

Adaptive Modeling Strategy for Continuous Multi-Objective Optimization

Aimin Zhou, Qingfu Zhang, Yaochu Jin and Bernhard Sendhoff

Abstract—The Pareto optimal set of a continuous multi-objective optimization problem is a piecewise continuous manifold under some mild conditions. We have recently developed several multi-objective evolutionary algorithms based on this property. However, the modeling methods used in these algorithms are rather costly. In this paper, a cheap and effective modeling strategy is proposed for building the probabilistic models of promising solutions. A new criterion is proposed for measuring the convergence of the algorithm. The locality degree of each local model is adjusted according to the proposed convergence criterion. Experimental results show that the algorithm with the proposed strategy is very promising.

I. INTRODUCTION

In this paper, we consider continuous multi-objective optimization problems (MOPs). Mathematically, a continuous MOP can be defined as follows:

$$\begin{aligned} &\text{minimize} \quad \vec{F}(x) = (f_1(x), \dots, f_m(x))^T \\ &\text{subject to} \quad x \in X \end{aligned} \quad (1)$$

where $X \subset R^n$ is the decision space and $x = (x_1, \dots, x_n)^T \in R^n$ is the decision variable vector. $\vec{F} : X \rightarrow R^m$ consists of m real-valued continuous objective functions $f_i(x)$ ($i = 1, \dots, m$). R^m is the objective space.

Under certain mild conditions, the Pareto set of a continuous MOP is an $(m - 1)$ -dimensional manifold [13]. However, such regularity has rarely been exploited in multi-objective evolutionary algorithms [7]. Very recently, we have proposed several methods based on this regularity property for dealing with continuous MOPs with variable linkages [8], [15], [16]. Local principal component analysis (LPCA) [9] and generative topographic mapping (GTM) [1] are utilized in our methods for modeling the distribution of promising search areas in the decision space. Experimental studies have shown that these methods are able to deal with MOPs with variable linkages. However, LPCA and GTM are rather expensive, particularly, when the MOP has many decision variables. To overcome this shortcoming, a simple yet effective modeling strategy is introduced in this paper. It builds a number of local models in the search space. The locality of each local model is adjusted during the search according to a new proposed convergence criterion, which measures the progress of the algorithm. This strategy provides a natural way for balancing exploitation and exploration.

A. Zhou and Q. Zhang are with Department of Computer Science, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, U.K.

Y. Jin and B. Sendhoff is with Honda Research Institute Europe, Carl-Legien-STD. 30, 63073 Offendens, Germany.

The remainder of the paper is organized as follows. Section II gives the algorithm framework and the adaptive modeling strategy. Section III presents the test instances and performance indicator used in the experiments. In Section IV, we empirically study the different ways for adjusting the locality parameter. Comparison with RM-MEDA [15] and GDE3 [11] has also been carried out in this section. Conclusion is presented in Section V.

II. EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION WITH ADAPTIVE MODELING

A. Algorithm Framework

Notations

- P_t : population at generation t .
- Q_t : offspring set at generation t .
- N : population size.
- N^P : number of points used for building a model.
- N^R : number of reference points, which is set to be $N^R = 2N/N^P$ in our implementation.
- N_t^L : locality parameter at generation t .
- T : maximal number of generations.

AM-MEDA

/ Initialization */*

Step 0 Set $t := 0$. Generate and evaluate an initial population P_0 .

/ Stopping condition */*

Step 1 If stopping condition is met, stop and return the nondominated solutions in P_t .

/ Model Building */*

Step 2 Select N^R reference points $x_1^R, \dots, x_{N^R}^R$ from P_t .

Step 3 Adjust the locality parameter, N_t^L .

Step 4 For each reference point x_k^R , do

Step 4.1 Uniformly randomly select N_t^L candidate points from P_t .

Step 4.2 Build a probability model based statistical information extracted from the N^P closest points to x_k^R in the above selected points.

/ Sampling */*

Step 5 Generate offspring set Q_t from the models built in Step 4 and evaluate Q_t .

/ Selection */*

Step 6 Select N individuals from $Q_t \cup P_t$ to create P_{t+1} .

Step 7 Set $t := t + 1$ and go to **Step 1**.

The details of the framework are given in the following.

B. Initialization

The individual points in the initial population P_0 are uniformly randomly sampled from the search space.

C. Stopping Condition

The algorithm stops after T generations.

