# An Estimation of Distribution Algorithm based on Decomposition for the Multiobjective TSP

Feng Gao
Department of Computer Science
East China Normal University
500 Dongchuan Road, Shanghai, China
Email: mantougf1987@gmail.com

Aimin Zhou and Guixu Zhang
Department of Computer Science
East China Normal University
500 Dongchuan Road, Shanghai, China
Email: {amzhou,gxzhang}@cs.ecnu.edu.cn

***Abstract*—The *multiobjective evolutionary algorithm based on decomposition (MOEA/D)* has gained much attention recently. It is suitable to use scalar objective optimization techniques for dealing with multiobjective optimization problems. In this paper, we propose a new approach, named *multiobjective estimation of distribution algorithm based on decomposition (MEDA/D)*, which combines MOEA/D with probabilistic model based methods for *multiobjective traveling salesman problems (MOTSPs)*. In MEDA/D, an MOTSP is decomposed into a set of scalar objective sub-problems and a probabilistic model, using both priori and learned information, is built to guide the search for each sub-problem. By the cooperation of neighbor sub-problems, MEDA/D could optimize all the sub-problems simultaneously and thus find an approximation to the original MOTSP in a single run. The experimental results show that MEDA/D outperforms BicriterionAnt, an ant colony based method, on a set of test instances and MEDA/D is insensible to its control parameters.**

## I. Introduction

The target of a *traveling salesman problem (TSP)* is to find a minimum-length tour for a salesman who would like to go from his home city to some other cities and come back to his home city, subject that each city will be visited once and only once. It has been proved that TSPs are NP-hard [1]. Due to their simplicity to describe and hardness to solve, TSPs have become combinational benchmarks to assess the performance of different heuristic optimization methods. Let $C = \{1, 2, \cdots, n\}$ be a set of city, and $D = (d_{i,j})_{n\times n}$ be a distance matrix where $d_{i,j}$ is the distance (or cost) between the $i^{th}$ and $j^{th}$ cities. Denote $\pi = (\pi_1, \pi_2, \cdots, \pi_n)$ as a permutation of $(1, 2, \cdots, n)$. A TSP can be formulated as

$$\min f(\pi) = \sum_{i=1}^{n-1} d_{\pi_i,\pi_{i+1}} + d_{\pi_n,\pi_1}. \tag{1}$$

In many applications, however, a salesman may consider not only the distance but also some other costs, such as time, risk, etc., simultaneously. In such cases, the problems become *multiobjective traveling salesman problems (MOTSPs)*. A MOTSP can be mathematically described as

$$\begin{aligned} \min \quad & F(\pi) = (f_1(\pi), f_2(\pi), \cdots, f_m(\pi)) \\ s.t. \quad & f_k(\pi) = \sum_{i=1}^{n-1} d_{\pi_i,\pi_{i+1}}(k) + d_{\pi_n,\pi_1}(k) \\ & k = 1, \cdots, m \end{aligned} \tag{2}$$

where $m$ is the number of objectives (criteria), $\pi = (\pi_1, \pi_2, \cdots, \pi_n)$ is a solution, i.e., a permutation of the city indices $(1, 2, \cdots, n)$, $D(k) = (d_{i,j}(k))_{n\times n}$ is the $k^{th}$ distance (cost) matrix, $f_k(\pi)$ is the $k^{th}$ objective, and $F(\pi)$ denotes the objective vector.

Since the objectives in (2) usually conflict with each other, for example a tour with shortest length might also be most expensive, there does not exist a single tour that can minimize all the objectives simultaneously. Like other *multiobjective optimization problems (MOPs)*, a set of tradeoff solutions, which is called *Pareto set (PS)* in the decision space (*Pareto front (PF)* in the objective space) [2], define the optimality of an MOTSP.

Like TSPs, deterministic methods are not suitable for MOTSPs when the number of cities is huge. Heuristic methods are thus used to approximate the PS (PF) of an MOTSP. Among them, *evolutionary algorithms (EAs)* [3], [4] are promising because they could approximate the whole PS (PF) in a single run. Actually, most of the *multiobjective evolutionary algorithms (MOEAs)* target to find a PS (PF) approximation which is as close to the true PS (PF) as possible and as diverse as possible [5]. To design MOEAs for MOTSPs, two issues, i.e., the reproduction operators and algorithm frameworks, should be considered. Some work related to the two issues is summarized as follows.

