to

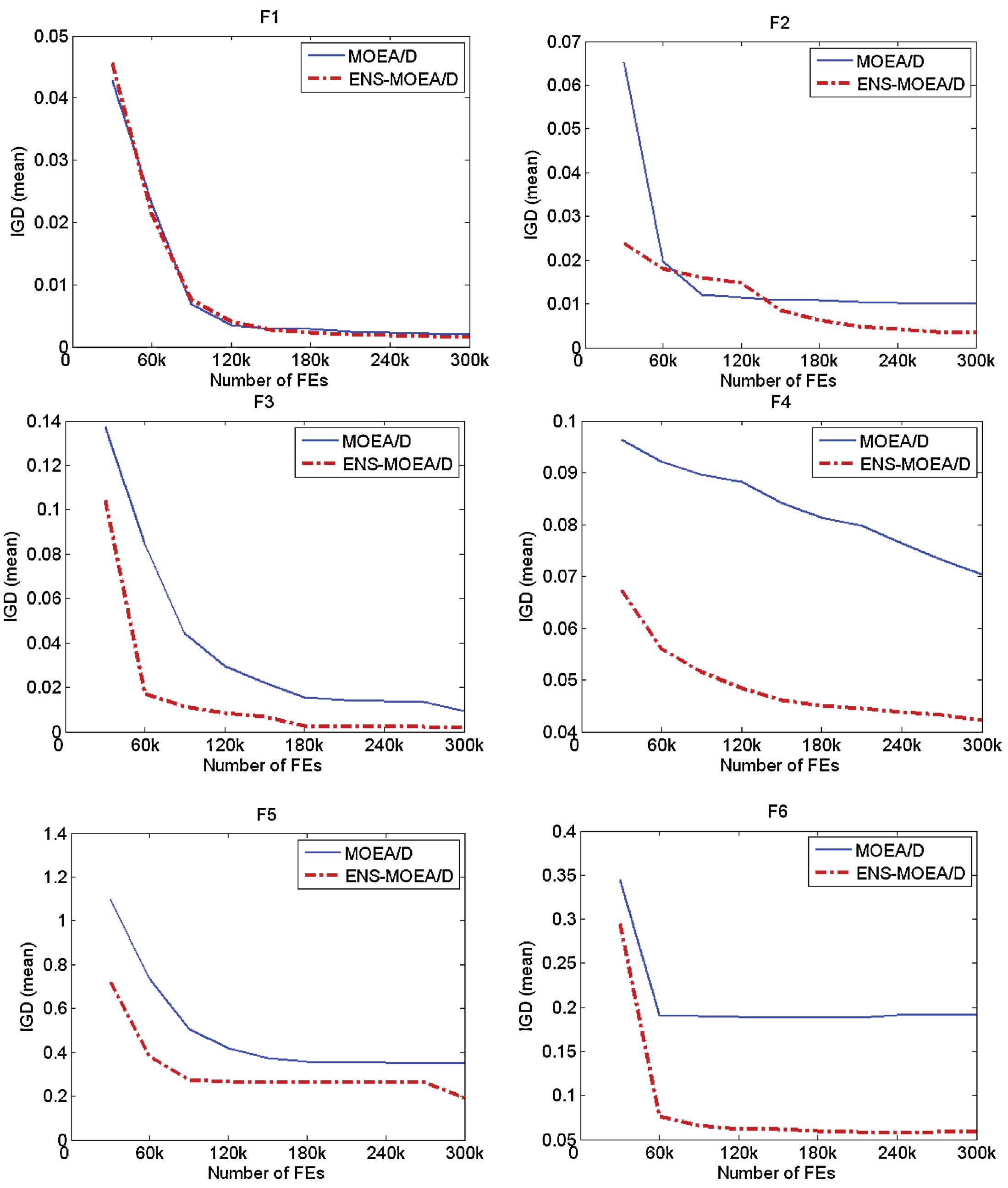


Fig. 3. Convergence graphs in term of mean of IGD obtained by two methods. (a) F1. (b) F2. (c) F3. (d) F4. (e) F5. (f) F6. (g) F7. (h) F8. (i) F9. (j) F10.

avoid the possible zero selection probabilities. To ensure that the probabilities of choosing strategies are always summed to 1, we further divide $R_{k,G}$ by $\sum_{k=1}^{K} R_k$ in calculating $p_{k,G}$.

The procedure of ENS-MOEA/D is presented below.

1) *Initialization.*
   Set generation counter as $G=0$; probability $p_{k,G} = 1/K$. Generate a uniform spread of N weight vectors, $\lambda^i = (\lambda_1, \ldots, \lambda_N)$. Generate an initial population $x^1, \ldots, x^N \in \Omega$ by uniformly randomly sampling from the search space. Initialize $z = (z_1, \ldots, z_m)^T$ by setting z_i to be the best value found so far for objective $f_i(x)$. Initialize the utility of each subproblem π^i.

