

The RM-MEDA Based on Elitist Strategy

Li Mo, Guangming Dai, and JianKai Zhu

Scholl of Computer, China University of Geosciences, Wuhan 430074, China
moliwh@gmail.com

Abstract. The Estimation of Distribution Algorithms(EDAs) is a new paradigm for Evolutionary Computation. This new class of algorithms generalizes Genetic Algorithms(GAs) by replacing the crossover and mutation operators by learning and sampling the probability distribution of the best individuals of the population at each iteration of the algorithm. In this paper, we review the EDAs for the solution of combinatorial optimization problems and optimization in continuous domains. The paper gives a brief overview of the multiobjective problems(MOP) and estimation of distribution algorithms(EDAs). We introduce a representative algorithm called RMMEDA (Regularity Model Based Multi-objective Estimation of Distribution Algorithm). In order to improve the convergence performance of the algorithm, we improve the traditional RM-MEDA. The improvement we make is using part of the parent population with better performance instead of the entire parent population to establish a more accurate manifold model, and the RM-MEDA based on elitist strategy theory is proposed. Experimental results show that the improved RM-MEDA performs better on the convergence metric and the algorithm runtime than the original one.

Keywords: EDA, RMMEDA, Elitist Strategy.

1 Introduction

Many optimization problems, in scientific research and engineering areas, belong to multi-objective optimization problems. Multi-objective problems are different with single objective problems, the optimal solutions of multi-objective problem is a set of solutions. As multi-objective evolutionary algorithms themselves are based on the population and the solutions from them are non-dominated solutions set, they are becoming the most effective approach to solve multi-objective problems. In recent decades, multi-objective evolutionary algorithms became hot spots and achieved many good results in multi-objective area.

Since the publication of Schaffer's seminal work [1], a number of evolutionary algorithms (EDAs) have been developed for multi-objective optimization problems [2]-[5]. The major advantage of these multi-objective evolutionary algorithms (MOEAs) over other methods are that they work with a population of candidate solutions and thus can produce a set of Pareto optimal solutions to approximate the Pareto front and Pareto set in a single run.

Estimation of distribution algorithms (EDAs) are a new computing paradigm in evolutionary computation [6]. There is no crossover or mutation in EDAs. Instead, they explicitly extract globally statistical information from the selected solutions and build a posterior probability distribution model of promising solutions, based on the extracted information. New Solutions are sampled from the model thus built and, fully or in part, replace the old population. Several EDAs have been developed for continuous MOPs [7]-[9]. However, these EDAs do not take the regularity into consideration in building probability models. Since probability modeling techniques under regularity have been widely investigated in the area of statistical learning [10], [11], it is suitable to take advantage of the regularity in the design of EDAs for a continuous MOP. In 2007, Qingfu Zhang etc. developed a regularity model-based multi-objective estimation of distribution (RM-MEDA) [12]. RM-MEDA captures and utilizes the regularity of the Pareto set in the decision space. Systematic experiments have show that, overall RM-MEDA outperforms GDE3 [13], PCX-NSGA-II [14] and MIDEA [8], on a set of test instances with variable linkages.

To obtain more accurate Pareto front, we improved the RM-MEDA. When establishing the manifold model of the population with N solutions, we didn't take the entire one but only half of it, which is a new population with just $N/2$ solutions that perform better, that is they are closer to the real PS manifold. Experiments have been conducted to compare the proposed algorithm with the traditional algorithm on a set of biobjective or triobjective test instances with linear or nonlinear variable linkages. Experiments results show that the improved RM-MEDA performs better than the original RM-MEDA on the convergence metric and the CPU-time costs.

Section 2 introduces continuous multi-objective optimization problems, Pareto optimality. Section 3 shows the ideas of the original RM-MEDA. Section 4 presents the motivation and the details of our improved RM-MEDA. Section 5 presents introduces the convergence and diversity metrics and shows the experiment results.