- *Reproduction Operators for MOTSPs:* The reproduction operators used for MOTSPs include: (1) neighborhood search: a new solution will be generated from the neighborhood of the parent [6], [7];(2) crossover/mutation operators: new solutions are generated by exchanging/mutating the components of the parents [8], [9]; (3) probabilistic model based approaches: new solutions are sampled from probability models which are built to capture the population distribution. The quantum algorithm [10] and ant colony optimization (ACO) [11], [12], [13], [14] based methods are using this idea; (4) local search strategies: in some work, the solutions are improved by local search methods [9], [10], [15], [16], [17].
- *MOEA Frameworks:* There are three MOEA frameworks which are widely used currently for dealing with generic MOPs: (1) MOEAs based on Pareto domination which

use the definition of Pareto optimality and density estimation strategies to select solutions [18]; (2) MOEAs based on performance indicators (IBEA) which apply the performance metrics to filter solutions [19]; and (3) MOEAs based on decomposition techniques (MOEA/D) which decompose a problem into a set of scalar objective problems and tackle them at the same time [20], [21].

Like ACO, *Estimation of distribution algorithms (EDAs)* use probabilistic models to extract the population distribution information and then to sample new trial solutions [22]. Unlike the case of TSPs, a single probabilistic model may not work well for MOTSPs. Thus how to maintain the probabilistic models play a key role in most of current probabilistic model based methods for MOTSPs. Since MOEA/D framework maintains a set of sub-problems, it is suitable for dealing with MOTSPs by using multiple probabilistic models. In this paper, we propose to combine MOEA/D and EDA for dealing with MOTSPs. In the new approach, named as *multiobjective estimation of distribution algorithm base on decomposition (MEDA/D)*, an MOTSP is decomposed into a set of sub-problems (TSPs), and each sub-problem is with a probabilistic model. A probabilistic model contains both priori heuristic information and learned information from the population in the running process, and a new tour is directly sampled from the probabilistic model. By the cooperation of neighbor sub-problems, MOEA/D could approximate the PF of an MOTSP in a single run.

The rest of the paper is organized as follows. In section II, the proposed algorithm MEDA/D is introduced with details. In Section III, MEDA/D is compared with BicriterionAnt [12] on a set of test instances and the sensitivity of MEDA/D to control parameters is also empirically studied. Finally, the paper is concluded in Section IV.

## II. MEDA/D Framework

### A. Basic Idea of MOEA/D

MOEA/D [20], [21] is a general EA framework for dealing with MOPs. Like generic MOEAs, an MOEA/D starts from an initial population of candidate solutions; in each iteration, it generates some new trial solutions and selects the fittest ones to the next iteration; and it repeats the process until some termination conditions are satisfied. Unlike generic MOEAs, an MOEA/D decomposes a problem into a set of scalar objective problems, and optimizes these problems simultaneously. This idea is realized by solution cooperation in neighborhood, i.e., the solutions in the same neighborhood are used to generate new trial solutions, and the new trial solutions only update the old solutions in the same neighborhood. The following two definitions are introduced to explain the idea.

- *Sub-problem:* A multiobjective optimization problems is decomposed into a set of scalar objective problems and each of them is called a *sub-problem*. Hopefully, the optimal solution of the $i^{th}$ sub-problem $g^i(\pi)$ lies in the PS (PF) of the original problem.
- *Neighborhood:* The neighborhood $B^i = (i_1, i_2, \cdots, i_K)$ of the $i^{th}$ sub-problem contains the indices of similar sub-problems, i.e., the $i_j^{th}$ $(j = 1, \cdots, K)$ sub-problems are the most similar ones to the $i^{th}$ sub-problem.

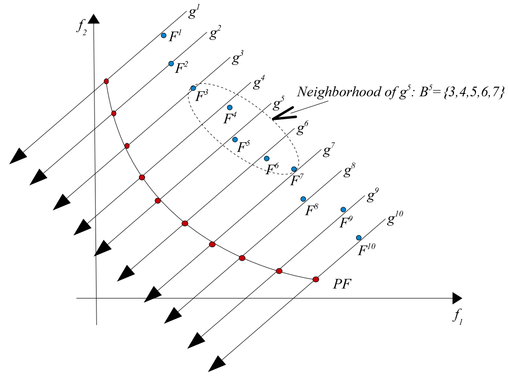



Fig. 1. An illustration of the basic idea of MOEA/D

Fig. 1 illustrates the basic idea of MOEA/D. In Fig. 1, a problem is scalarized into 10 sub-problems $\{g^1, \cdots, g^{10}\}$ and the neighborhood of the $5^{th}$ sub-problem contains the $3^{rd}$, $4^{th}$, $5^{th}$, $6^{th}$, and $7^{th}$ sub-problems.

