

Modelling the Population Distribution in Multi-objective Optimization by Generative Topographic Mapping

Aimin Zhou¹, Qingfu Zhang¹, Yaochu Jin², Bernhard Sendhoff², and Edward Tsang¹

¹ Department of Computer Science, University of Essex, Colchester, CO4 3SQ, U.K.

² Honda Research Institute Europe, Carl-Legien-Str. 30, 63073, Offenbach, Germany

Abstract. Under mild conditions, the Pareto set of a continuous multi-objective optimization problem exhibits certain regularity. We have recently advocated taking into consideration such regularity in designing multi-objective evolutionary algorithms. Following our previous work on using Local Principal Component Analysis for capturing the regularity, this paper presents a new approach for acquiring and using the regularity of the Pareto set in evolutionary algorithms. The approach is based on the Generative Topographic Mapping and can be regarded as an Estimation of Distribution Algorithm. It builds models of the distribution of promising solutions based on regularity patterns extracted from the previous search, and samples new solutions from the models thus built. The proposed algorithm has been compared with two other state-of-the-art algorithms, NSGA-II and SPEA2 on a set of test problems.

1 Introduction

Multi-objective optimization problems (MOP) arise from many practical applications where several objectives have to be optimized. In this paper, we consider the following continuous MOP:

$$\begin{array}{ll} \text{minimize} & F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \\ \text{subject to} & x \in X \end{array} \quad (1)$$

where $X \in \mathbb{R}^n$ is the decision (variable) space, $F : X \rightarrow \mathbb{R}^m$ consists of m continuous objective functions and \mathbb{R}^m is called the objective space. Very often, no point in X minimizes all the objectives simultaneously. The best tradeoffs among these objectives can be defined in terms of Pareto optimality.

Let $u, v \in \mathbb{R}^m$, u is said to *dominate* v if and only if $u_i \leq v_i$ for every $i \in \{1, 2, \dots, m\}$ and $u_j < v_j$ for at least one index $j \in \{1, 2, \dots, m\}$. A point $x^* \in S$ is *Pareto optimal* to (1) if there is no point x such that $F(x)$ dominates $F(x^*)$. $F(x^*)$ is then called a *Pareto optimal (objective) vector*. The set of all the Pareto optimal points is called the *Pareto set (PS)* and the set of all the Pareto optimal objective vectors is the *Pareto front (PF)*.

A number of evolutionary algorithms (EA) for dealing with multi-objective optimization problems, have been suggested over the past two decades [1,2]. These multi-objective evolutionary algorithms (MOEA) work with a population of candidate solutions and are able to produce a set of Pareto optimal vectors for approximating the PF. Most of these algorithms can be regarded as extensions of EAs for scalar optimization

problems. Selection, crossover and mutation are major operators in EAs. Conventional crossover and mutation operators can be used without any modification in MOEAs (although these operators may not lead to satisfactory performances in MOEAs), while the selection operators in scalar optimization EAs cannot be directly applied to MOPs. A major research issue in the area of MOEAs is the so-called fitness assignment [3], which assigns a relative fitness to each individual in a population for facilitating selection.

Estimation Distribution Algorithms (EDA) are a new class of EAs [4,5,6]. There is no traditional crossover or mutation in EDAs. Instead they explicitly extract global statistical information from the previous search and build a posterior probability distribution model of promising solutions, based on the extracted information. New trial solutions are sampled from the model thus built. Several EDAs for continuous MOPs have been proposed, among them are Mixture-based Iterated Density Estimation Evolutionary Algorithms (MIEDA) [7], EDAs based on Bayesian networks [8] and Voronoi-based Estimation of Distribution Algorithm (VEDA) [9].

Under mild conditions, the PS (or PF) of a continuous MOP is a piecewise continuous $(m - 1)$ dimensional manifold where m is the number of the objectives. This property has been used in several mathematical programming methods for approximating the PF [10]. In fact, it has also been found that, for the most widely-used test problems of continuous multi-objective optimization in the evolutionary computations, their PS are $(m - 1)$ dimensional linear or piecewise linear manifolds [11,12]. This regularity has been ignored in most current MOEAs.