D. Selection in Steps 2 and 6

The MaxiMin sorting scheme [14] is used in Steps 2 and 6. This scheme is a modified version of the nondominated sorting scheme used in NSGA-II [3]. The major reason that we choose the MaxiMin is that its selected solutions are more uniformly distributed than those generated by the original nondominated sorting scheme.

E. Update of the Locality Parameter Based on Convergence Degree in Step 3

N_t^L , the locality parameter, plays a crucial rule in AM-MEDA. The smaller N_t^L is, the lower the locality degree of each model built in Step 4 is. In this paper, N_t^L is an integer from $[N_{min}, N_{max}]$ where $N_{min} = N^P$ and $N_{max} = N$. If $N_t^L = N_{min}$, the statistical information extracted in Step 4.2 is from N^P points uniformly randomly selected from P_t , and then the model built in Step 4.2 characterizes the distribution of the whole population. On the other hand, if $N_t^L = N_{max}$, the statistical information in Step 4.2 are from N^P closest points in $P(t)$ to the reference point, then the model characterizes the distribution of promising solutions in the neighborhood of the reference point. Our basic idea is that the global aspect of the current population is more important than its local aspect when the search is far from convergence, and vice versa when the search is close to convergence. Therefore, N_t^L should be adjusted based on convergence of the population.

In this algorithm, the following convergence degree is used:

$$C(P_t) = \begin{cases} +\infty & t < \Delta t \\ \frac{D(P_t, P_{t-\Delta t})}{\Delta t} & t \geq \Delta t \end{cases} \quad (2)$$

where

$$D(P_{t_1}, P_{t_2}) = \frac{1}{\|P_{t_1}\|} \sum_{x \in P_{t_1}} \min_{y \in P_{t_2}} \|\vec{F}(x) - \vec{F}(y)\|_2$$

measures the distance from population P_{t_1} to population P_{t_2} with interval $\Delta t = t_1 - t_2$, and $\Delta t = 10$ in this paper. The smaller $C(P_t)$ is, the more likely that the population is close to convergence.

Four strategies for updating N_t^L are studied in this paper.

1) Strategy 1 (S1)

This strategy works as follows:

- a) $N_0^L = N_{min}$.
- b) For $t = 1, 2, \dots, T$,
 - If $C(P_t) > \varepsilon$, $N_t^L = N_{t-1}^L$.

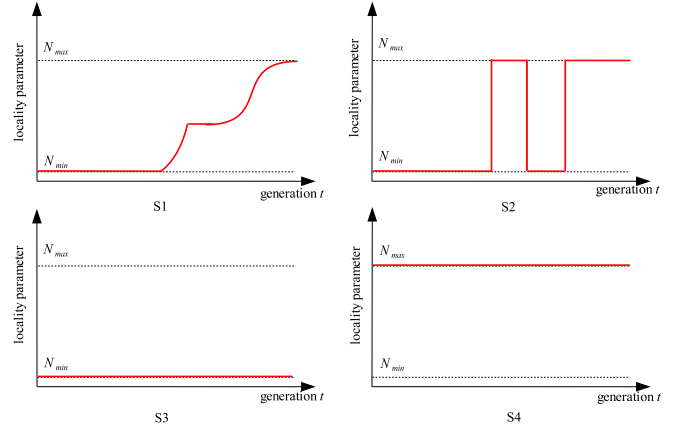


Fig. 1. Illustration of four strategies for updating N_t^L .

- Else if $C(P_t) \leq \varepsilon$ and $C(P_{t-1}) > \varepsilon$,

$$N_t^L = A = N_{t-1}^L \text{ and } \alpha = \frac{1}{4} + \frac{3t}{4T}$$

- Otherwise,

$$N_t^L = A + \lfloor \frac{N_{max} - A}{1 + \exp[-15(t/T + \alpha)]} \rfloor$$

In this strategy, if $C(P_t) < \varepsilon$, N_t^L increases, otherwise, it has no change. The update rule ensures that N_t^L always takes its value from $[N_{min}, N_{max}]$. A is a temporary value and the other constants are empirical values.

2) Strategy 2 (S2)

In this strategy, N_t^L can only takes two values, N_{min} , or N_{max} .

$$N_t^L = \begin{cases} N_{min} & C(P_t) > \varepsilon \\ N_{max} & C(P_t) \leq \varepsilon \end{cases} \quad (3)$$

If $C(P_t) \leq \varepsilon$, each model built in the algorithm is for the local area around its reference point. Otherwise, each model is for the whole population.

3) Strategy 3(S3)

$$N_t^L = N_{min} \quad (4)$$

where $0 < t \leq T$. In this strategy, each model is always for the whole population.

