Global Multiobjective Optimization via Estimation of Distribution Algorithm with Biased Initialization and Crossover

Aiming Zhou
Department of Computer Science
University of Essex
Colchester, CO4 3SQ, U. K.
azhou@essex.ac.uk

Qingfu Zhang
Department of Computer Science
University of Essex
Colchester, CO4 3SQ, U. K.
qzhang@essex.ac.uk

Yaochu Jin
Honda Research Institute Europe
Carl-Legien-Str.30,63073
Offenbach,Germany
yaochu.jin@honda-ri.de

Bernhard Sendhoff
Honda Research Institute Europe
Carl-Legien-Str.30,63073
Offenbach,Germany
bernhard.sendhoff@honda-ri.de

Edward Tsang
Department of Computer Science
University of Essex
Colchester, CO4 3SQ, U. K.
edward@essex.ac.uk

ABSTRACT

Multiobjective optimization problems with many local Pareto fronts are a big challenge to evolutionary algorithms. In this paper, two operators, biased initialization and biased crossover, are proposed to improve the global search ability of RM-MEDA, a recently proposed multiobjective estimation of distribution algorithm. Biased initialization inserts several globally Pareto optimal solutions into the initial population; biased crossover combines the location information of some best solutions and globally statistical information in the current population. Experiments have been conducted to study the effects of these two operators.

Categories and Subject Descriptors

I.2.8 [**Artificial intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithm

Keywords

estimation of distribution algorithm, global optimization, multiobjective optimization, biased initialization, biased crossover

1. INTRODUCTION

Global optimization by evolutionary algorithms (EAs) has been widely studied for scalar objective optimization problems [**?**, **?**, **?**, **?**]. However, there is not much effort on it for multiobjective optimization problems (MOPs) [**?**, **?**]. Strategies for global scalar objective optimization may not be steadily extended to multiobjective optimization evolutionary algorithms (MOEAs).

Estimation of distribution algorithms (EDAs) are a new computing paradigm in evolutionary computation [**?**]. 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. The Pareto set (PS) of a continuous MOP is a piecewise continuous $(m-1)$-D manifold. This property has been used in several mathematical programming methods. However, such regularity has not yet been exploited by evolutionary algorithms. Recently, we proposed RM-MEDA [**?**, **?**, **?**, **?**], a regularity model based EDA for continuous MOPs. Experimental results have shown that RM-MEDA can effectively deal with variable linkages. However, if a MOP has many local Pareto fronts (PFs), RM-MEDA could fail in locating the global PF.

This paper introduces two new operators, i.e., biased initialization and biased crossover, for improving the global search ability of RM-MEDA. These two operators aim at guiding the search toward the global PS. Biased initialization inserts several globally Pareto optimal solutions into the initial population; biased crossover combines the location information of some best solutions and globally statistical information in the current population.

The rest of the paper is organized as follows. Section 2 gives some notations and definitions. In Section 3, RM-MEDA is briefly described. The biased initialization and biased crossover are introduced in Section 4. In Section 5, the experimental results are presented. And the paper is

concluded in Section 6.

2. NOTATIONS AND DEFINITIONS

We consider the following continuous MOP:

$$\min F(X) = (f_1(X), \cdots, f_m(X))^T \quad (1)$$

where $X \in \Omega \subseteq R^n$, Ω is a continuous search space, and each objective $f_i : \Omega \to R$ is continuous of decision variable X. Very often, the objectives in a MOP conflict with each other, no single solution can optimize all the objectives at the same time. Pareto optimality is used for defining the best trade-off solutions of a MOP.

A vector $u \in R^m$ dominates another vector $v \in R^m$, denoted as $u \prec v$, iff $u_i \leq v_i$ for all $i = 1, \cdots, m$ and $u_j < v_j$ for at least one $j \in \{1, \cdots, m\}$. The following two definitions are based on this Pareto domination.

Definition 1 (Local Pareto Set): For a given $\varepsilon > 0$, a local Pareto set of (1) is a set of solutions $X \in \Omega$ which can not be dominated by other solution $Y \in \Omega$ which satisfies $||X - Y|| < \varepsilon$. Mathematically, it can be denoted as $LPS = \{X | X \in \Omega, \nexists Y \in \Omega, ||X - Y|| < \varepsilon, F(Y) \prec F(X)\}$.

Definition 2 (Global Pareto Set): Global Pareto set of (1) contains all optimal solutions which can not be dominated by any solutions in the search space and it can be denoted as $GPS = \{X | X \in \Omega, \nexists Y \in \Omega, F(Y) \prec F(X)\}$.