Recently, we suggested that such regularity should be used in MOEAs for improving the algorithms' performances. We have proposed three EDAs [13,14,15] which employ Local Principal Component Analysis algorithms [16] for capturing and modeling the regularity of the Pareto set in a continuous MOP. The experimental results are very encouraging, the algorithms outperform NSGA-II [17] and SPEA2 [3] on several test problems with high interactions among the decision variables. Bueche et al [18] also attempted to use self-organizing mapping to learn the shape of the Pareto set of a MOP, although they have not explicitly discussed their method in the context of the regularity.

The Generative Topographic Mapping (GTM) [19] is a tool for extracting regularity from data. This paper continues our work on improving MOEAs by utilizing the regularity. We propose to use GTM in EDAs for extracting the regularity of the PS of (1). GTM can provide a probability model of promising solutions in terms of latent variables. Such a model is very easy for sampling new trial solutions.

The remainder of this paper is organized as follows: Section 2 gives a brief introduction to GTM. Section 3 presents the details of the model building and sampling techniques. The framework of the proposed algorithm is given in Section 4. Section 5 compares our proposed algorithm with NSGA-II and SPEA2 on a set of test problems. The final section provides concluding remarks.

2 The Generative Topographic Mapping

GTM can discover some underlying regularity of a set of unlabeled data in a high dimensional space. It considers a nonlinear transformation mapping from a latent-variable space with lower dimensionality L to the data space:

$$x = y(v, W) = W\phi(v)$$

where x is a point in the data space, v is the latent variable. W is a parameter matrix and $\phi(v)$ is a vector of prefixed basis functions.

GTM models the distribution of data x , for given value of W and v , to be a radially-symmetric Gaussian centered on $y(v, W)$ with variance β^{-1} :

$$p(x|v, W, \beta) = \left(\frac{\beta}{2\pi}\right)^{-n/2} \exp\left\{-\frac{\beta}{2}\|W\phi(v) - x\|^2\right\}$$

Therefore, $x = W\phi(v)$ can be envisaged as a central L -D manifold of the data, as illustrated in Fig. 1.

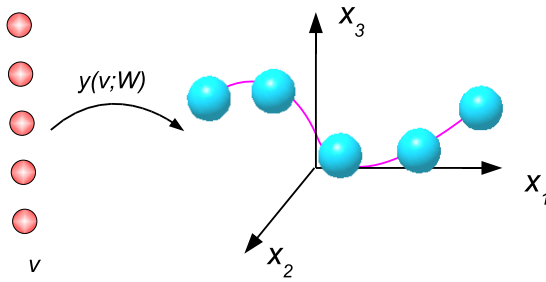


Fig. 1. Generative Topographic Mapping from 1-D latent space (left) to data space (right)

Given a number of points x^1, x^2, \dots, x^N in the data space, GTM estimates the values of W and β by maximizing the following log likelihood:

$$L(W, \beta) = \ln \prod_{i=1}^N \int p(x^i|v, W, \beta)p(v)dv.$$

where $p(v)$ is the distribution of v . In our experiments, we use the GTM code developed by NCRG group at Aston University¹. The details of GTM can be found in [19].

3 Model Building Based on GTM

3.1 Basic Idea

The population in a MOEA for (1) will hopefully approximate to the PS and be uniformly distributed around the PS as the search goes on. Therefore, we can envisage the solutions in the population as independent observations of a random vector whose center is an approximation of the PS of (1). Since the PS is a piecewise continuous ($m - 1$)

¹ The source codes are from <http://www.ncrg.aston.ac.uk/GTM>

dimensional manifold, a promising solution x can be regarded as an observation of the following n -D random vector:

$$\xi = \xi_1 + \xi_2 \quad (2)$$

where ξ_1 is uniformly distributed along a $(m - 1)$ dimensional manifold Φ . ξ_2 is a random noise vector. Φ is called the central manifold of ξ in this paper. For simplicity, we assume that ξ_1 and ξ_2 are independent of each other.

Under the above assumption, the modeling of promising solutions consists of two tasks: the modeling of the central manifold Φ and noise ξ_2 .

3.2 Modeling and Sampling

Given a population, which is a data set in R^n , we apply the GTM with $(m - 1)$ latent variables to it (where m is the number of objectives) and obtain the values of W and β . In GTM training procedure, we set the range of the latent variable $v = (v_1, \dots, v_{m-1})$ as:

$$-1 \leq v_i \leq 1 \quad i = 1, 2, \dots, m - 1.$$

Since the population often can not cover the whole PS thus the central manifold found can only approximate part of the PS. To overcome this problem and explore more ranges, in [13], [14] and [15], the search range is extended along the center manifold when we do sampling. In this paper, this idea is implemented by extending the range of the latent variables by 20%.