4) Strategy 4 (S4)

$$N_t^L = N_{max} \quad (5)$$

where $0 < t \leq T$. In this strategy, each model is always for the local area around its reference point.

Figure 1 illustrates these four strategies.

F. Model Building

Let S^k contain the N^P closest points to x_k^R in Step 4.2. The mean of S^k is

$$\bar{x}^k = \frac{1}{N^P} \sum_{x \in S^k} x.$$

The covariance matrix of S^k is

$$Cov^k = \frac{1}{N^P - 1} \sum_{y \in S^k} (y - \bar{x}^k)(y - \bar{x}^k)^T.$$

The i th principal component ν_i^k is a unity eigenvector associated with the i th largest eigenvalue λ_i^k of the matrix Cov^k .

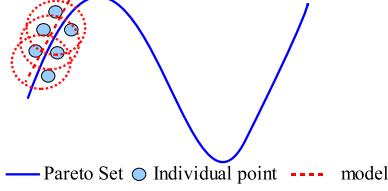


Fig. 2. Illustration of model building.

The following linear model with Gaussian noise, which is called A^k , is used for modeling the distribution of the points in S^k :

$$x = \bar{x}^k + \sum_{i=1}^{m-1} s_i \nu_i^k + \varepsilon^k \quad (6)$$

where s_i is a random variable uniformly distributed in $[s_i^{k,min}, s_i^{k,max}]$,

$$s_i^{k,min} = \min_{x \in S^k} (x - \bar{x}^k)^T \nu_i^k$$

and

$$s_i^{k,max} = \max_{x \in S^k} (x - \bar{x}^k)^T \nu_i^k.$$

$\varepsilon^k \sim N(0, \delta^k I)$, I is an identity matrix and

$$\delta^k = \frac{1}{n - m + 1} \sum_{i=m}^n \lambda_i^k.$$

G. Model Sampling

When sampling new trial solutions, the probability that A^k is selected is:

$$P(A^k) = \frac{evenness(S^k)}{\sum_{i=1}^{N^R} evenness(S^k)} \quad (7)$$

where $evenness(S^k)$ denotes the evenness of the points in S^k and it is calculated as the average density of solutions in S^k ,

$$evenness(S^k) = \frac{1}{N^P} \sum_{x \in S^k} density(x)$$

where $density(x)$ denotes the density of a solution x and it is set as the distance in objective space between x and its second closest point in P_t in our implementation.

In Step 5, offspring are generated as follows:

Reproduction by Sampling

Step 0 Set $Q_t = \phi$.

Step 1 Randomly select a model A^k according to (7).

Step 2 Uniformly randomly generate $s_i \in [s_i^{k,min} - 0.25(s_i^{k,max} - s_i^{k,min}), s_i^{k,max} + 0.25(s_i^{k,max} - s_i^{k,min})]$, $i = 1, \dots, m - 1$.

Step 3 Generate a noise vector $\varepsilon^k \sim N(0, \delta^k I)$.

Step 4 Generate a new trial solution x by (6) and let $Q_t = Q_t \cup \{x\}$.

Step 5 Go to **Step 1** if $|Q_t| < N$.

III. TEST INSTANCES AND PERFORMANCE INDICATORS

A. Test Instances

TABLE I
TEST INSTANCES

	Var.	Obj.
F1	$[-2, 2]^n$	$f_1(x) = 1 - \exp(-\sum_{i=1}^n (x_i - \frac{1}{\sqrt{n}})^2)$ $f_1(x) = 1 - \exp(-\sum_{i=1}^n (x_i + \frac{1}{\sqrt{n}})^2)$
F2	$[0, 1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^n (x_i^2 - x_1)^2)/(n - 1)$
F3	$[0, 1]^n$	$f_1(x) = \sqrt{x_1}$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^n (x_i^2 - x_1)^2)/(n - 1)$
F4	$[0, 1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)} - \frac{x_1}{g(x)} \sin(10\pi x_1)]$ $g(x) = 1 + 9(\sum_{i=2}^n (x_i^2 - x_1)^2)/(n - 1)$
F5	$[0, 1]^n$	$f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9[\sum_{i=2}^n (x_i^2 - x_1)^2/9]^{0.25}$
F6	$[0, 1]^n$	$f_1(x) = \cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)(1 + g(x))$ $f_2(x) = \cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2)(1 + g(x))$ $f_3(x) = \sin(\frac{\pi}{2}x_1)(1 + g(x))$ $g(x) = \sum_{i=3}^n (x_i^2 - x_1)^2$

The test instances in Table I are used to test the proposed algorithm. Among the instances, F1 (FON [2]) has linear linkages among decision variables. F2 – F6 are modified versions [12] of ZDT1, ZDT2, ZDT3, ZDT6 [3], and DTLZ2 [5] respectively, which have nonlinear variable linkages and may cause difficulties for some algorithms [15].