The image of local Pareto set and global Pareto set in objective space are called local Pareto front and global Pareto front, denoted as LPF and GPF respectively.

MOEAs for global optimization aim to find an approximation of the GPS and GPF of (1).

3. THE FRAMEWORK OF RM-MEDA

Under some mild conditions, the PS of (1) defines a $(m-1)$-dimensional manifold where m is the number of objectives.

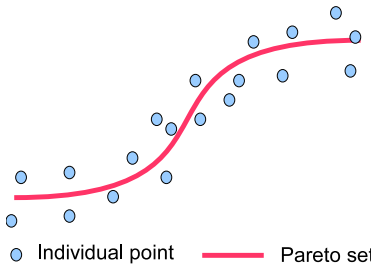

Figure 1: Illustration of individual solutions scattered around the PS in the decision space.

As shown in Figure 1, the population in the decision space in an ideal MOEA for (1) will hopefully approximate the PS and be uniformly scattered around the PS as the search goes on. Therefore, we can envisage the points in the population as independent observations of a random vector $\xi \in R^n$ whose centroid is the PS of (1) and can be naturally described by:

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

where ζ is uniformly distributed over a piecewise continuous $(m-1)$-dimensional manifold. ε is an n-dimensional zero-mean noise vector and n is the number of decision variables.

In RM-MEDA, piecewise $(m-1)$-dimensional linear models are used to approximate model ζ in (2). Local principal component analysis [**?**] is applied to partition a population. In each cluster, the parameters of linear model and noise, ε in (2), are estimated by principal component analysis. New trial solutions are than sampled from model (2).

Let P_t denote a population at generation t, P_t^O denote the offspring generated at generation t and P_t^N denote all the nondominated solutions in P_t. The size of both P_t and P_t^O is fixed to be N. The algorithm works as follows:

RM-MEDA

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

Step 1: If stopping condition is met, stop and return P_t^N which constitutes an approximation to the PF (PS).

Step 2: Build the probability model (2) for modelling the distribution of the solutions in P_t.

Step 3: Sample a new solution set P_t^O from the model (2) and evaluate P_t^O.

Step 4: Select N individuals from $P_t^O \bigcup P_t$ to create P_{t+1}.

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

In the framework of RM-MEDA, the population is initialized randomly in the search space in **Step 0** and a modified version of selection based on the nondominated sorting and crowding distance of NSGA-II [**?**] is used in **Step 4**. The algorithm will stop according to a predefined maximal function evaluations. More details of RM-MEDA can be found in [**?**].

4. BIASED OPERATORS

4.1 Biased Initialization

It is not a new idea to add some 'good' points in initial population to improve the performance of MOEAs. This strategy makes MOEAs like two stage search methods: in stage I, efforts are spent on finding solutions near/in PF while in stage II, the whole PF is generated. In [**?**, **?**], by optimizing a few of aggregation functions with deterministic gradient based optimization methods, a few points, called supporting solutions, are put into an initial population to improve the performance of MOEAs. In [**?**], a two-phase local search is designed for bi-objective traveling salesman problems. In stage I, an initial solution is generated by optimizing only one single objective, and then in stage II, the whole PF is generated by optimizing a sequence of scalar objective problems based on aggregations of the objectives. In [**?**], the algorithm focuses on finding one solution near PF by a method similar to $(1+1)$-ES in stage I and in stage II, a steady state EA is used to spread the individuals along the PF.

For global multiobjective optimization, 'good' points near/in PF might (a) prevent population from trapping into local PF because these 'good' points will dominate some points in local PF, and (b) guide population to global population if they are used in generating offspring.

In this paper, only m (m is the number of objective) points are generated by an EA for global optimization.

There are many ways to convert a MOP into a single objective optimization problem [**?**]. In this paper, the first m initial solutions are generated as follows:

$$X_i = \text{argmin}_{X \in \Omega} \sum_{j=1}^{m} \alpha_j^i f_j(X)$$

where $i = 1, \cdots, m$, α_j^i are randomly chosen weights which satisfy $0 < \alpha_j^i < 1$ and $\sum_{j=1}^{m} \alpha_j^i = 1$.

All the other initial solutions are randomly sampled from the decision space:

$$X_i = \text{rand}(\Omega)$$

where $i = m+1, \cdots, N$ and N is the population size. rand(Ω) returns a uniformly random point in Ω.

In this initialization, m initial solutions will hopefully be near/in the global PF (PS).

4.2 Biased Crossover

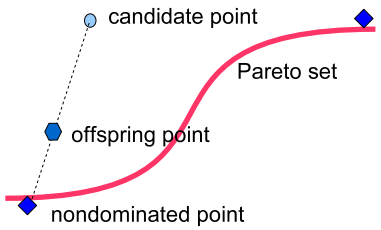

Figure 2: Illustration of biased crossover.