Sampling a new point x from the model built by the GTM is quite straightforward. We can do it in the following way:

Step 1: Uniformly and randomly select a v from $[-1.1, 1.1]^{m-1}$. Set $x^1 = y(v, W)$.

Step 2: Sample x^2 from $N(0, \frac{1}{\beta}I)$.

Step 3: Set $x = x^1 + x^2$.

where m is the objective number, I is a $n \times n$ identity matrix and n is the variable dimension.

4 Algorithm Framework

Our proposed modeling and sampling approach can be adopted as operators for producing new solutions in most current MOEAs. In this paper, we use the following algorithm framework:

Step 0 Initialization: Randomly generate a population P of N solutions, N is the population size.

Step 1 Reproduction:

Step 1.1 Modeling: Build a probability model based on statistical information extracted from P ;

Step 1.2 Sampling: Sample N new solutions from the model and store them in Q .

Step 2 Selection: Select N solutions from $P \cup Q$. Let these selected solutions to replace all the solutions in P .

Step 3 Stopping Condition: If the stopping condition is met, stop; otherwise, go to Step 1.

In our implementation, we use the selection scheme of NSGA-II in Step 2. We call our algorithm the model-based evolutionary algorithm with GTM (MEA/GTM).

5 Experimental Results

We use five test problems shown in Table 1 in our simulation studies.

Table 1. Test problems

MOP	n	Constraints	Objectives
$FON2$ [1]	30	$x \in [-2, 2]^n$	$f_1(x) = 1 - \exp(-\sum_{i=1}^n (x_i - \frac{1}{\sqrt{n}})^2)$ $f_2(x) = 1 - \exp(-\sum_{i=1}^n (x_i + \frac{1}{\sqrt{n}})^2)$
$OKA4$ [20]	2	$x \in [0, 8]^n$	$f_1(x) = 2 - \frac{1}{4}(x_1 - x_2 + 4)$ $+ \frac{1}{4}\sqrt{-x_1^2 - x_2^2 - 16 + 2x_1x_2 + 8x_1 + 8x_2}$ $f_2(x) = \frac{1}{4}(x_1 - x_2 + 4)$ $+ \frac{1}{4}\sqrt{-x_1^2 - x_2^2 - 16 + 2x_1x_2 + 8x_1 + 8x_2}$ $x_1 - 4\sqrt{x_1} - x_2 + 4 \leq 0$ $x_1 + 4\sqrt{x_1} - x_2 + 4 \geq 0$
$ZDT1.2$ [13]	30	$x \in [0, 1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x) \times (1.0 - \sqrt{\frac{f_1(x)}{g(x)}})$ $g(x) = 1.0 + \frac{9}{n-1} \sum_{i=2}^n (x_i^2 - x_1)^2$
$ZDT2.2$ [13]	30	$x \in [0, 1]^n$	$f_1(x) = \sqrt{x_1}$ $f_2(x) = g(x) \times (1.0 - (\frac{f_1(x)}{g(x)})^2)$ $g(x) = 1.0 + \frac{9}{n-1} \sum_{i=2}^n (x_i^2 - x_1)^2$
$DTLZ2.2$	10	$x \in [0, 1]^n$	$f_1(x) = \cos(\frac{\pi}{2}x_1) \cos(\frac{\pi}{2}x_2)(1 + g(x))$ $f_2(x) = \cos(\frac{\pi}{2}x_1) \sin(\frac{\pi}{2}x_2)(1 + g(x))$ $f_3(x) = \sin(\frac{\pi}{2}x_1)(1 + g(x))$ $g(x) = \sum_{i=3}^n (x_i^2 - x_1)^2$

$ZDT1.2$, $ZDT2.2$ and $DTLZ2.2$ ² are a respective modified version of $ZDT1$ [1], $ZDT2$ [1] and $DTLZ2$ [21]. In their original versions, the Pareto set is parallel to a coordinate axis which makes the problems easy to tackle.