B. Performance Indicator

The inverted general distance is used in assessing the performance of the algorithms.

Let P^* be a set of uniformly distributed points in the objective space along the Pareto front. Let P be an approximation to the Pareto front, The general distance from P^* to P is defined as:

$$D(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}$$

where $d(v, P)$ is the minimum Euclidean distance between v and the points in P . If $|P^*|$ is large enough to represent the Pareto front very well, $D(P^*, P)$ could measure both the diversity and convergence of P in a sense. To have a low value of $D(P^*, P)$, P must be very close to the Pareto front and cannot miss any part of the whole Pareto front.

In our experiments, we select 500 evenly distributed points on Pareto front and let these points to be P^* for each test instance with 2 objectives, and 1,000 points for each test instance with 3 objectives.

Another indicator called difference of hypervolume (I_H^-) [10] is also tried. Since the results are very consistent with those of the D -metric, only the results with D -metric are reported in this paper.

IV. EXPERIMENTAL RESULTS

A. Experimental Settings

In this section, the proposed method, AM-MEDA is compared on $F1 - F6$ with RM-MEDA [15] and GDE3 [11], which is empirically proved to be better than some other algorithms on the given test instances [15]. The parameters are as follows.

- *The dimension of decision variables:* It is set to be 30 for all test problems.
- *The population size (number of new trial solution in each generation):* 100 for 2-objective problems and 200 for 3-objective problems.
- *T:* 100 for $F1, F2, F3$ and $F4$, 500 for $F5$ and 200 for $F6$.
- *The number of runs:* Each algorithm repeats 20 times on each test instance.
- *Parameter setting in AM-MEDA:* N^P is set to be 10 for 2-objective problems, and 20 for 3-objective problems. The threshold of convergence test is 0.001.
- *Parameter setting in RM-MEDA:* In LPCA algorithm, K , the number of clusters, is set to be 5.
- *Parameter setting in GDE3:* Both CR and F in the DE operator is set to be 1, which was the best setting for most test instances in the simulation studies in [4].

B. Comparison of Updating Strategies for Locality Parameter

AM-MEDA with four updating strategies for locality parameter is tested on $F1 - F4$ and the evolution of the average D -metric of the nondominated solutions in the current populations among 20 independent runs is shown in Fig. 3.

Fig. 3 shows that for the three problems, $F2 - F4$ with nonlinear variable linkages, the performance decreases with order of $S1, S2, S3$ and $S4$, while for $F1$ with linear linkage among variables, there is no difference among the performances of $S1, S2$ and $S3$, but they are superior to the performance of $S4$. For $F1$, since its Pareto set is a segment line, any points near/in the Pareto set will be proper to build model (6). Thus $S1, S2$ and $S3$ have similar performance while for $S4$, it might mislead the search at the beginning and its performance is worse than the others.

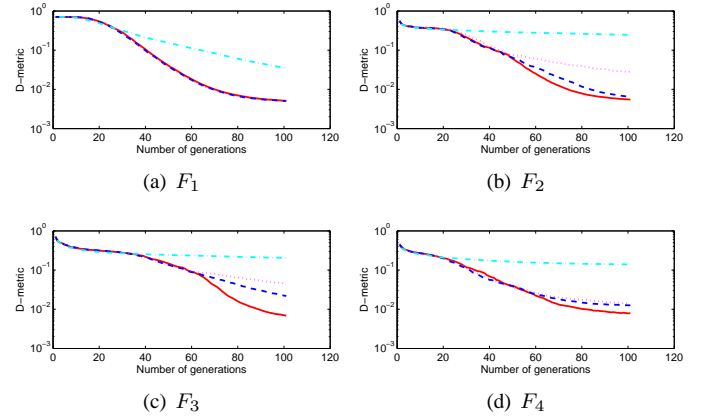


Fig. 3. The evolution of the average D -metric of the nondominated solutions in the current populations among 20 independent runs in four matting strategies. The solid line is with $S1$, dash line with $S2$, dot line with $S3$ and dash-dot line with $S4$.