2 Problems Definition

In this paper, we consider the following continuous multi-objective optimization problem (continuous MOP):

$$\begin{aligned} \min \vec{F}(x) &= (f_1(x), f_2(x), \dots, f_m(x))^T \\ \text{s.t. } x &\in X \end{aligned} \quad (1)$$

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 $f_i(x)$ ($i = 1, 2, \dots, m$). R^m is the objective space.

Let $a = (a_1, \dots, a_m)^T, b = (b_1, \dots, b_m)^T \in R^m$ be two vectors, a is said to dominate b , denoted by $a \prec b$, if $a_i \leq b_i$ for all $i = 1, \dots, m$, and $a \neq b$. A point

$x^* \in X$ is called (globally) Pareto optimal if there is no $x \in X$ such that $\vec{F}(x) \prec \vec{F}(x^*)$. The set of all the Pareto optimal points, denoted by PS , is called the Pareto set. The set of all the Pareto objective vectors, $PF = \{y \in R^m \mid y = \vec{F}(x), x \in PS\}$, is called the Pareto front [2], [15].

3 Traditional RM-MEDA

3.1 Theoretical Foundation

Under certain smoothness assumptions, it can be induced from the Karush-Kuhn-Tucker condition that the PS of a continuous MOP defines a piecewise continuous (m-1)-dimensional manifold in the decision space [16], [17]. Therefore, the PS of a continuous biobjective optimization problem is a piecewise continuous curve in R^n , while the PS of a continuous MOP with three objectives is a piecewise continuous surface.

3.2 Basic Idea

EDAs build a probability model for characterizing promising solutions in the search space based on statistical information extracted from the previous search and then sample new trial solutions from the model thus built.

The population in the decision space in an EA for (1) will hopefully approximate the PS and be uniformly scattered around the PS as the search goes on. Therefore, the algorithm envisage the points in the population as independent observations of a random vector $\xi \in R^n$ whose centroid is the PS of (1). Since the PS is an (m-1)-dimensional piecewise continuous manifold, ξ can be naturally described by

$$\xi = \zeta + \varepsilon \quad (2)$$

where ζ is uniformly distributed over a piecewise continuous (m-1)-dimensional manifold, and ε is an n-dimensional zeromean noise vector.

3.3 Algorithm Framework

At each generation t , the RM-MEDA maintains:

- a population of N solutions (i. e. points in X)

$$Pop(t) = \{x^1, x^2, \dots, x^N\}$$

- their \vec{F} values: $\vec{F}(x^1), \vec{F}(x^2), \dots, \vec{F}(x^N)$

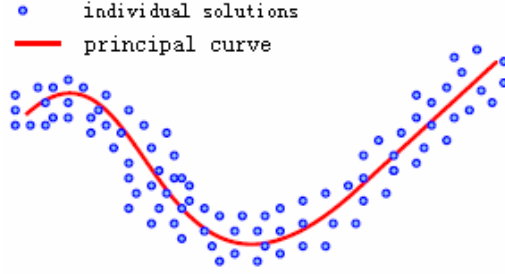


Fig. 1. Illustration of the basic idea. Individual solutions should be scattered around the PS in the decision space in a successful MOEA.

The algorithm works as follows.

RM-MEDA

Step 0) Initialization: Set $t := 0$. Generate an initial population $Pop(0)$ and compute the \vec{F} values of each individual solution in $Pop(0)$.

Step 1) Stopping Condition: If stopping condition is met, stop and return the nondominated solutions in $Pop(t)$ and their corresponding \vec{F} vectors. All these \vec{F} vectors constitute an approximation to the PF .

Step 2) Modeling: Build the probability model (2) for modeling the distribution of the solutions in $Pop(t)$.

Step 3) Reproduction: Generate a new solution set Q from the model (2). Evaluate the \vec{F} value of each solution in Q .

Step 4) Selection: Select N individuals from $Q \cup Pop(t)$ to create $Pop(t+1)$.