We compared the performances of MEA/GTM, NSGA-II³ and SPEA2⁴ on the test problems in Table 1. The parameter setting of these algorithms are as follows: The

² $ZDT1.2$, $ZDT2.2$ and $DTLZ2.2$ are provided by Hui Li.

³ The source codes are from <http://www.iitk.ac.in/kangal/codes.shtml>

⁴ The source codes are from <http://www.tik.ee.ethz.ch/pisa>

population size for all the algorithms is set to 100 for 2-objective test instances and 200 for the 3-objective test instances. To have a fair comparison, the other parameters in NSGA-II and SPEA2 are set as in the default setting in their source codes. In GTM, the number of training steps is 15, the latent points number is 1×25 for 2-objective test instances and 5×5 in case of 3 objective, the basis function number is 2 for 2-objective problems and 2×2 in case of 3-objective and other parameters take their default values in their source codes. All the algorithms stops after 200 generations. We run each algorithm for each test instances 20 times.

To measure the performances of the algorithms, we use \mathcal{Y} [17] metric to measure the convergence of a population to the PF , and Δ [17] metric to measure the diversity of a population.

\mathcal{Y} [17] metric is defined as:

$$\mathcal{Y}(S, S^*) = \frac{1}{|S|} \sum_{x \in S} d(x, S^*), \quad (3)$$

where

$$d(x, S^*) = \min_{y \in S^*} \|F(x) - F(y)\|^2.$$

S is an obtained non-dominated set by an algorithm, S^* is a set which is uniformly distributed in Pareto Front.

Δ [17]⁵ metric is defined as:

$$\Delta(S, S^*) = \frac{\sum_{i=1}^m d(e_i, S) + \sum_{x \in S} |d(x, S) - \bar{d}|}{\sum_{i=1}^m d(e_i, S) + |S| \bar{d}}, \quad (4)$$

where $\{e_1, \dots, e_m\}$ are m extreme solutions in S^* and

$$d(x, S) = \min_{y \in S, y \neq x} \|F(x) - F(y)\|^2,$$

$$\bar{d} = \frac{1}{|S|} \sum_{y \in S} d(y, S).$$

It should be noticed that $d(x, S)$ measures the distance between point x and its nearest neighbor which is different from the original definition.

Table 2 gives the means and standard derivations of \mathcal{Y} and Δ on 20 runs at different stages, generation 40, 80, 120, 160 and 200, of the algorithms for each problem. Fig.2 shows the distributions of nondominated solutions generated in 20 runs in the objective space for each algorithm for each problem.

From the values of Δ in Table 2 it is clear that MEA/GTM outperforms NSGA-II and SPEA2 in terms of the diversity of the nondominated solutions found. Fig.2 also supports this claim. For all five test problems, MEA/GTM can cover the whole Pareto Fronts, while NSGA-II can only cover the Pareto Fronts of OKA4 and DTLZ2.2 and

⁵ We notice this metric is only fit in the case of small m .