From these results, it can be concluded that the strategy $S1$ can capture the dynamics of evolution better than the others. And it will be used as the default strategy in the following sections.

C. Comparison with RM-MEDA and GDE3

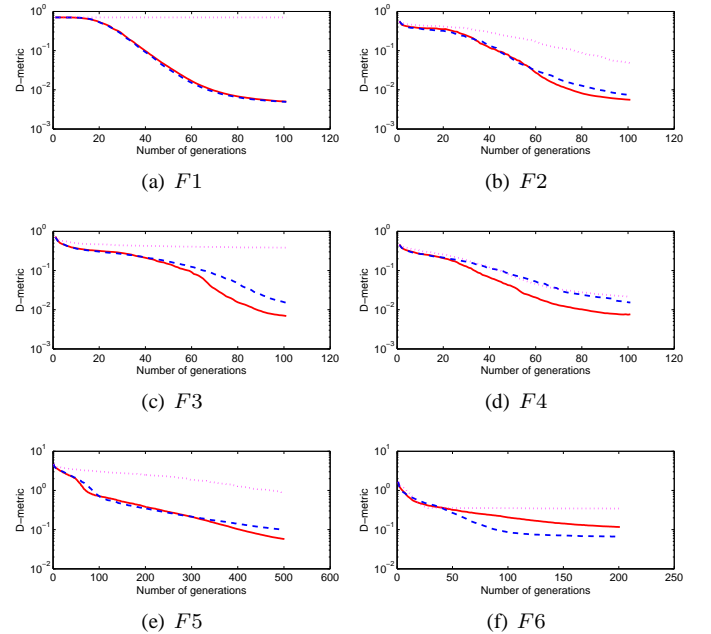


Fig. 4. The evolution of the average D -metric of the nondominated solutions in the current populations among 20 independent runs in three algorithms. The solid line is with AM-MEDA, dash line with RM-MEDA, and dot line with GDE3.

In this section, the proposed AM-MEDA is compared with RM-MEDA and GDE3 on the seven test problems shown in Table I. The average values of D -metric at each generation on $F1 - F6$ are shown in Fig. 4. The Pareto fronts of the

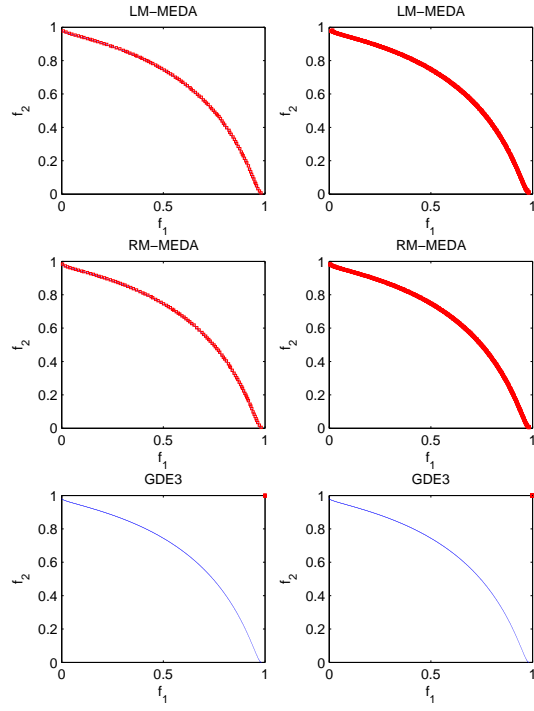


Fig. 5. The final nondominated fronts found by each algorithm on F1. The left panels show the nondominated fronts with the lowest D -metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

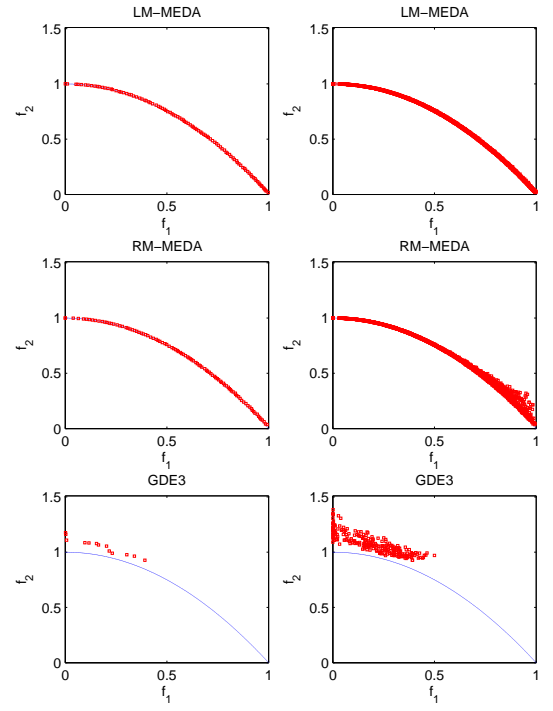


Fig. 7. The final nondominated fronts found by each algorithm on F3. The left panels show the nondominated fronts with the lowest D -metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