Step 5) Set $t := t + 1$ and go to **Step 1**.

4 RM-MEDA Based on Elitist Strategy

4.1 Basic Idea

Traditional RM-MEDA build probability model (2) from the $Pop(t)$ with N solutions. Firstly, we considered that the N solutions in the $Pop(t)$ also have “good” or “bad”, and the “good” ones are the non-dominated solutions which we would like to find in the end. Then we can just don’t use the entire $Pop(t)$ but only the “good” ones or better ones to build more accurate manifold model, which could better characterize promising solutions in the search space, and approximate the real PS more closely! So it can enhance the convergence of the algorithm. Secondly,

when running the process, each generation should establishment model (2), which leads to a high degree of time complexity, then if we reduce the number of the solutions to build probability model (2), it would simply reduce the algorithm's complexity and improve the algorithm speed.

4.2 Algorithm Framework

Based on the basic idea above, we take some improvements during each generation on building the manifold model (2).

The improved algorithm works as follows:

RM-MEDA base on elitist strategy

Step 0) Initialization: Set $t := 0$. Generate an initial population $Pop(0)$ and compute the \vec{F} values of each individual solution in $Pop(0)$.

Step 1) Stopping Condition: If stopping condition is met, stop and return the nondominated solutions in $Pop(t)$ and their corresponding \vec{F} vectors. All these \vec{F} vectors constitute an approximation to the PF .

Setp 2) Selection: Sort the N solutions in $Pop(t)$ based on the nondominated sorting of NSGA-II [14]. Then make a new population $Pop'(t)$ with $N/2$ better solutions in $Pop(t)$.

Step 3) Modeling: Build the probability model (2) for modeling the distribution of the solutions in $Pop'(t)$.

Step 4) Reproduction: Generate a new solution set Q from the model (2). Evaluate the \vec{F} value of each solution in Q .

Step 5) Selection: Select N individuals from $Q \cup Pop(t)$ to create $Pop(t+1)$.

Step 6) Set $t := t + 1$ and go to Step1.

5 Comparison Studies

5.1 Performance Metric

In order to measure the algorithm numerically, we used two metrics γ and Δ to evaluate the algorithm [17].

$$\gamma = \sum_{i=0}^H \min d(f, f_{i, front}) / n \quad (3)$$

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

The first metric γ measures the extent of convergence to a known set of Pareto optimal solutions. The smaller γ is, the better convergence toward the Pareto optimal front. The second metric Δ measures the extent of spread achieved among the obtained solutions. The smaller Δ is, the better diversity of the optimal solutions.

5.2 General Experimental Setting

All algorithms is implemented in C. The machine used in our experiments is Core 2 Duo T6570 (2.1GHz, 2.00GB RAM). In this section, the experimental setting is as follows.

- The number of new trial solutions generated at each generation: The number of new solutions generated at each generation in the two algorithms is set to be 200 for all the test instances with two and three objectives.
- The number of decision variables: It is set to be 30 for all the test instances.
- In Local PCA algorithm, K , the number of clusters, is set to be 5.
- Number of runs and stopping condition: We run each algorithm independently 10 times for each test instance.
- Initialization: Initial populations in all the algorithms are randomly generated.

5.3 Test Instances

The test instances we used is the same as in the traditional RM-MEDA[12]. They are ten instances named ZZJ07_F1-9[17]. ZZJ07_F1-4 are problems with linear PS shapes, and ZZJ05-9 with Quadratic PC shapes.