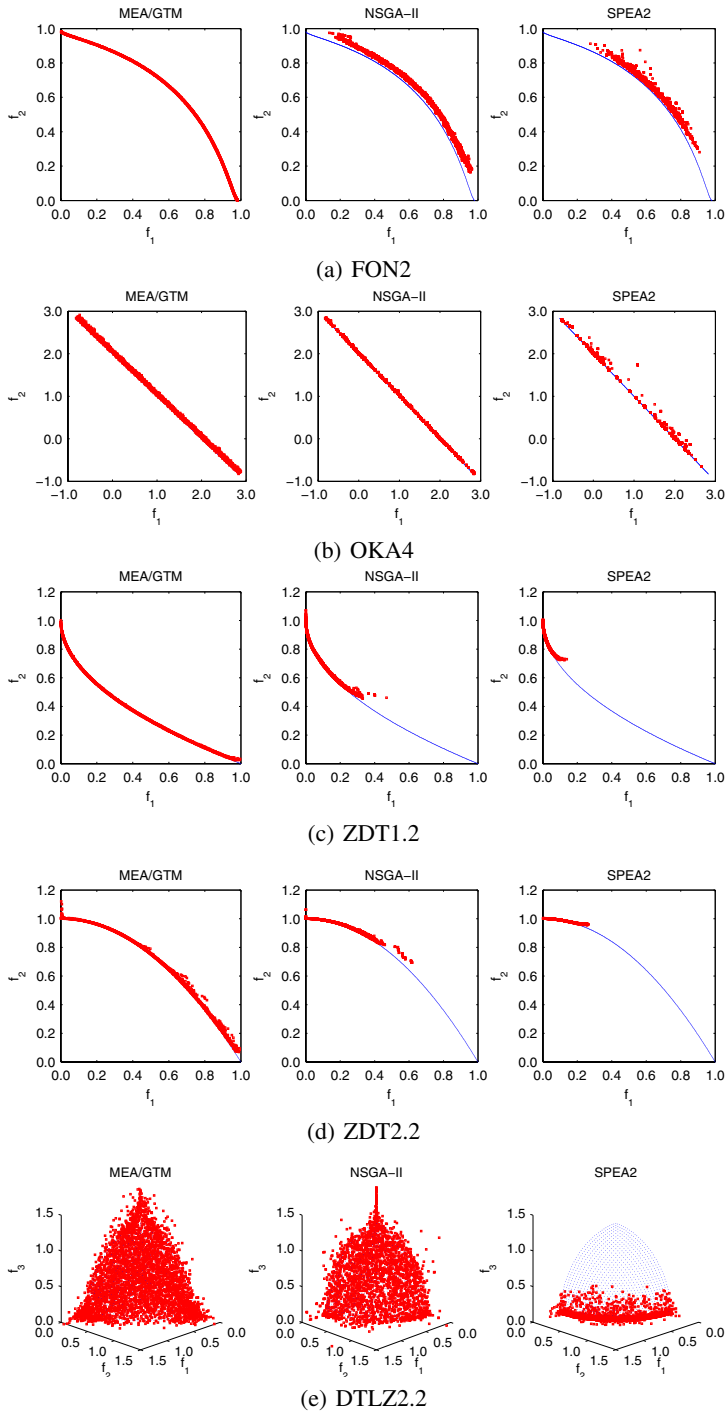


Fig. 2. Pareto fronts produced by MEA/GTM, NSGA-II and SPEA2

Table 2. Experimental results

Gen	MEA/GTM		NSGA-II		SPEA2	
	\mathcal{T}	Δ	\mathcal{T}	Δ	\mathcal{T}	Δ
FON2						
40	0.022 ± 0.002	0.500 ± 0.071	0.215 ± 0.027	0.849 ± 0.049	0.184 ± 0.025	0.769 ± 0.118
80	0.002 ± 0.000	0.160 ± 0.019	0.075 ± 0.007	0.748 ± 0.037	0.061 ± 0.009	0.726 ± 0.079
120	0.002 ± 0.000	0.154 ± 0.016	0.049 ± 0.003	0.690 ± 0.038	0.037 ± 0.005	0.750 ± 0.075
160	0.002 ± 0.000	0.151 ± 0.017	0.036 ± 0.003	0.629 ± 0.051	0.029 ± 0.003	0.786 ± 0.069
200	0.002 ± 0.000	0.156 ± 0.018	0.029 ± 0.002	0.514 ± 0.040	0.023 ± 0.003	0.798 ± 0.053
OKA4						
40	0.086 ± 0.008	0.483 ± 0.069	0.035 ± 0.005	0.706 ± 0.110	0.038 ± 0.012	0.730 ± 0.263
80	0.065 ± 0.005	0.485 ± 0.055	0.020 ± 0.003	0.821 ± 0.104	0.025 ± 0.010	0.757 ± 0.286
120	0.055 ± 0.005	0.468 ± 0.044	0.016 ± 0.003	0.873 ± 0.108	0.018 ± 0.005	0.815 ± 0.274
160	0.049 ± 0.004	0.449 ± 0.048	0.014 ± 0.003	0.878 ± 0.125	0.015 ± 0.006	0.899 ± 0.322
200	0.045 ± 0.003	0.399 ± 0.057	0.012 ± 0.002	0.841 ± 0.149	0.017 ± 0.011	0.811 ± 0.318
ZDT1.2						
40	0.013 ± 0.002	0.806 ± 0.053	0.024 ± 0.005	0.890 ± 0.043	0.012 ± 0.003	0.915 ± 0.025
80	0.008 ± 0.001	0.425 ± 0.085	0.011 ± 0.002	0.814 ± 0.022	0.005 ± 0.000	0.928 ± 0.013
120	0.006 ± 0.001	0.273 ± 0.042	0.009 ± 0.001	0.792 ± 0.020	0.005 ± 0.001	0.917 ± 0.010
160	0.004 ± 0.000	0.225 ± 0.023	0.008 ± 0.001	0.783 ± 0.028	0.005 ± 0.000	0.921 ± 0.013
200	0.004 ± 0.000	0.206 ± 0.018	0.007 ± 0.001	0.775 ± 0.019	0.005 ± 0.001	0.915 ± 0.013
ZDT2.2						
40	0.047 ± 0.021	0.937 ± 0.112	0.024 ± 0.010	0.933 ± 0.017	0.019 ± 0.007	0.968 ± 0.038
80	0.013 ± 0.006	0.765 ± 0.168	0.009 ± 0.004	0.947 ± 0.018	0.004 ± 0.001	0.965 ± 0.012
120	0.009 ± 0.003	0.551 ± 0.242	0.006 ± 0.002	0.915 ± 0.030	0.003 ± 0.000	0.981 ± 0.016
160	0.007 ± 0.002	0.412 ± 0.213	0.005 ± 0.003	0.870 ± 0.020	0.002 ± 0.000	0.982 ± 0.014
200	0.006 ± 0.001	0.329 ± 0.207	0.005 ± 0.003	0.870 ± 0.031	0.002 ± 0.000	0.984 ± 0.012
DTLZ2.2						
40	0.156 ± 0.031	0.491 ± 0.056	0.145 ± 0.036	0.647 ± 0.074	0.144 ± 0.113	0.950 ± 0.260
80	0.113 ± 0.024	0.473 ± 0.075	0.218 ± 0.292	0.602 ± 0.077	0.643 ± 0.393	0.798 ± 0.215