### B. MOTSP Sub-problem Constructing

In this paper, we use the following Tchebycheff approach to define the sub-problems.

$$\min g^i(\pi) = g(\pi|\lambda^i, z^*) = \max_{1 \le j \le m} \lambda_j^i |f_j(\pi) - z_j^*| \quad (3)$$

where $\lambda^i = (\lambda_1^i, \cdots, \lambda_m^i)^T$ is a weight vector with the $i^{th}$ sub-problem, $z^* = (z_1^*, \cdots, z_m^*)^T$ is a reference point. It is clear that all the sub-problems are with the same form and can be differentiated by the weight vectors. If two vectors are close to each other, the corresponding sub-problems should be similar with each other and their optima should also be close in both the decision and objective spaces in most cases. By using the weight vectors, the neighborhood could be determined before the algorithm execution. It should be noticed that it is not necessary that all sub-problems have the same form.

In MEDA/D, the $i^{th}$ $(i = 1, \cdots, N)$ sub-problem is with

- a scalar objective problem $\min g^i(\pi)$ defined in (3),
- a decision vector $\pi^i$ and its objective vector $F^i = F(\pi^i)$,
- a set of indices $B^i$ of the neighborhood sub-problems which are with the $K$ closest weight vectors to $\lambda^i$, and
- a probability model $P^i$ which stores the information of the sub-problem.

### C. Probabilistic Model Constructing

A probabilistic model $P^i$ stores information extracted from the population for the $i^{th}$ sub-problem. Like ACO based approaches, $P^i$ contains both the prior heuristic and learned distribution information.

Without other information, it is natural to select the next city from some closest cities. Thus the distances between cities could be used as priori information. However, in the case of MOTSPs, the distance matrices are only available for the extreme sub-problems, i.e., the sub-problems with the objective functions as their sub-problem objectives. For the $i^{th}$ sub-problem, we define a pseudo distance matrix as

$$D^i = \sum_{k=1}^{m} \lambda_k^i D(k). \tag{4}$$

It is clear that the pseudo distance between the $s^{th}$ and $t^{th}$ cities is $d_{s,t}^i = \sum_{k=1}^{m} \lambda_k^i d_{s,t}(k)$ which is sub-problem dependent.

Let $Q^i = (q_{s,t}^i)_{n\times n}$ be a matrix which represents the learned information for the $i^{th}$ sub-problem, where $q_{s,t}^i$ denotes the connection strength between the $s^{th}$ and $t^{th}$ cities. Let

$$q_0 = \min_{s,t=1,\cdots,n,k=1,\cdots,m} d_{s,t}(k),$$

i.e., $q_0$ is the minimum distance between any pair of cities in all the given distance matrices. The learned information $Q^i$ is initialized as

$$q_{s,t}^i = q_0 \tag{5}$$

for all $s, t = 1, \cdots, n$ and $i = 1, \cdots, N$. In the running process, once the $i^{th}$ sub-problem is updated by a new trial solution $\pi$, its learned information matrix $Q^i$ is updated as

$$q_{s,t}^i = \begin{cases} (1-\rho)q_{s,t}^i + \rho q_0 & \text{if (s, t) is in } \pi \\ (1-\rho)q_{s,t}^i & \text{otherwise} \end{cases} \tag{6}$$

where $\rho$ denotes the learning rate.

Based on the above pseudo distance matrix $D^i$ and learned information matrix $Q^i$, we define the probability matrix $P^i$ for the $i^{th}$ sub-problem as

$$P^i = (p_{s,t}^i)_{n\times n} = \left( \frac{\left(q_{s,t}^i / d_{s,k}^i\right)^\alpha}{\sum_{k=1}^{n} \left(q_{s,k}^i / d_{s,k}^i\right)^\alpha} \right)_{n\times n} \tag{7}$$

where $\alpha$ balances the contributions of the priori and learned information; $p_{s,t}^i$ denotes the probability that the $s^{th}$ and $t^{th}$ cities are connected in the route of the $i^{th}$ sub-problem.