F1-F4 are variants of ZDT1, ZDT2, ZDT6[17], and DTLZ2[18], respectively. Due to $g(x)$ used in these instances, F1-F3 have the same *PS*. Their *PS* is a line segment

$$x_1 = \dots = x_n, \quad 0 \leq x_i \leq 1, \quad 1 \leq i \leq n.$$

The *PS* of F4 is a 2-D rectangle

$$x_1 = x_3 = \dots = x_n, \quad 0 \leq x_i \leq 1, \quad 1 \leq i \leq n.$$

There are linear variable linkages in these test instances. The variable linkages in these instances [14] are obtained by performing the following linear mapping on the variables in the original ZDT and DTLZ instances:

$$x_1 \rightarrow x_1, \quad x_i \rightarrow x_i - x_1, \quad i = 2, \dots, n.$$

F5-F8 are test instances with nonlinear variable linkages. The *PS* of F5-F8 is a bounded continuous curve defined by

$$x_1 = x_i^2, \quad i = 2, \dots, n. \quad 0 \leq x_1 \leq 1.$$

The *PS* of F8 is a 2-D bounded continuous surface defined by

$$x_1 = x_i^2, \quad i = 3, \dots, n. \quad 0 \leq x_1, x_2 \leq 1.$$

The variable linkages in these instances [19] are obtained by performing the following nonlinear mapping on the variables in the original ZDT and DTLZ instances:

$$x_1 \rightarrow x_1, \quad x_i \rightarrow x_i^2 - x_1, \quad i = 2, \dots, n.$$

F9 are nonlinear variable linkages. Further more, this instance have many local Pareto fronts since their $g(x)$ has many locally minimal points.

5.4 Test Instances with Linear Variable Linkages

We first compare improved RM-MEDA with traditional RM-MEDA on four continuous MOPs with linear variable linkages. The test instances F1-F4 are used for this purpose.

Table 1-4 show the evolution of the average and standard deviation γ and Δ metric of the non-dominated solutions in the current populations among 10 independent runs with the number of function evaluations in four algorithms. It is clear from the following results that improved RM-MEDA performs better than traditional RM-MEDA on convergence metric γ on all these four instances.

We notice that the improved RM-MEDA performs much better on F3. As we know, F3 is the hardest among these four instances. The distribution of the Pareto optimal solutions in the object space in this instance is very different from that in the other three. If we uniformly sample a number of points in the PS of F3 in the decision space, most of the corresponding Pareto optimal vectors in the objective space will be more likely to be in the left apart of the PF. This makes it very difficult for an algorithm to approximate the whole PF. Also traditional RM-MEDA performs better on F3, but our improved RM-MEDA performs much better than it! All this could be attributed to the fact that improved RM-MEDA does extend along the principal directions more closely so that it has a good chance of approximation the whole PF, and the elitist strategy makes the manifold model more closely to the real PS.

Table 1. The statistic results on F1 on 10 runs

Algorithm	Convergence γ	Diversity Δ
traditional RM-MEDA	0.001638 ± 0.000945397	0.5384 ± 0.035587428
improved RM-MEDA	0.001284 ± 0.000120017	0.54896 ± 0.02698283

Table 2. The statistic results on F2 on 10 runs

Algorithm	Convergence γ	Diversity Δ
traditional RM-MEDA	$0.000806 \pm 5.004\text{E-}05$	0.533682 ± 0.036616846
improved RM-MEDA	$0.000788 \pm 5.17301\text{E-}05$	0.524034 ± 0.03505074

Table 3. The statistic results on F3 on 10 runs

Algorithm	<i>Convergence</i> γ	<i>Diversity</i> Δ
traditional RM-MEDA	0.45525 ± 0.293599102	0.716244 ± 0.086163036
improved RM-MEDA	0.338337 ± 0.121726532	0.799542 ± 0.04512084

Table 4. The statistic results on F4 on 10 runs

Algorithm	<i>Convergence</i> γ	<i>Diversity</i> Δ
traditional RM-MEDA	0.128035 ± 0.022854906	0.701646 ± 0.062609226
improved RM-MEDA	0.109137 ± 0.008761657	0.673993 ± 0.039923816

5.5 Test Instances with Nonlinear Variables Linkages

We compare improved RM-MEDA with traditional RM-MEDA on four continuous MOPs with nonlinear variable linkages. F5-F8 are used for this purpose.

Table 5-8 show the results with γ and Δ metric.