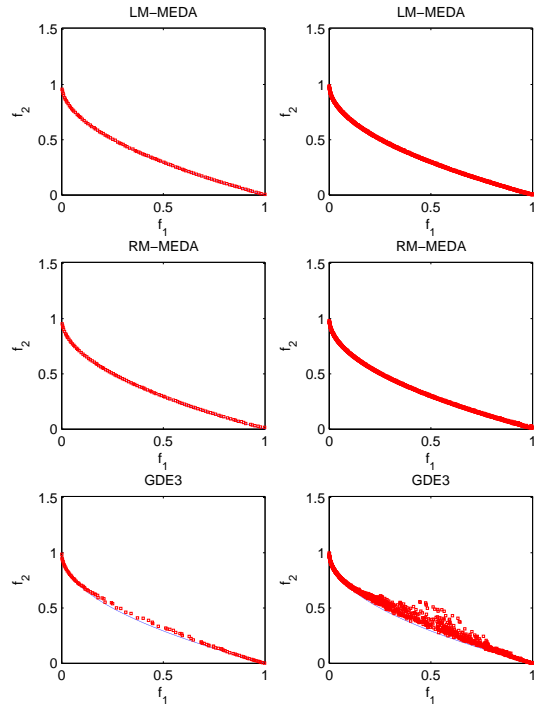


Fig. 6. The final nondominated fronts found by each algorithm on F2. The left panels show the nondominated fronts with the lowest D -metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

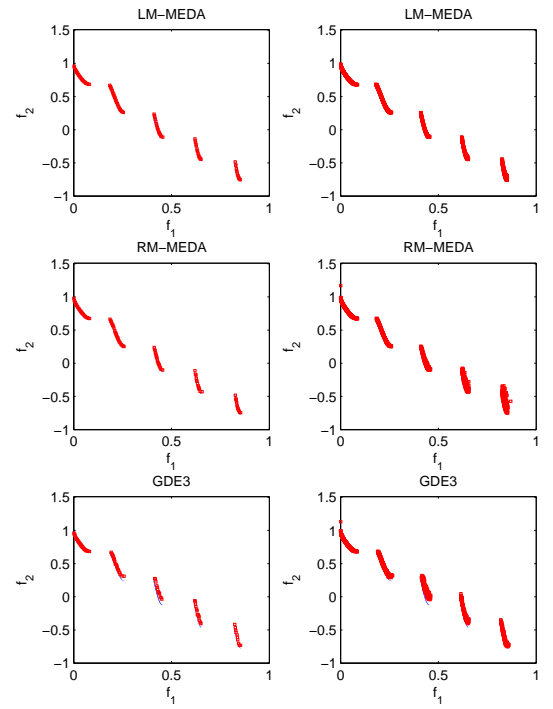


Fig. 8. The final nondominated fronts found by each algorithm on F4. The left panels show the nondominated fronts with the lowest D -metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

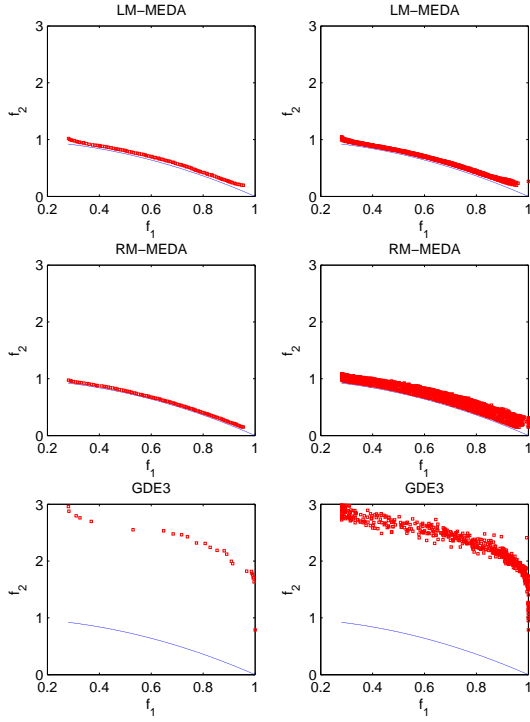


Fig. 9. The final nondominated fronts found by each algorithm on F5. The left panels show the nondominated fronts with the lowest D -metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

obtained nondominated points at the last generation are also drawn in Fig. 5-10 respectively.