### D. Algorithm Framework

The main framework of MEDA/D is as follows.

---

**Step 1 Initialization:**

1.1 Convert an MTSP into $N$ sub-problems by (3) and randomly generate a solution for each sub-problem. Initialize the reference point $z^*$ as

$$z_k^* = \arg \min_{i=1,\cdots,N} f_k(\pi^i)$$

for $k = 1, \cdots, m$. Initialize the weight vectors $\lambda^i$ which are well distributed.

1.2 Initialize the neighborhood $B^i$ for each sub-problem $i$.

1.3 Initialize the probability matrix $P^i$ for each sub-problem $i$ as (7) in which $d_{s,t}^i$ is estimated as in (4) and $q_{s,t}^i$ is initialized as in (5).

**Step 2 Sample and Update:** For each sub-problem $i = 1, \cdots, N$, do

2.1 Sample a new solution $\pi = (\pi_1, \cdots, \pi_n)$: Let $C = \{1, \cdots, n\}$ be the unvisited cities. The first city $\pi_1$ is randomly selected from $C$ and let $C = C/\{\pi_1\}$. Then the next city $\pi_j$ $(j > 1)$ is randomly selected according to the probability $P^i$, i.e.,

$$\pi_j \sim \frac{p_{\pi_{j-1},\pi_j}^i}{\sum_{\pi_k \in C} p_{\pi_{j-1},\pi_k}^i}$$

where $\pi_j \in C$. Once $\pi_j$ is chosen, set $C = C/\{\pi_j\}$. Repeat the process until the whole tour is constructed.

2.2 Update of reference point: For each $k = 1, \cdots, m$, if $z_k^* > f_k(\pi)$, then set $z_k^* = f_k(\pi)$.

2.3 Update solutions: Set $A^i = \{k | k \in B^i, g^k(\pi) < g^k(\pi^k)\}$, i.e., $A^i$ denotes all the neighbor solutions which are worse than the newly sampled solution $\pi$ according to the corresponding sub-problem definitions. Randomly update at most $\mu$ solutions in $A^i$ by $\pi$.

2.4 Update probability model: For each sub-problem which is updated in the above step, update its corresponding learned information matrix $Q$ as in (6).

**Step 3 Stopping Criterion:** If the stop conditions are satisfied, then stop; otherwise go to **Step 2**.

---

We would like to make some comments to the algorithm.

- In **Step 1.1**, the weight vectors are generated to make them well distributed. In the case of bi-objective problems, the weight $\lambda^i = \left(\frac{i-1}{N-1}, \frac{N-i}{N-1}\right)$ for $i = 1, \cdots, N$. The method to generate weighs for problems with more than 2 objectives is referred to [2] for more details.
- In **Step 3**, the algorithm stops after a given maximum number of generations.