One of the major shortcoming of an EDA is that modelling may ignore isolated high-quality individuals. To overcome this shortcoming, guided mutation has been proposed by combining location information of individuals and global statical information [**?**]. Biased crossover uses the same idea and tries to keep the influence of the best solutions particularly when they are few.

Let P_t^E be a set of solutions generated from P_t by the EDA operator. The biased crossover generates a set of new solutions, P_t^O, in the following way:

Biased Crossover

Step 1: Select the non-dominated set P_t^N from P_t, and set P_t^O empty.

Step 2: If $\frac{|P_t^N|}{|P_t|} > \theta$, set $P_t^O = P_t^E$ and stop, else go to **Step 3**.

Step 3: For each point $X^E \in P_t^E$, randomly select a point $X^N \in P_t^N$, generate a new point

$$X = X^E + \beta(X^N - X^E)$$

where $\beta \in [0, 1]$ is a random number, and put it into offspring set $P_t^O = P_t^O \cup \{X\}$.

In our experiments, the threshold is fixed to be 0.2. In the above operator, when the size of P_t^N is small compared with that of P_t (i.e., $\frac{|P_t^N|}{|P_t|} \leq \theta$), all the new solutions are recombined with solutions in P_t^N. On the other hand, if nondominated solutions have a large fraction in the population, the population distribution model will represent these individuals and thus there is no need to emphasize them again in biased crossover.

In **Step 3**, a new solution is generated between a reference nondominated point and a candidate point which is illustrated in Figure 2.

4.3 Enhanced RM-MEDA for Global Optimization

The above two biased operators can be incorporated into RM-MEDA for global optimization and the resultant method works as follows:

Enhanced RM-MEDA for Global Optimization

Step 0: Set $t := 0$. Generate an initial population P_0 by biased initialization.

Step 1: If stopping condition is met, stop and return P_t^N which constitutes an approximation to the PF (PS).

Step 2: Build the probability model (2) for modelling the distribution of the solutions in $_t$.

Step 3: Generate a candidate solution set P_t^E from the model (2).

Step 4: Generate an offspring set P_t^O from P_t and P_t^E via biased crossover and and evaluate P_t^O.

Step 5: Select N individuals from $P_t^O \bigcup P_t$ to create P_{t+1}.

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

The only differences between the original of RM-MEDA and the above method in Section 3 are in **Step 0** and **Step 4** where biased initialization and biased crossover are used respectively.

5. EXPERIMENTAL RESULTS

We have conducted experimental studies on several test instances. In the following, we report our results on the following modified ZDT4 [**?**] and DTLZ3 [**?**]:

$$ZDT4 \quad \begin{array}{l} f_1(X) = x_1 \\ f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}] \end{array}$$

where $X \in [0, 1] \times [0, 10]^9$ and g used in the experiments are

$$g_1(x) = \frac{1}{4000} \sum_{i=2}^{10} (x_i^2 - x_1)^2 - \prod_{i=2}^{10} cos(\frac{x_i^2 - x_1}{\sqrt{i-1}}) + 2$$

and

$$g_2(x) = 91 + \sum_{i=2}^{10} [(x_i^2 - x_1)^2 - 10cos(2\pi(x_i^2 - x_1))].$$

The instances of ZDT4 with g_1 and g_2 are denoted as F_1 and F_2, respectively.

$$DTLZ3 \begin{cases} f_1(X) = (1 + g(X))cos(x_1\pi/2)cos(x_2\pi/2) \\ f_2(X) = (1 + g(X))cos(x_1\pi/2)sin(x_2\pi/2) \\ f_3(X) = (1 + g(X))sin(x_1\pi/2) \end{cases}$$

Table 1: Mean and Std. of D and Υ achieved on F_1 and F_2.

	F_1		F_2	
	D	Υ	D	Υ
A1	0.0193(0.0114)	0.0182(0.0119)	3.0120(2.7929)	2.7581(2.9146)
A2	0.0383(0.0195)	0.0372(0.0196)	132.3238(126.8185)	133.2056(127.8889)
A3	0.0261(0.0177)	0.0246(0.0174)	159.7894(177.7885)	161.5801(180.1942)
A4	**0.0043**(0.0001)	**0.0020**(0.0001)	**0.1118**(0.2376)	**0.0431**(0.1902)

where $X \in [0,1]^2 \times [0,10]^8$ and g used in the experiments are

$$g_3(x) = \frac{1}{4000}\sum_{i=3}^{10}(x_i^2 - x_1)^2 - \prod_{i=3}^{10} cos(\frac{x_i^2 - x_1}{\sqrt{i-2}}) + 1$$

and

$$g_4(x) = 81 + \sum_{i=3}^{10}[(x_i^2 - x_1)^2 - 10cos(2\pi(x_i^2 - x_1))].$$

The instances of DTLZ3 with g_3 and g_4 are denoted as F_3 and F_4, respectively.

