Scale Adaptive Reproduction Operator for Decomposition based Estimation of Distribution Algorithm

Bo Wang*, Hua Xu*, Yuan Yuan*
*State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
Email: wbthu@outlook.com, xuhua@tsinghua.edu.cn, yyxhdy@gmail.com

***Abstract*—Multi-objective evolutionary algorithm based on decomposition (MOEA/D) uses crossover operator which often either breaks the building blocks or mix them ineffectively. Multi-objective estimation of distribution algorithm based on decomposition (MEDA/D) evolves a probability vector for each sub-problem to guide the search instead of using crossover operator.However, since the number of the weight vectors in the neighborhood of each weight vector is relatively small and MEDA/D does not provide a way to maintain diversity, the performance of MEDA/D is limited. To overcome the drawbacks of MEDA/D, we proposed a new reproduction operator. This operator could promote diversity. We introduced it into MOEA/D framework and the new algorithm is called s-MEDA/D. We also prove that the parameter newly introduced has physical significance and the reproduction operator is not susceptible to the scale of the problem. The s-MEDA/D was tested on nine instances of the 0/1 multi-objective knapsack problem. Empirical evaluation suggests that the proposed algorithm is effective and efficient.**

I. Introduction

Many real-world optimization problems consist of several conflicting objectives. Optimizing a particular solution with respect to a single objective may result in inferior results with respect to other objectives. Evolutionary algorithms (EAs) are population-based metaheuristic optimization algorithms. Much research has been going on in the evolutionary algorithm community around multi-objective problem, resulting in a plethora of multi-objective evolutionary algorithms (MOEAs) [1, 2]. MOEAs are used to solve many real-world optimization problems [3, 4, 5].

Multi-objective evolutionary algorithm based on decomposition (MOEA/D) [6] has been proposed by Zhang and Li. MOEA/D decomposes the problem into a number of sub-problems by some decomposition approaches. All these sub-problems are processed simultaneously. The main idea behind MOEA/D is that if some sub-problems are close to each other, the solutions of these sub-problems could provide useful information for further optimizing. However, MOEA/D uses genetic operators which often either break the building blocks or mix them ineffectively [7]. Li et al. proposed multi-objective estimation of distribution algorithm based on decomposition (MEDA/D) [8]. MEDA/D uses the MOEA/D framework and it employs probability vectors to model and generate neighbouring solutions for each sub-problem instead of using genetic operators. However, since the number of the weight vectors in the neighborhood of each weight vector in the framework is often relatively small and MEDA/D does not provide any way to maintain diversity, the performance is limited. To overcome the drawbacks of MEDA/D, we propose a new reproduction operator. The main idea of our work is that the algorithm could prevent the probability vector from convergence by adding a small value to it. This provides a chance that solutions generated by the operator are different from their neighbouring solutions even all neighbouring solutions are the same. We would like this approach to reach two aims:

1) The parameter newly introduced has physical significance.

2) This method is not susceptible to the parameter and the scale of the problem.

The main contributions of this paper are as follows: We proposed a new reproduction operator based on probability vector and incorporated it into MOEA/D framework, the new algorithm is denoted as s-MEDA/D. We also provide the physical significance of the parameter. This approach meets the two aims mentioned above and this is verified by two proofs and empirical evaluation. The s-MEDA/D has been tested on 0/1 multi-objective knapsack problem and experimental results show that the proposed algorithm outperforms MOEA/D and MEDA/D. The influence of the parameter was also studied and reported.

The remaining parts of this paper are organized as follows, we briefly review previous works relevant to this work in section II. In section III, the proposed reproduction operator and the complete algorithm are described. In section IV, the 0/1 multi-objective knapsack problem and experimental results are reported. This section also provides a study on the parameter. Finally, conclusion and future work are in section V.

II. Preliminaries

This section provides some background knowledge and related work of this paper.