2) *Optimization Loop.*

```
**If** The stopping criteria is not satisfied, repeat:
    For $k = 1$ to $N$, based on the current probability $p_{k,G}$, select one NS from the NS pool and then define the neighborhood of subproblem $k$.
    Apply the MOEA/D [3] (MOEA/D is shown in Appendix 2) for one generation. $G = G + 1$; AQ:1
    Update $FEs_{k,G}$ and $FEs\_success_{k,G}$ for all the NSs.
        **If** $(mod(G, \mathrm{LP}) == 0)$
        Update the probability $P_{k,G}$ based on the current $FEs_{k,G}$
         and $FEs\_success_{k,G}$ for all the $k$th NS as in (4).
         Reinitialize the $FEs_{k,g}$ and $FEs\_success_{k,g}$ as 0
         for the next LP loop.
        **Endif**
**Else** Stop and output the final $N$ solutions.
**Endif**
```

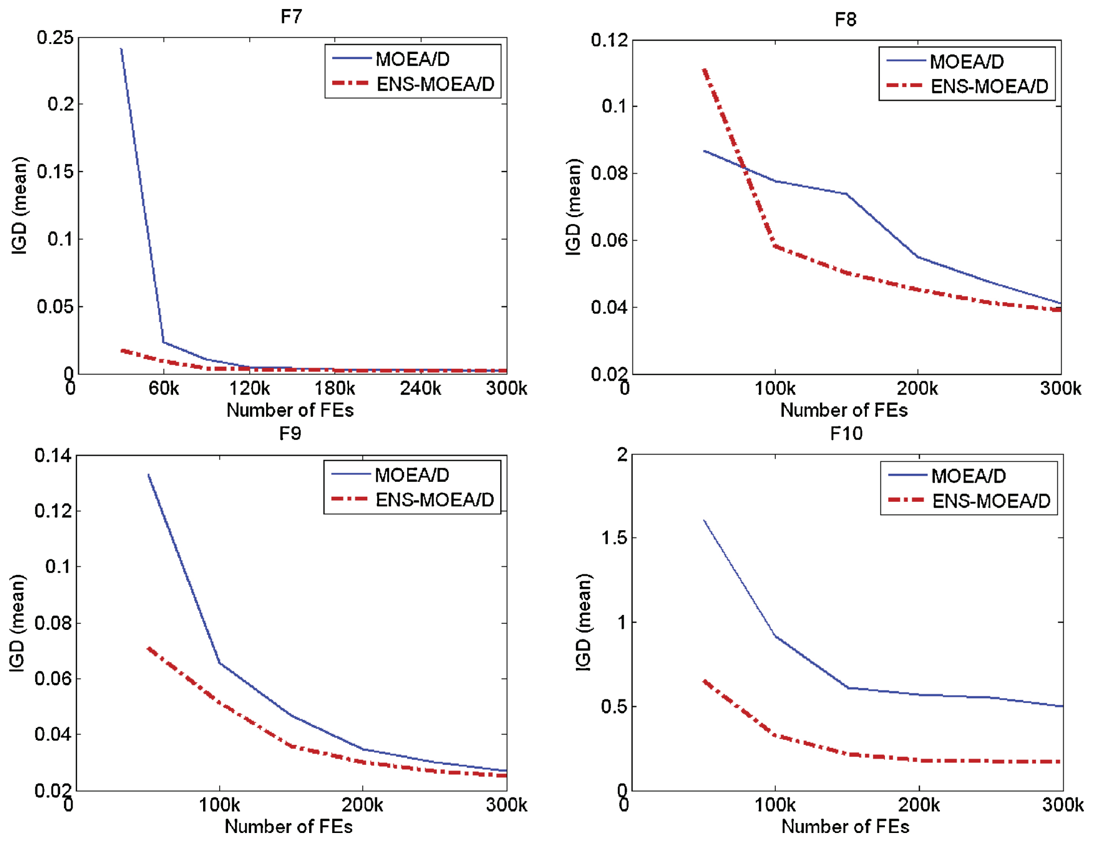


Fig. 3 (*Continued*).

IV. Experimental Results

AQ:2

ENS-MOEA/D is tested on the first ten CEC 2009 Contest unconstrained test instances [4]. The IGD performance measure is used as in the CEC 2009 Contest. The IGD measures both the convergence as well as the diversity among solutions. The common parameter settings are the same as in MOEA/D [3]. N is set to be 600 for two objectives and 1000 for three objectives; CR = 1.0 and F = 0.5 in DE operators; the number of function evaluations (FEs) is 300 000. The four different NSs for the two-objective problems are 30, 60, 90, and 120, where NS = 60 is the original parameter setting in the MOEA/D in [3]; the NSs for the three-objective problems are 60, 80, 100, 120, and 140, where 100 is the original parameter setting for NS in the MOEA/D [3]. Both MOEA/D and ENS-MOEA/D are implemented in MATLAB. Final N population members are used to compute IGD values for both MOEA/D[1] and Ens-MOEA/D in this letter.