- F1** AM-MEDA is comparable with RM-MEDA on $F1$. The D -metric curves are very close to each other from the very beginning. Since the Pareto set of $F1$ is a line, it is very strange that GDE3 can not tackle this problem. The reason may be that the Pareto set only covers a very small part of the search space.
- F2** $F2$ has convex Pareto front and there is nonlinear variable linkages. AM-MEDA performs slightly better than RM-MEDA from Fig. 4(b) and they both can approximate the Pareto Front very well. From Fig. 6, it can be seen that GDE3 can cover the whole Pareto front but with big variance.
- F3** $F3$ has concave Pareto front and there is nonlinear variable linkages. AM-MEDA performs slightly better than RM-MEDA both from Fig. 4(c) and Fig. 7. Fig. 7 shows that the Pareto fronts obtained by RM-MEDA have a little variance near the bottom right end of the Pareto front. For GDE3, both the diversity and convergence of the obtained Pareto fronts are not good.
- F4** $F4$ has disconnected Pareto front and there is nonlinear linkages between variables. Although AM-MEDA shows slightly better results on D -metric in Fig. 4(d), there is not much difference between the obtained Pareto fronts in Fig. 8. The reason might be that AM-MEDA has higher probability to cover the whole Pareto front in

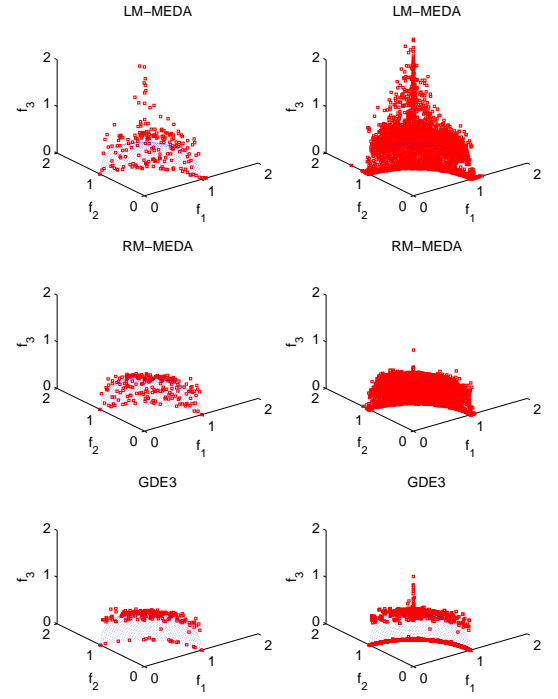


Fig. 10. The final nondominated fronts found by each algorithm on F6. The left panels show the nondominated fronts with the lowest D -metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

a single run.

- F5** $F5$ has nonuniformly distributed Pareto front and this might be the reason why GDE3 can not converge to the Pareto front. The Pareto fronts obtained by AM-MEDA are more close to Pareto front than those of RM-MEDA.
- F6** $F6$ has three objectives and there is nonlinear variable linkages. RM-MEDA is slightly better than AM-MEDA on this problem. Fig. 10 shows that none of the three algorithms can approximate the Pareto front ideally. AM-MEDA and RM-MEDA cover the whole Pareto front, but GDE3 can only cover part. Although AM-MEDA can cover the whole Pareto front, some points are still far away from Pareto front.

Overall, AM-MEDA and RM-MEDA outperform GDE3 on these test problems because they take into account the population distribution and thus use more information in generating new trial solutions. AM-MEDA is slightly better than RM-MEDA on 2-objective problems with nonlinear linkages among decision variables and RM-MEDA is slightly better than AM-MEDA on 3-objective problems with nonlinear variable linkages. For problems with linear linkages, RM-MEDA and AM-MEDA have similar performance.

D. CPU-Time Cost

We also record the average CPU time¹ used by AM-MEDA, RM-MEDA and GDE3 for these test problems. The statistical results are given in Table II.

¹The machine used is Pentium(R) 4 3.00GHz, 1.00GB RAM.