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

A. Multi-objective Optimization and Multi-objective Evolutionary Algorithms

Many real world problems involve more than one objective. Generally, objectives contradict each other, that is, improvement in one objective usually results in decreasing in other objectives. In this paper, we consider maximization problem. The multi-objective optimization problem (MOP) can be formally defined as follows.

$$\begin{cases} \max & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^{\mathrm{T}} \\ \text{subject to} & \mathbf{x} \in \Omega \end{cases} \quad (1)$$

where Ω is the n-dimensional decision space and $\mathbf{x} = (x_1, x_2, ..., x_n)^{\mathrm{T}}$ is decision variable vector. $\mathbf{F} : \Omega \rightarrow \mathbb{R}^m$ is made up of m objective functions. Given $\mathbf{g} = (g_1, g_2, ..., g_m)^{\mathrm{T}}, \mathbf{h} = (h_1, h_2, ..., h_m)^{\mathrm{T}} \in \mathbb{R}^m$, $\mathbf{g}$ is said to dominate $\mathbf{h}$ if and only if $g_i \geq h_i$ for every $i \in \{1, 2, \ldots, m\}$ and $g_j > h_j$, for at least one index $j \in \{1, 2, \ldots, m\}$. A decision vector $\mathbf{x}^* \in \Omega$ is Pareto optimal if there is no $\mathbf{x} \in \Omega$ such that $\mathbf{F}(\mathbf{x})$ dominates $\mathbf{F}(\mathbf{x}^*)$. The Pareto set (PS) includes all the Pareto optimal decision vectors. The Pareto front (PF) is defined as $PF = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m | \mathbf{x} \in PS\}$. Evolutionary algorithms are population based approaches. There are three multi-objective evolutionary frameworks widely used.The first is based on the concept of Pareto-dominance. The Non-dominated sorting genetic algorithm (NSGA-II) [9] is one of the most popular algorithm on this category. Several variants of NSGA-II were carried out for improving the performance or applying to different problems [10, 11]. The second framework is based on decomposition. MOEA/D [6], which is the representation of multi-objective evolutionary algorithms based on decomposition approach, have been paid attentions widely [12]. MOEA/D decomposes the multi-objective problem into sub-problems each of which is a scalar objective optimization problem and then optimizes these sub-problems simultaneously. The last multi-objective evolutionary framework is based on performance indicators [13]. These approaches use performance metrics, such as hypervolume [14], to filter evolutionary population.

B. Estimation of Distribution Algorithms

Estimation of distribution algorithms (EDAs) are a kind of evolutionary algorithms. EDAs use probabilistic models to represent population. The main difference between EDAs and genetic algorithm is that EDAs generate offsprings by sampling a probabilistic model build from selected population instead of using crossover and mutation operators. According to the probabilistic model employed, EDAs are divided into three categories: univariate EDA, bivariate EDA and multivariate EDA. Univariate EDA assumes independent relationship between variables of the problem. This kind of EDAs includes population-based incremental learning (PBIL) [15], compact genetic algorithm (cGA) [16], and the univariate marginal distribution algorithm (UMDA) [17]. These algorithms use probability vectors to model population. Han and Kim proposed the quantum-inspired evolutionary algorithm (QEA). It has been proven that QEA is a multimodel EDA [19]. QEA uses Q-individuals which are equivalent to probability vectors to model population [20]. Bivariate EDA and multivariate EDA take the interactions between variables into account. The representative algorithms are mutual information maximization for input clustering algorithm (MIMIC) [21], extended compact genetic algorithm (ECGA) [22] and the Bayesian optimization algorithm (BOA) [23].The development of EDAs accompanied by the increasing of complexity of the probabilistic models. However, complicated probabilistic model requires more computing resources. The complex algorithm performs better on a function that exactly matches the structure of its model, but the effectiveness of introducing the complexity model is unclear on other problems [24]. In EDAs, diversity maintain also need special attention. For example, in PBIL, mutation is used with the probability vector to promote diversity and it could aid PBIL in finding better final solutions [15]. Similarly, to prevent the premature convergence of QEA, Han and Kim presented H_ϵ gate which is way to prevent the Q-individuals from convergence. In multivariate EDA, Yang et al. use L1-regularized learning to control the network structure learning in BOA [26]. This is also a diversity promoting method.