D-metric [**?**] and Υ-metric [**?**] are used here to measure the performance. Let P^* be a set of uniformly distributed points in the objective space along the PF, and let P be an approximation to the PF. D-metric and Υ-metric are defined as:

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

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

where $d(a, A)$ is the minimum Euclidean distance between a and the points in A. If $|P^*|$ is large enough to represent the PF very well, $D(P^*, P)$ could measure both the diversity and convergence of P in a sense when P is close to P^*, while $\Upsilon(P, P^*)$ only measures the convergence of P.

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

In the following, RM-MEDA, RM-MEDA with biased initialization, RM-MEDA with biased crossover, and RM-MEDA with both biased operators are denoted as $A1$, $A2$, $A3$, and $A4$, respectively.

The parameters are as follows: for bi-objective problems F_1 and F_2, the population size is 100; maximal function evaluation is 40,000 for (among which 20,000 is used by EDA/L [**?**] in biased initialization if necessary); the weights used in biased initialization are fixed to $\alpha = (0.9, 0.1)$ and $\alpha = (0.1, 0.9)$. For tri-objective problems F_3 and F_4, the population size is 200; maximal function evaluation is 70,000 for (among which 30,000 is used by EDA/L in biased initialization if necessary); the weights used in biased initialization are fixed to $\alpha = (0.8, 0.1, 0.1)$, $\alpha = (0.1, 0.8, 0.1)$ and $\alpha = (0.1, 0.1, 0.8)$. In all executions, the cluster number in RM-MEDA is 5. The results are based on 100 independent runs.

5.1 Results for modified ZDT4 problems

The mean and standard deviation of the two metrics are shown in Table 1. The PFs obtained by the four algorithms in the final generation are shown in Figure 3.

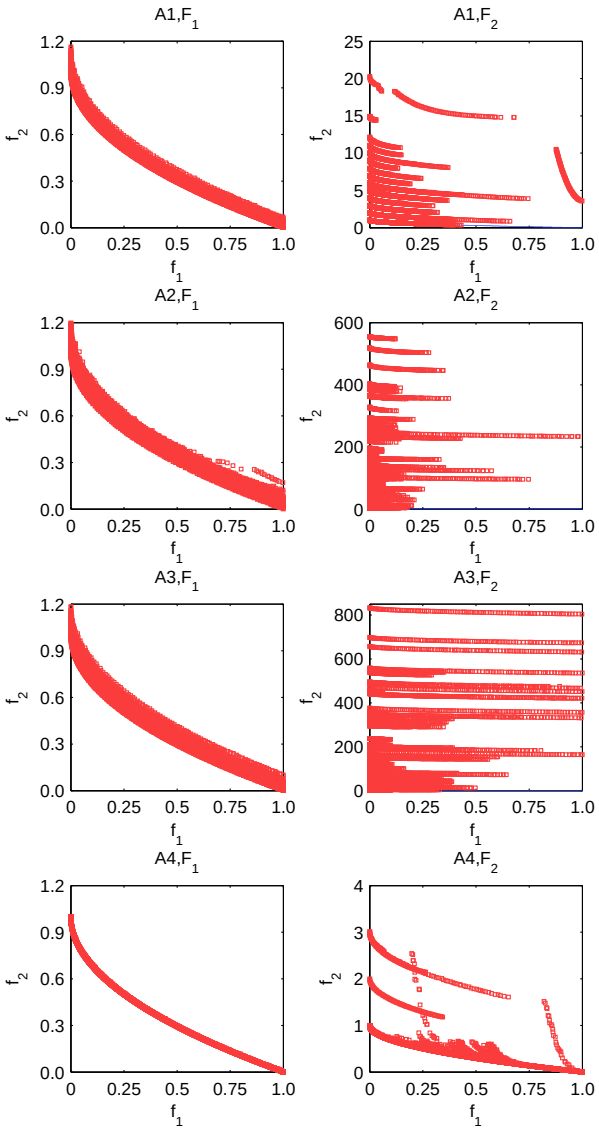

Figure 3: Pareto fronts obtained by the four algorithms on F_1 and F_2.

Table 2: Mean and Std. of D and Υ achieved on F_3 and F_4.