We conducted a parameter sensitivity investigation of LP for ENS-MOEA/D using four different values (10, 25, 50, and 75) on the ten benchmark instances. By observing the mean of IGD values over 25 runs, we can conclude that the LP is not so sensitive to most of the instances. In the experiments reported in this letter, LP = 50.

The mean of IGD values over 25 runs among all the variants of MOEA/D with different fixed NSs and ENS-MOEA/D are ranked in Fig. 1. In both of the bar charts in Fig. 1, the different fixed values of NS settings are shown in the horizontal axis, while the vertical axis represents the accumulation of rank values for each variant. Our results also showed that the ENS-MOEA/D yields superior performance (smallest accumulation of ranks) over all other implementations with only one fixed value of NS setting.

To investigate the self-adaptive selection of different NSs in ENS-MOEA/D, the average value of the selection probabilities for each NS as the evolution progresses are plotted in Fig. 2. At the beginning, each NS is assigned the same selection probability. For $UF3$, NS = 30 is rarely used during the whole evolution progress. It is because NS = 30 cannot yield better results during the search. On the contrary, NS = 90 and NS = 120 are often assigned high probabilities, especially at the last search stage. For UF9, it is obvious that NS = 80 dominates others when the FEs are around 200 k. However, at the last stage, NS = 80 has received very small selection probability. Overall, there is not any NS dominates others throughout the whole evolution, the selection probabilities for each NS varies very much. From this observation, one conclusion can be made, the ensemble of different NSs with online self-adaptation is necessary for MOEA/D.

The comparison, in terms of the IGD value, between the original MOEA/D and the proposed ENS-MOEA/D are presented in Fig. 3 and Table I. The t-test at the 5% significance level has been conducted to compare the final IGD values obtained at 300 000 function evaluations of the two methods. h = 1 in Table I means that the difference is significant and h =

[1]MOEA/D results presented at http://cswww.essex.ac.uk/staff/zhang/webofmoead.htm used 100 filtered points and 150 filtered points from final population to compute IGD values for two-objective problems and three-objective problems.

TABLE I
MEANS OF THE FINAL IGD VALUES OBTAINED BY TWO METHODS

f	MOEA/D		ENS-MOEA/D		h
	Mean	Std.	Mean	Std.	
1	0.0020211	0.00029442	**0.0016423**	0.00012564	1
2	0.0047824	0.001809	**0.0040487**	0.0010057	1
3	0.010552	0.0025461	**0.0025916**	0.0004564	1
4	0.062411	0.0022919	**0.04207**	0.0013259	1
5	0.31666	0.067811	**0.24811**	0.042555	1
6	0.1886	0.08487	**0.060847**	0.01984	1
7	0.0017668	0.0009525	**0.0017286**	0.00085227	0
8	0.042793	0.0039138	**0.031006**	0.0030051	1
9	0.029236	0.010788	**0.027874**	0.009573	1
10	0.48577	0.044731	**0.21173**	0.019866	1

0 implies that the t-test cannot detect a significant difference. It is clear from Table I that the quality of the final solutions obtained by ENS-MOEA/D is significantly better than the original MOEA/D on all the instances except f_7. Fig. 3 also indicates that the proposed ENS-MOEA/D converges faster than the original MOEA/D on all the instances except f_7.

V. CONCLUSION

In this letter, an ensemble of different NSs of the subproblems with online self-adaptation was integrated with MOEA/D. We have shown that the online self-adaptation is necessary for improving the algorithm performances since different NS values were needed during the different search stage. Our experimental results on the CEC 2009 unconstrained test instances indicated that our proposed ENS-MOEA/D outperformed the original MOEA/D with different fixed NSs. It has been shown very recently that the use of two different aggregation functions in MOEA/D can be beneficial in solving some many-objective knapsack problems [6]. Our future work will consider employing an ensemble of aggregation functions as well as an ensemble of NSs.

REFERENCES

[1] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*. Chischester, U.K.: Wiley, 2001.

[2] K. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA: Kluwer, 1999.

[3] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. CEC*, 2009, pp. 203–208.

[4] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-487, 2008.

[5] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization," in *Proc. CEC*, 2009, pp. 2508–2515.

[6] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. GECCO*, 2010, pp. 519–526.

[7] E. L. Yu and P. N. Suganthan, "Ensemble of niching algorithms," *Inform. Sci.*, vol. 180, no. 15, pp. 2815–2833, Aug. 2000.

[8] R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Trans. Evol. Computat.*, vol. 14, no. 4, pp. 561–579, Aug. 2010.

[9] S. Z. Zhao and P. N. Suganthan, "Multi-objective evolutionary algorithm with ensemble of external archives," *Int. J. Innovative Comput., Inform. Contr.*, vol. 6, no. 1, pp. 1713–1726, Apr. 2010.