C. Multi-objective Estimation of Distribution Algorithms

There are some algorithms combining estimation of distribution method with multi-objective evolutionary algorithm framework in literatures. Some of these algorithms [8, 27, 28, 29] are based on decomposition approach, mainly MOEA/D [6]. The major motivation behind MOEA/D is that any information provided by solutions of close weight vector should be helpful for optimizing. Therefore, building a probabilistic model from all neighboring solutions may use the information more efficiently than crossover operator.

Li et al. proposed an algorithm named multi-objective estimation of distribution algorithm based on decomposition (MEDA/D) [8]. MEDA/D is designed to solve the 0/1 multi-objective knapsack problem. MEDA/D uses probability vectors. This is the main characteristic that identify and differentiate MEDA/D from MOEA/D.At each generation, a probability vector is built from all neighboring solutions for a sub-problem. Then new solution is created by sampling the probability vector. To solve continuous multi-objective optimization problem using probabilistic model based approach, Zhou et al. incorporated multivariate Gaussian models into MOEA/D [27]. This algorithm builds multivariate Gaussian models from the neighboring solutions. A new solution is firstly sampling from the Gaussian model and then added a small variation by polynomial mutation [30] to enhance diversity. Zhou, Gao, and Zhang designed an algorithm using the MOEA/D framework and probabilistic model based method to solve multi-objective traveling salesman problem [29].

Zhou, Zhang, and Jin proposed MMEA, which is short for probabilistic model-based multi-objective evolutionary algorithm [31]. MMEA clusters population in the objective space into a number of subpopulations. In each subpopulation, a probabilistic model is built for the distribution of the Pareto-optimal solutions in the decision space. Approximating the set of Pareto-optimal solutions in both the decision and objective spaces could promote the population diversity and provide more information for decision maker.

III. The Proposed Algorithm

In this section, our proposed algorithm is described in detail. Firstly, we describe the decomposition method and the framework of MOEA/D for the sake of completeness. Then the proposed scale adaptive reproduction operator is discussed.

A. Decomposition Methods and General Framework

MOEA/D decomposes the multi-objective problem into a number of scalar problems. This framework is not limited to some specific decomposition methods. In this paper, two decomposition approaches are considered.

1) Weighted Sum Decomposition: The multi-objective functions can be transformed into a single objective function by weighting method. The weight sum of the objective functions is defined as follows.

$$\begin{cases} \max & g^w(\mathbf{x}|\boldsymbol{\lambda}) = \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \Omega \end{cases} \tag{2}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_m)^{\mathrm{T}}$ is a weight vector. Usually, a weight vector $\boldsymbol{\lambda}$ should satisfy normalization condition $\sum_{1\leq i\leq m}^{m} \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i \in \{1, 2, ..., m\}$. With these conditions, theoretical results show that all the Pareto optimal solutions of original problem can be found by the weighting method if original problem is convex [32]. When the problem is nonconvex, not all of the Pareto optimal solutions could be found by weighting method. Each weight vector corresponds to a sub-problem. In MOEA/D, different weight vectors are generated at beginning and remain unchanged. Therefore, all sub-problems are remain unchanged throughout the entire procedure.