Table 5. The statistic results on F5 on 10 runs

Algorithm	<i>Convergence</i> γ	<i>Diversity</i> Δ
traditional RM-MEDA	0.001706 ± 0.000161753	0.552248 ± 0.024664749
improved RM-MEDA	0.001655 ± 0.000140801	0.52892 ± 0.03440603

Table 6. The statistic results on F6 on 10 runs

Algorithm	<i>Convergence</i> γ	<i>Diversity</i> Δ
traditional RM-MEDA	0.001712 ± 0.000157594	0.533772 ± 0.025690472
improved RM-MEDA	$0.00186 \pm 9.95992E-05$	0.551718 ± 0.02569711

Table 7. The statistic results on F7 on 10 runs

Algorithm	<i>Convergence</i> γ	<i>Diversity</i> Δ
traditional RM-MEDA	0.541667 ± 0.046831853	$0.999692 \pm 4.93559E-05$
improved RM-MEDA	0.11812 ± 0.065838184	$0.999806 \pm 5.9867E-05$

Table 8. The statistic results on F8 on 10 runs

Algorithm	Convergence γ	Diversity Δ
traditional RM-MEDA	0.128035 ± 0.022854906	0.701646 ± 0.062609226
improved RM-MEDA	0.109137 ± 0.008761657	0.673993 ± 0.039923816

The experimental results show that improved RM-MEDA outperforms the traditional RM-MEDA on all these instances on convergence γ except for the instance F6. However, the difference is very small. And what encourages us is the result on F7. F7 is the hardest instance for the traditional RM-MEDA due to the fact that the Pareto optimal solutions of F7 are not uniformly distributed in its linkage counterpart. The RM-MEDA based on elitist strategy takes the better half solutions of the whole populations to create the manifold model may be the key factor for this.

5.6 Test Instances with Many Local Pareto Fronts

Here we compare improved RM-MEDA with traditional RM-MEDA on a continuous MOPs with nonlinear variable linkages. Furthermore, this instance has many local Pareto fronts. Table 9 shows the results with γ metric and Δ metric.

Table 9. The statistic results on F9 on 10 runs

Algorithm	Convergence γ	Diversity Δ
traditional RM-MEDA	0.018754 ± 0.005322969	0.588791 ± 0.059738113
improved RM-MEDA	0.021819 ± 0.010345351	0.603867 ± 0.08466081

However, improved RM-MEDA falls back with the traditional RM-MEDA over F9. This is due to the fact that this instance has many local Pareto fronts. Since improved RM-MEDA choice the elite individuals and abandoned the other individuals, this may loses diversity metric of populations to some extent, so the algorithm is more easily trapped into local optimum. This may be the shortcoming of the RM-MEDA based on elitist strategy.

5.7 CPU-Time Cost

The CPU times used by traditional RM-MEDA and improved RM-MEDA with the same experimental settings are given in Table 10.

Table 10. The CPU-Time Cost on one runs

Instance	<i>Traditional RM-MEDA</i> <i>run time(s)</i>	<i>improved RM-MEDA</i> <i>run time(s)</i>
F1	2041.343	1112.938
F2	1890.234	989.25
F3	896.813	331.719
F4	920.172	587.844
F5	1168.172	686.61
F6	1148.079	494.172
F7	880.391	205.391
F8	950.25	165.672
F9	1203.532	450.36

It's clear that traditional RM-MEDA needs extra CPU time compared with the improved RM-MEDA. This is all because that traditional RM-MEDA needs to run Local PCA on a whole population at each generation while improved RM-MEDA just run on half elitist population, then at least half the time can be reduced.