	F_3		F_4	
	D	Υ	D	Υ
A1	0.0617(0.0044)	0.0514(0.0118)	2.9890(4.8322)	10454.9999(17126.2022)
A2	0.0629(0.0053)	0.0531(0.0120)	0.5949(0.1566)	6505.2868(14582.7179)
A3	**0.0608**(0.0042)	**0.0489**(0.0113)	0.9853(2.0288)	2813.9000(9869.7266)
A4	0.0612(0.0046)	0.0508(0.0132)	**0.4990**(0.0400)	**0.0410**(0.0955)

The results in Table 1 show that if only the biased initialization or biased crossover is used, the performance of RM-MEDA will become poorer on both test instances. The reason is that for $A2$, although some good solutions are put into the initial population, they don't play any role in the EDA operator; and for $A3$, the biased crossover might mislead the population into local PFs and it is why A3 failed in F_2, as shown in Figure 3. By using both biased initialization and biased crossover, the performance of RM-MEDA has been significantly improved. It can also be seen from Figure. 3, that the final PFs of $A4$ are closer to the global PFs than those of $A1$.

5.2 Results for modified DTLZ3 problems

The mean and standard deviation of the two metrics are shown in Table 2. The PFs obtained by the four algorithms in the final generation are shown in Figure 4.

The results show that A3 performs slightly better than A4 on F3.

For F_4, it is clear that only A4 can converge to the global PF in most of runs. The Υ-metric values indicate that Pareto fronts achieved by the other three algorithms are still far away from global Pareto front.

6. CONCLUSIONS

In this paper, a biased initialization and a biased crossover have been introduced to improve the global search ability of RM-MEDA. In biased initialization, by solving several scalar objective optimization problems converted from a multiobjective optimization problem, some 'good' points are generated near/in global Pareto front. These 'good' points will then prevent the population from trapping into local Pareto fronts and guide the population into global Pareto front by biased crossover.

The proposed strategy is tested on bi-objective and tri-objective problems and the results show that the global search ability of RM-MEDA is improved remarkably compared to RM-MEDA and RM-MEDA with only one of the biased operators.

To achieve an optimal approximation of global Pareto front, the costs used in initialization and in the main evolving process should be balanced. In this paper, the costs used in two stages are fixed. A more practical way should allocate the cost in these two phases adaptively. And this will be our future work.

7. REFERENCES

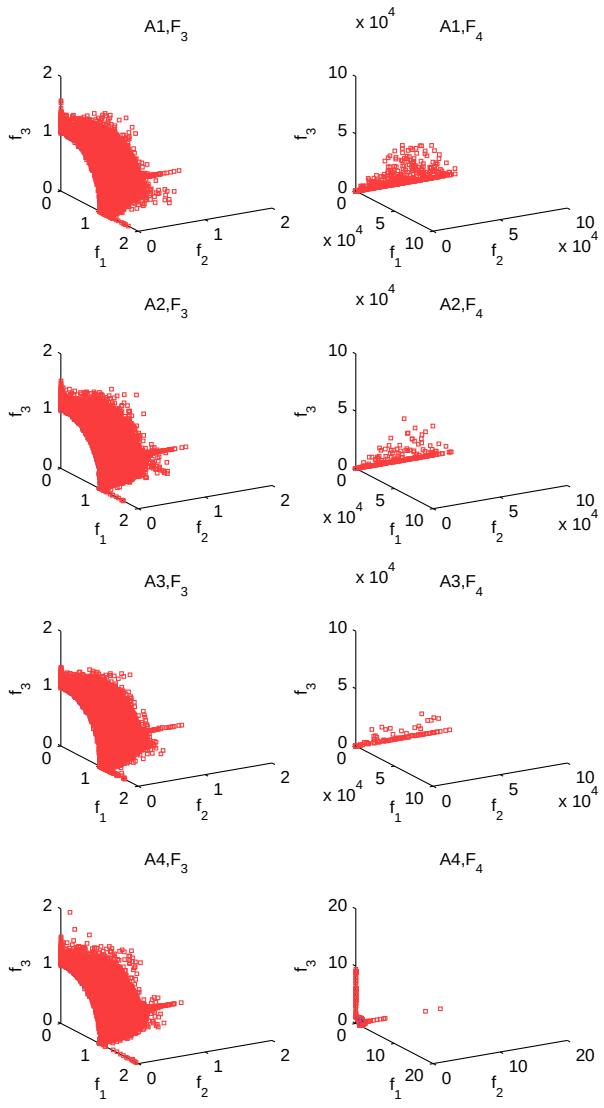

Figure 4: Pareto fronts obtained by the four algorithms on F_3 and F_4.