2) Tchebycheff Decomposition: We first introduce the ideal objective vector and utopian objective vector before introducing Tchebycheff decomposition. The ideal objective vector $\mathbf{z}^* = (z_1^*, z_2^*, ..., z_m^*)^{\mathrm{T}}$ maximizes each of the objective functions, that is $z_i^* = \sup_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$ for all $i \in \{1, 2, ..., m\}$. A utopian objective vector $\mathbf{z}^{**}$ is an infeasible objective vector whose components are formed by $z_i^{**} = z_i^* + \epsilon_i$ for all $i \in \{1, 2, ..., m\}$ and $\epsilon_i > 0$. These vectors can be served as the reference point [33]. The weighted Tchebycheff problem is of the form

$$\begin{cases} \min & g^t(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^*) = \max_{1\leq i\leq m} \{\lambda_i |f_i(\mathbf{x}) - z_i^*|\} \\ \text{subject to} & \mathbf{x} \in \Omega \end{cases} \tag{3}$$

Algorithm 1 MOEA/D Framework

```
 1: $EP \leftarrow \{\}$
 2: Initialize a set of weight vectors
 3: Figure out the closest weight vectors to each weight vector
 4: Generate an initial population and evaluate the population
 5: Initialize the reference point $\mathbf{z}$, if needed
 6: while termination criteria not met do
 7:     for $i = 1, 2, \ldots, K$ do
 8:         Generate a new solution $\mathbf{x}$ for the $i$ sub-problem
 9:         Improve $\mathbf{x}$ by problem-specific method
10:         Update reference point $\mathbf{z}$, if needed
11:         Update neighboring solutions
12:         Update $EP$
13:     end for
14: end while
```

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_m)^{\mathrm{T}}$ is the weight vector. It has been proven that, given $\mathbf{x}^* \in \Omega$ who is Pareto optimal, there exists a weighting vector $\mathbf{0} < \mathbf{w} \in \mathbb{R}^m$ such that $\mathbf{x}^*$ is a solution of weighted Tchebycheff problem where the reference point is the utopian objective vector [32]. The ideal objective vector and utopian objective vector are unknown. In MOEA/D, an components in the reference point is set to be the best value found for the corresponding objective.

3) MOEA/D Framework: The framework of MOEA/D [6] is shown in Algorithm 1. The details of the procedure are described as follows. Lines 1-5 are initialization. Firstly, an external population is initialized to be an empty set. It is used to store all nondominated solutions found by the algorithm and as the final output of the algorithm. Then in line 2, a set of K weight vectors is generated, which is denoted as $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_K\}$. In line 3, the distances between any two weight vectors are worked out. For a weight vector $\boldsymbol{\lambda}_i$, T closest weight vectors compose the neighborhood of $\boldsymbol{\lambda}_i$. The indexes of these neighboring weight vectors are stored in a set $B(i)$, that is, if $\boldsymbol{\lambda}_j$ is the neighborhood of $\boldsymbol{\lambda}_i$, then $j \in B(i)$. In line 4, an initial population $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_K\}$ is created randomly or by a problem-specific method. For a solution $\mathbf{x}_i$, the objective function $\mathbf{F}(\mathbf{x}_i)$ is calculated. Whether algorithm needs to implement line 5 depends on the decomposition method used. If Tchebycheff decomposition approach is employed, a reference point should be initialized. Elements in $\mathbf{z}$ are estimated by the max value for objective f_i.

Lines 6-13 are the main loop. At each iteration, MOEA/D deals with the sub-problems sequentially. In line 8, a new solution $\mathbf{y}$ is generated based on the solutions from neighbouring sub-problems. In MOEA/D, genetic operators are used. In our algorithm, this procedure is handled by scale adaptive reproduction operator which will be discussed in section III-B. In line 9, one can use problem-specific method or local search technologies to improve the new solution. If the solution is not feasible, that is, $\mathbf{y} \notin \Omega$, a repair method should be used upon it. If a reference point is involved, there should be a progress updating the reference point according to the objective function value of the new solution. To be more precise, for each $j = 1, 2, ..., m$, if $z_j < f_j(\mathbf{y})$, then $z_j = f_j(\mathbf{y})$. Line 11 updates the neighboring solutions. Because the new solution is generated based on the information of the neighboring solutions, the neighboring solutions and the corresponding fitness vectors should be updated.Let $g(\mathbf{x})$ be $g^w(\mathbf{x}|\boldsymbol{\lambda})$ in Equ.(2) or $g^t(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^*)$ in Equ.(3). If $g(\mathbf{y}) \geq g(\mathbf{x}_j)$, then $\mathbf{x}_j = \mathbf{y}$. Line 12 updates EP. This procedure includes two steps: first, remove all vectors dominated by $\mathbf{F}(\mathbf{y})$; second, add $\mathbf{F}(\mathbf{y})$ if it is not dominated by any vector in EP.