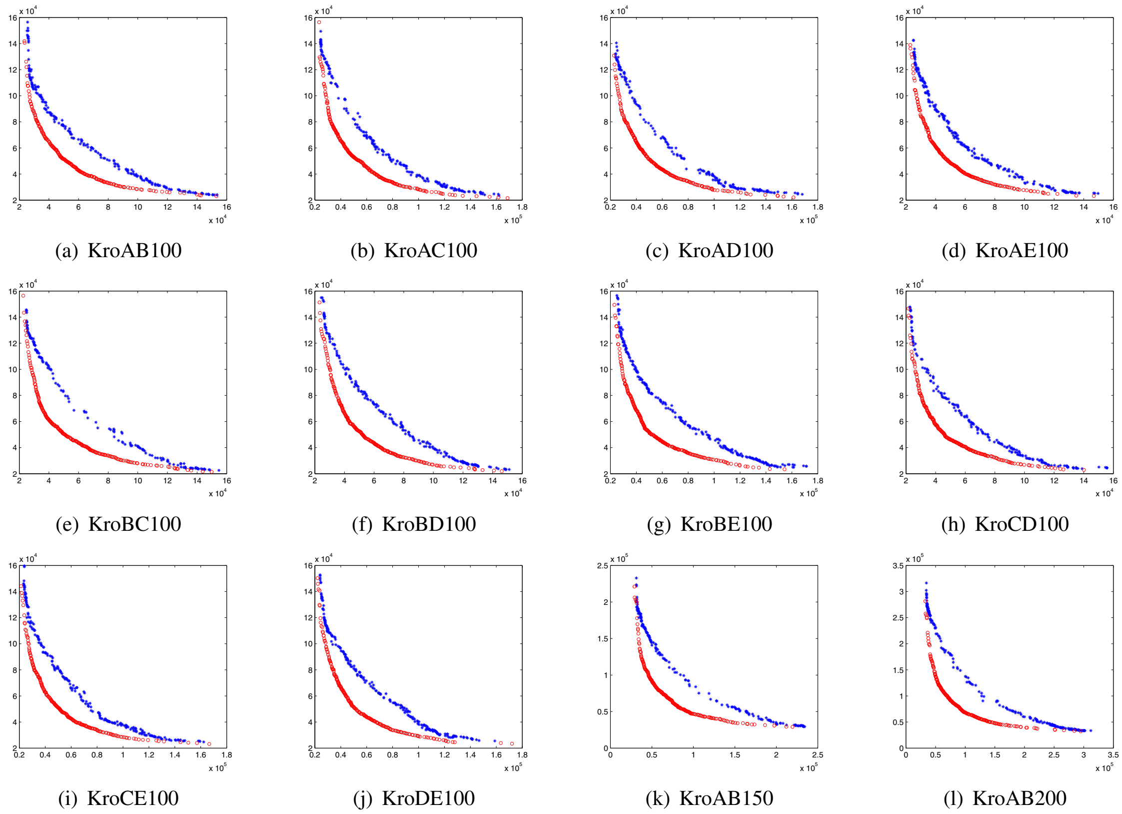
## III. Experimental Results

### A. Settings

In this section, we compare the proposed approach, MEDA/D, with the BicriterionAnt algorithm [12]. All the test problems are from the TSPLIB [23]. The distance matrices of two TSPs with the same number of cities are used to construct an MOTSP. The name of the MOTSP is from the two corresponding TSPs. For example, KroAB100 represents the combination of KroA100 (the 1st objective) and KroB100 (the 2nd objective).

To have a fair comparison, the Coverage and Hypervolume [24] metrics are used to access the performance of the two algorithms. The Coverage metric $C(A, B)$ measures the percentage of solutions in $B$ which are dominated by solutions in $A$. The Hypervolume metric $H(A, z)$ denotes the volume covered by a set $A$ and a reference point $z$. $H(A, z)$ can measure both the diversity and convergence of a set. To have a high value of $H(A, z)$ for a given reference point $z$, the approximation set $A$ must be as close to the true PF as possible and as diverse as possible. In the experiments, $z = (3 \times 10^5, 3 \times 10^5)$ is used for all problems.

The two algorithms are implemented by C++ and executed in the same environment. For both algorithms, the population

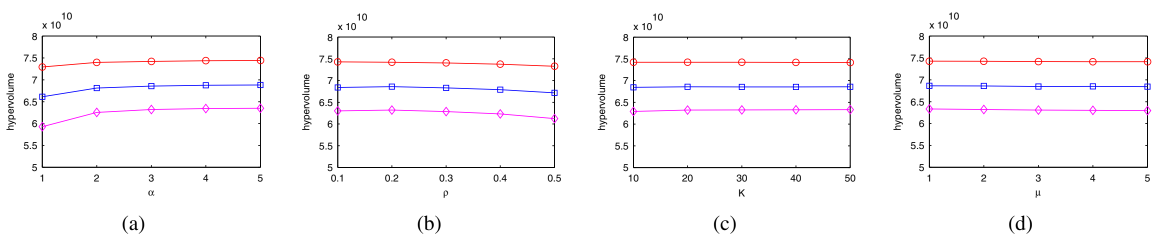


(a) KroAB100 (b) KroAC100 (c) KroAD100 (d) KroAE100

(e) KroBC100 (f) KroBD100 (g) KroBE100 (h) KroCD100

(i) KroCE100 (j) KroDE100 (k) KroAB150 (l) KroAB200

Fig. 2. The best approximations obtained by the two algorithms according to the Hypervolume values over 50 runs (the red circles are with MEDA/D, and the blue stars are with BicriterionAnt)




(a) (b) (c) (d)

Fig. 3. Sensitivity to MEDA/D parameters on KroAB100 (circles), KroAB150 (squares), and KroAB200 (diamonds)

size is set to be 250, and the number of generations is 1000 for all problems. The parameters of MEDA/D are as follows: $\alpha = 3$, $\rho = 0.1$, $K = 20$ and $\mu = 2$. The statistical results are based on 50 independent runs. The parameters of BicriterionAnt are the same as in [12].