TABLE II
CPU TIME (IN SECONDS) USED BY AM-MEDA, RM-MEDA AND GDE3

Problem	Methods					
	AM-MEDA		RM-MEDA		GDE3	
	mean	std.	mean	std.	mean	std.
F1	2.925	0.044	8.296	0.292	2.649	0.037
F2	2.564	0.049	7.140	0.295	0.333	0.025
F3	2.501	0.029	7.282	0.340	0.378	0.030
F4	2.599	0.040	7.298	0.290	0.302	0.025
F5	12.838	0.249	24.189	2.871	1.628	0.066
F6	10.376	0.211	43.141	2.035	10.919	1.636

Table II indicates that although AM-MEDA needs much more time than GDE3, AM-MEDA can significantly reduce the CPU time comparing to RM-MEDA because for AM-MEDA, there does not exist an inner optimization process as in RM-MEDA.

V. CONCLUSIONS

In this paper, we have proposed a multi-objective estimation of distribution algorithm in which an adaptive modeling technique is used for modeling the distribution of promising solutions. Unlike the adaptive tuning of variance used in [6], in this paper, the locality of models is dynamically adjusted based on a convergence degree which measures how well a population converges. Our basic idea is that the search should focus more on local statistical information if it is close to converge and vice versa. The experimental results have shown that our proposed method performs similarly to RM-MEDA but the its CPU time is much less than that of RM-MEDA. In the future, we plan to refine this method for dealing with complicated MOPs.

REFERENCES

- [1] Christopher M. Bishop, Markus Svensén, and Christopher K. I. Williams. GTM:the generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.
- [2] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, LTD, Baffins Lane, Chichester, 2001.
- [3] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [4] Kalyanmoy Deb, Ankur Sinha, and Saku Kukkonen. Multi-Objective Test Problems, Linkages, and Evolutionary Methodologies. In Maarten Keijzer et al., editor, *2006 Genetic and Evolutionary Computation Conference (GECCO 2006)*, volume 2, pages 1141–1148, Seattle, Washington, USA, July 2006. ACM Press.
- [5] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.
- [6] Jörn Grahl, Peter A.N. Bosman, and Franz Rothlauf. The correlation-triggered adaptive variance scaling idea. In Maarten Keijzer et al., editor, *2006 Genetic and Evolutionary Computation Conference (GECCO 2006)*, pages 397–404, Seattle, Washington, USA, July 2006. ACM Press.
- [7] Yaochu Jin and Bernhard Sendhoff. Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC 2003)*, pages 1910–1917, Canberra, Australia, December 2003. IEEE Press.
- [8] Yaochu Jin, Aimin Zhou, Qingfu Zhang, Bernhard Sendhoff, and Edward Tsang. *Multi-Objective Problem Solving*, chapter Modeling Regularity to Improve Scalability of Model-based Multi-objective Optimization Algorithms. Springer. to appear.
- [9] Nandakishore Kambhatla and Todd K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516, October 1997.
- [10] Joshua D. Knowles, Lothar Thiele, and Eckart Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report 214, Computer Engineering and Networks Laboratory, ETH Zurich, Gloriastrasse 35, 8092 Zurich, Switzerland, 2006.
- [11] Saku Kukkonen and Jouni Lampinen. GDE3: The third evolution step of generalized differential evolution. In *Proceedings of the Congress on Evolutionary Computation (CEC 2005)*, pages 443–450, Edinburgh, U.K, September 2005. IEEE Press.
- [12] Hui Li and Qingfu Zhang. A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. In *Parallel Problem Solving from Nature (PPSN IX)*, volume 4193 of *Lecture Notes in Computer Science*, pages 583–592, Reykjavik, Iceland, 2006. Springer.
- [13] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *Kluwer's International Series in Operations Research & Management Science*. Kluwer Academic Publishers, 1999.
- [14] E J Solteiro Pires, J A Tenreiro Machado, and P B de Moura Oliveira. Multi-objective MaxiMin sorting scheme. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 3410 of *Lecture Notes in Computer Science*, pages 165–175, Guanajuato, Mexico, March 2005. Springer.
- [15] Qingfu Zhang, Aimin Zhou, and Yaochu Jin. RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 2007. in press.
- [16] Aimin Zhou, Qingfu Zhang, Yaochu Jin, Bernhard Sendhoff, and Edward Tsang. Modelling the population distribution in multi-objective optimization by generative topographic mapping. In *Parallel Problem Solving From Nature (PPSN IX)*, volume 4193 of *Lecture Notes in Computer Science*, pages 443–452, Reykjavik, Iceland, September 2006. Springer-Verlag.