120	0.099 ± 0.021	0.448 ± 0.068	0.462 ± 0.640	0.576 ± 0.057	0.603 ± 0.384	0.832 ± 0.222
160	0.092 ± 0.019	0.433 ± 0.067	0.829 ± 0.725	0.563 ± 0.083	0.675 ± 0.506	0.814 ± 0.227
200	0.081 ± 0.012	0.427 ± 0.066	1.121 ± 0.499	0.532 ± 0.035	0.717 ± 0.441	0.875 ± 0.240

SPEA2 can not cover any of the Pareto Fronts. The reason may be: (1) MEA/GTM can cover the whole Pareto Fronts by extension and (2) MEA/GTM uniformly samples new solutions from the model.

In terms of \mathcal{T} metric, MEA/GTM performances much better than NSGA-II and SPEA2 on FON2 and DTLZ2.2. \mathcal{T} values in MEA/GTM is slightly higher than in SPEA2 on ZDT1.2 and ZDT2.2, it does not imply that MEA/GTM is worse. Since NSGA-II and SPEA2, as shown in Fig.2, cannot cover the whole Pareto Front as MEA/GTM does. Only on OKA4, MEA/GTM performs worse than NSGA-II and SPEA2. From Fig.2, however, EA/GTM can also approximate the Pareto Front very well.

These algorithms have been run in a desktop computer (Pentium(R) 4 3.40GHz CPU, 1.00GB of RAM). The average run time (in seconds) are listed in Table 3. Although

Table 3. Average run time(in seconds)

Method	FON2	OKA4	ZDT1.2	ZDT2.2	DTLZ2.2
NSGA-II	0.323	0.172	0.323	0.327	1.106
SPEA2	1.611	1.662	1.669	1.565	7.207
MEA/GTM	15.206	3.634	10.727	12.354	15.972

MEA/GTM needs more time to optimize each test problem than NSGA-II and SPEA2, it is still affordable especially when the fitness evaluation is much time consuming.

6 Conclusions

Incorporating problem-specific knowledge into evolutionary algorithms is a basic strategy for enhancing their performances [22]. The Pareto set of a continuous MOP is usually a piecewise continuous $(m - 1)$ -D manifold. Most current MOEAs ignore this regularity. Our recent work has showed that such regularity could be beneficial for modeling the population distribution in estimation of distribution algorithms for continuous multi-objective optimization. This paper demonstrated that GTM can be used for capturing and modeling the regularity and proposed an estimation of distribution algorithms with GTM for multi-objective optimization. The preliminary experiments show that our proposed method outperforms NSGA-II and SPEA2.