### B. Statistical Results

The best approximations obtained by MEDA/D and BicriterionAnt are plotted in Fig. 2. It is clear that MEDA/D outperforms BicriterionAnt on all the given test instances. The solutions obtained by MEDA/D are more close to the PF and have better distributions than those obtained by BicriterionAnt especially in the middle areas. In the extreme areas, both algorithms could get similar results but the distributions of the solutions obtained by BicriterionAnt are better than those obtained by MEDA/D.

TABLE I
STATISTICAL RESULTS (MEAN±STD.) OF MEDA/D (A) AND BICRITERIONANT (B) OVER 50 RUNS.

| Instance | Coverage | | Hypervolume($\times 10^{10}$) | |
|---|---|---|---|---|
| | C(A, B) | C(B, A) | A | B |
| KroAB100 | **0.997**±0.007 | 0.000±0.001 | **7.436**±0.007 | 7.212±0.012 |
| KroAC100 | **0.999**±0.003 | 0.000±0.000 | **7.456**±0.007 | 7.210±0.013 |
| KroAD100 | **1.000**±0.000 | 0.000±0.000 | **7.462**±0.007 | 7.228±0.009 |
| KroAE100 | **0.999**±0.005 | 0.000±0.001 | **7.461**±0.010 | 7.234±0.010 |
| KroBC100 | **0.995**±0.009 | 0.000±0.001 | **7.470**±0.006 | 7.223±0.009 |
| KroBD100 | **0.996**±0.010 | 0.000±0.001 | **7.443**±0.007 | 7.204±0.010 |
| KroBE100 | **0.998**±0.004 | 0.000±0.000 | **7.401**±0.007 | 7.132±0.012 |
| KroCD100 | **0.997**±0.012 | 0.000±0.000 | **7.512**±0.008 | 7.294±0.010 |
| KroCE100 | **0.992**±0.014 | 0.001±0.002 | **7.466**±0.006 | 7.233±0.011 |
| KroDE100 | **1.000**±0.001 | 0.000±0.000 | **7.427**±0.008 | 7.185±0.012 |
| KroAB150 | **0.974**±0.029 | 0.001±0.002 | **6.875**±0.008 | 6.329±0.015 |
| KroAB200 | **0.958**±0.043 | 0.002±0.003 | **6.342**±0.008 | 5.347±0.015 |

Table I presents the statistical results of the Coverage and Hypervolume metrics. It shows that on average, at least $95.8\%$ solutions obtained by BicriterionAnt are dominated by those obtained by MEDA/D. On the contrary, however only at most $0.2\%$ solutions obtained by MEDA/D are dominated by those obtained by BicriterionAnt. This indicates that according to the convergence metric, MEDA/D performs much better than BicriterionAnt. The Hypervolume values represent both the diversity and convergence qualities of the final approximations. It can be seen from Table I that for all the problems, the areas covered by the solutions of MEDA/D are bigger than those of BicriterionAnt. It is clear that MEDA/D outperforms BicriterionAnt according to both the convergence and diversity metrics.

### C. Sensitivity to Algorithm Parameters

In MEDA/D, there are four parameters: $\alpha$ which balances the priori and learned information in the probabilistic model; learning rate $\rho$; neighborhood size $K$; and maximum number updated solutions $\mu$. In this section, we study the sensitivities of MEDA/D on these parameters. For this purpose, KroAB100, KroAB150, and KroAB200, are used in the study. The parameters are set as $\alpha = 1, 2, 3, 4, 5$, $\rho = 0.1, 0.2, 0.3, 0.4, 0.5$, $K = 10, 20, 30, 40, 50$, and $\mu = 1, 2, 3, 4, 5$. In the experiments, we only change one parameter and the other parameters are the same as in the previous section.

The average Hypervolume values over the change of parameters are plotted in Fig. 3. We can see that (1) the performance of MEDA/D increases slightly as $\alpha$ increases; (2) the performance of MEDA/D decreases slightly as $\rho$ increases; (3) $K$ and $\mu$, do not influence MEDA/D much. Overall, although $\alpha$ and $\rho$ influence the performances of MEDA/D, our new approach is not sensitivity to the algorithm parameters.