6 Conclusion

In this paper, the traditional RM-MEDA has been improved. Experimental results show that the RM-MEDA based on elitist strategy performs better to the Traditional RM-MEDA on the convergence metric. This is mainly due to the strategy that we use half better solutions of parent population instead of the whole one to create the manifold model, which would produce a more imminent manifold closely to the real PS manifold. Also the algorithm's time complexity will be reduced. However, since we abandoned for half of the population to model, the diversity of the final solutions along the *PS* has a little decline, therefore the decrease is not significantly obvious. On some instances the improved RM-MEDA also performs even better. Overall, the RM-MEDA based on elitist strategy can really enhance its convergence capabilities, and reduce the algorithm's time complexity. In our following studies, we would study how to improve the convergence of the solutions while maintaining the diversity of them. Furthermore, the percentage of the parent population we chose to create the probability model will be studied to get the best performance.

Acknowledgment

This paper is supported by the Natural Science Foundation No.60873107, the National High Technology Research and Development Program No.2008AA12A201, and Natural Science Foundation of Hubei Province No.2008CDB348.

References

- [1] Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: Proc. 1st Int. Conf. Genetic Algorithms, Pittsburgh, PA, pp. 93–100 (1985)
- [2] Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Baffins Lane (2001)

- [3] Coello Coello, C.A., van Veldhuizen, D.A., Lamont, G.B.: *Evolutionary Algorithms for solving Multi-Objective Problems*. Kluwer, Norwell (2002)
- [4] Tan, K.C., Khor, E.F., Lee, T.H.: *Multiobjective Evolutionary Algorithms and Applications*. Springer, Heidelberg (2005)
- [5] Knowles, J., Corne, D.: Memetic algorithms for multiobjective optimization: Issues, methods and prospects. In: *Recent Advances in Memetic Algorithms. Studies in Fuzziness and Soft Computing*, vol. 166, pp. 313–352. Springer, New York (2005)
- [6] Larranaga, P., Lozano, J.A. (eds.): *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Norwell (2001)
- [7] Okabe, T., Jin, Y., Sendhoff, B., Olhofer, M.: Voronoi-based estimation of distribution algorithm for multi-objective optimization. In: *Proc. Congr. Evol. Comput (CEC 2004)*, Portland, OR, pp. 1594–1601 (2004)
- [8] Bosman, P.A.N., Thierens, D.: The naive MIDEA: A baseline multi-objective EA. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) *EMO 2005. LNCS*, vol. 3410, pp. 428–442. Springer, Heidelberg (2005)
- [9] Pelikan, M., Sastry, K., Goldberg, D.: Multiobjective HBOA, clustering, and scalability. *Illinois Genetic Algorithms Laboratory (IlliGAL)*, Tech. Rep. 2005005 (2005)
- [10] Cherkassky, V., Mulier, F.: *Learning from Data: Concepts. Theory. and Methods*. Wiley, New York (1998)
- [11] Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, Berlin (2001)
- [12] Zhang, Q., Zhou, A., Jin, Y.: RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm. *IEEE Transactions on Evolutionary Computation* 12(1), 41–63 (2008)
- [13] Kukkonen, S., Lampinen, J.: GDE3: The third evolution step of generalized differential evolution. In: *Proc. Congr. Evol. Comput (CEC 2005)*, Edinburgh, U.K., pp. 443–450 (2005)
- [14] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 182–197 (2002)
- [15] Miettinen, K.: *Nonlinear Multiobjective Optimization*. Kluwer's International Series in Operations Research & Management Science, vol. 12. Kluwer, Norwell (1999)
- [16] Schutze, O., Mostaghim, S., Dellnitz, M., Teich, J.: Covering Pareto sets by multilevel evolutionary subdivision techniques. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) *EMO 2003. LNCS*, vol. 2632, pp. 118–132. Springer, Heidelberg (2003)
- [17] Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans. Evol. Comput.* 7, 204–223 (2003)
- [18] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: *Evolutionary Multiobjective Optimization, Theoretical Advances and Applications*, pp. 105–145. Springer, New York (2005)
- [19] Li, H., Zhang, Q.: A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) *PPSN 2006. LNCS*, vol. 4193, pp. 583–592. Springer, Heidelberg (2006)