Our previous work in [13], [14] and [15] and this paper are on the design of EDAs by making the use of the regularity property. In the future, we plan to study how to incorporate such regularity and other properties of MOPs into other evolutionary algorithms.

References

1. Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, LTD, Baffins Lane, Chichester, 2001.
2. Ashish Ghosh and Satchidananda Dehuri. Evolutionary algorithms for multi-criterion optimization: A survey. *International Journal of Computing & Information Sciences*, 2(1):38–57, 2004.
3. Eckart Zitzler, Marco Laumanns, and Stefan Bleuler. A tutorial on evolutionary multiobjective optimization. In *Workshop on Multiple Objective Metaheuristics (MOMH 2002)*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, pages 3–37, Berlin, Germany, 2004. Springer-Verlag.
4. Pedro Larrañaga and Jose A Lozano, editors. *Estimation of distribution algorithms : a new tool for evolutionary computation*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.
5. Qingfu Zhang. On stability of fixed points of limit models of univariate marginal distribution algorithm and factorized distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 8(1):80–93, 2004.
6. Qingfu Zhang and Heinz Mühlenbein. On the convergence of a class of estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2):127–136, 2004.
7. Peter A N Bosman and Dirk Thierens. The naive MIDEA: A baseline multi-objective EA. In *Third International Conference on Evolutionary Multi-Criterion Optimization(EMO 2005)*, volume 3410 of *Lecture Notes in Computer Science*, pages 428–442, Guanajuato, Mexico, 2005. Springer.
8. Kumara Sastry, Martin Pelikan, and David E Goldberg. Decomposable problems, niching, and scalability of multiobjective estimation of distribution algorithms. IlliGAL Report 2005004, Illinois Genetic Algorithms Laboratory, 2005.
9. Tatsuya Okabe, Yaochu Jin, Bernhard Sendhoff, and Markus Olhofer. Voronoi-based estimation of distribution algorithm for multi-objective optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC 2004)*, pages 1594–1601, Portland, Oregon, USA, June 2004. IEEE Press.

10. Oliver Schütze, Sanaz Mostaghim, Michael Dellnitz, and Jürgen Teich. Covering Pareto sets by multilevel evolutionary subdivision techniques. In *Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*, pages 118–132, Faro, Portugal, April 2003. Springer.
11. 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.
12. Tatsuya Okabe, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. On test functions for evolutionary multi-objective optimization. In *Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *Lecture Notes in Computer Science*, pages 792–802, Birmingham, UK, September 2004. Springer.
13. Aimin Zhou, Qingfu Zhang, Yaochu Jin, Edward Tsang, and Tatsuya Okabe. A model-based evolutionary algorithm for bi-objective optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC 2005)*, pages 2568–2575, Edinburgh, U.K, September 2005. IEEE Press.
14. Aimin Zhou, Yaochu Jin, Qingfu Zhang, Tatsuya Okabe, and Edward Tsang. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. to be published in *Proceedings of the Congress on Evolutionary Computation (CEC 2006)*.
15. Qingfu Zhang, Aimin Zhou, and Yaochu Jin. Modelling the regularity in estimation of distribution algorithm for continuous multi-objective evolutionary optimization with variable linkages. 2006. to be submitted.
16. Nandakishore Kambhatla and Todd K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516, October 1997.
17. Kalyanmoy Deb. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
18. Dirk Büche, Michele Milano, and Petros Koumoutsakos. Self-organizing maps for multi-objective optimization. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2002)*, pages 152–155, New York, July 2002. Morgan Kaufmann Publishers.
19. Christopher M. Bishop, Markus Svensén, and Christopher K. I. Williams. GTM:the generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.
20. Tatsuya Okabe. *Evolutionary Multi-Objective Optimization: On the Distribution of Offspring in Parameter and Fitness Space*. PhD thesis, Honda Research Institute Europe GmbH, 2004.
21. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multi-objective optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), 2001.
22. Yaochu Jin, editor. *Knowledge Incorporation in Evolutionary Computation*. Number 167 in *Studies in Fuzziness and Soft Computing*. Springer, Berlin Heidelberg, 2004.