## IV. Conclusions

In this paper, we proposed to combine MOEA/D and EDA to solve multiobjective traveling salesman problems (MOTSPs). In the framework of MOEA/D, an MOTSP was decomposed into a set of scalar-objective sub-problems. A probabilistic model, which uses both priori heuristic and learned information, was built to store information and to sample new trial solutions for each sub-problem. The new approach, named MEDA/D, was compared with BicriterionAnt [12] on a variety of test instances. The statistical results shown that MEDA/D outperformed BicriterionAnt for both convergence and diversity metrics on the given test instances. The sensitivity study also indicated that MEDA/D was not sensible to its algorithm parameters. All the experiments in this paper demonstrated that probabilistic model based techniques for scalar-objective problems could be easily extended to tackle multiobjective problems by using the MOEA/D framework.

## Acknowledgment

This work is supported by the National Science Foundation of China (No.61005050), and the Fundamental Research Funds for the Central Universities.

## References

[1] R. Baraglia, J. Hidalgo, and R. Perego, "A hybrid heuristic for the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 6, pp. 613 –622, 2001.

[2] K. Miettinen, *Nonlinear multiobjective optimization*. Springer, 1999.

[3] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective evolutionary algorithms and applications*. Springer, 2005.

[4] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary algorithms for solving multi-objective problem*. Kluwer, 2002.

[5] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons LTD, 2001.

[6] H. Li and D. Landa-Silva, "Evolutionary multi-objective simulated annealing with adaptive and competitive search direction," in *IEEE Congress on Evolutionary Computation (CEC 2008)*, 2008, pp. 3311–3318.

[7] ——, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, 2011.

[8] Z. Yan, L. Zhang, L. Kang, and G. Lin, "A new MOEA for multi-objective TSP and its convergence property analysis," in *Evolutionary Multi-Criterion Optimization*. Springer Netherlands, 2003, pp. 342–354.

[9] W. Peng, Q. Zhang, and H. Li, "Comparison between MOEA/D and NSGA-II on the multi-objective travelling salesman problem," in *Multi-Objective Memetic Algorithms*. Springer, 2009, pp. 309–324.

[10] H. Li, D. Landa-silva, and X. Gandibleux, "Evolutionary multi-objective optimization algorithms with probabilistic representation based on pheromone trails," in *IEEE Congress on Evolutionary Computation (CEC 2010)*, 2010, pp. 1–8.

[11] L. M. Gambardella and M. Dorigo, "Ant-Q: A reinforcement learning approach to the traveling salesman problem," in *Twelfth International Conference on Machine Learning (ML-95)*, 1995, pp. 252–260.

[12] S. Iredi, D. Merkle, and M. Middendorf, "Bi-criterion optimization with multi colony ant algorithms," in *Evolutionary Multi-Criterion Optimization*. Springer, 2001, vol. 1993, pp. 359–372.

[13] K. Doerner, W. Gutjahr, R. Hartl, C. Strauss, and C. Stummer, "Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection," *Annals of Operations Research*, vol. 131, no. 1, pp. 79–99, 2004.

[14] A. M. P. Cardoso, M. Jesus, "MONACO - multi-objective network optimisation based on an ACO," in *Proceedings of X Encuentros de Geometria Computacional*, 2003.

[15] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 402 – 412, 2002.

[16] E. Angel, E. Bampis, and L. Gourvès, "Approximating the Pareto curve with local search for the bicriteria TSP(1,2) problem," in *Fundamentals of Computation Theory*. Springer, 2003, pp. 39–48.

[17] L. Paquete and T. Stützle, "A two-phase local search for the biobjective traveling salesman problem," in *Evolutionary Multi-Criterion Optimization*. Springer, 2003, pp. 479–493.

[18] J. Knowles and D. Corne, "The pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation," in *IEEE Congress on Evolutionary Computation (CEC 1999)*, 1999, pp. 98–105.

[19] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[20] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[21] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[22] P. Larranaga and J. A. Lozana, *Estimation of distribution algorithms : A new tool for evolutionary computation*. Kluwer, 2001.

[23] G. Reinelt, "TSPLIB - A traveling salesman problem library," *ORSA Journal on Computing*, vol. 3, pp. 376–384, 1991.

[24] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.