B. Scale Adaptive Reproduction Operator

The proposed reproduction operator is for binary coded problems. It is based on probability vectors. A probability vector for the i-th sub-problem is denoted as

$$\mathbf{P}^i = (p_1^i, p_2^i, ..., p_n^i) \tag{4}$$

where n is the dimension of the problem, each individual component p_j^i must be in the interval $[0, 1]$ for all $j \in \{1, 2, ..., n\}$. The indexes of neighboring weight vectors of ith sub-problem are stored in $B(i) = \{i_1, i_2, ..., i_T\}$. Then component p_j^i is calculated as follows:

$$p_j^i = \frac{\sum_{l=1}^{T} x_j^{i_l} + \xi}{T + 2\xi} \tag{5}$$

where T is the neighborhood size. The jth element of the solution for the i_l sub-problem is denoted as $x_j^{i_l}$. The ξ in Equ.(5) is the small value adding to probability vector. It is used to promote diversity. It is defined as follows.

$$\xi = \frac{T \cdot s}{n - 2s} \tag{6}$$

where s is a parameter of this reproduction operator and will be discussed after the sampling method is described. A new solution $\mathbf{y}^i = (y_1, y_2, ..., y_n)^{\mathrm{T}}$ is generated based on $\mathbf{p}^i$.

$$y_j = \begin{cases} 1 & \text{if } rand(0,1) < p_j^i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $rand(0,1)$ is a random function returning a real number between 0 and 1 following uniform distribution. The detailed steps of the reproduction procedure are shown in Algorithm 2.

The parameter s in Equ.(6) determines the least number of components of the new solution $\mathbf{y}^i$ sampled from $\mathbf{P}^i$ in expectation. Even if all neighbouring solutions have the same value, the new solution would have no less than s bits that are different from the neighbouring solutions in expectation. The proof is as follows:

Proof: Suppose $\mathbf{x}^l = \mathbf{x}^* = (x_1^*, x_2^*, ..., x_n^*)^{\mathrm{T}}$ for all $l \in B(i)$. That means all neighbouring solutions are same. Substitute Equ.(6) into Equ.(5) yields

$$\begin{aligned} p_j &= \frac{(n-2s)\sum_{l=1}^{T} x_j^{i_l} + T \cdot s}{(n-2s)T + 2T \cdot s} && (8) \\ &= \frac{(n-2s)T \cdot x_j^* + T \cdot s}{(n-2s)T + 2T \cdot s} && (9) \\ &= \frac{(n-2s)x_j^* + s}{n} && (10) \end{aligned}$$

Algorithm 2 Scale Adaptive Reproduction Operator

```
procedure REPRODUCE($\mathbf{X}$, $B(i)$, $\mathbf{y}^{new}$)
    $\mathbf{P} \leftarrow \mathbf{0}$
    for each $l \in B(i)$ do
        for $j = 1, 2, \ldots, n$ do
            if $x_j^l == 1$ then
                $p_j \leftarrow p_j + 1$
            end if
        end for
    end for
    $\xi \leftarrow (T * s)/(n - 2 * s)$
    for $j = 1, 2, \ldots, n$ do
        $p_j \leftarrow (p_j + \xi)/(T + 2 * \xi)$
        if $rand(0,1) < p_j$ then
            $y_j^{new} \leftarrow 1$
        else
            $y_j^{new} \leftarrow 0$
        end if
    end for
end procedure
```

If $x_j^* = 0$, then $p_j = s/n$, with Equ.(7)

$$P(y_j = 1 | x_j^* = 0) = \frac{s}{n} \tag{11}$$

If $x_j^* = 1$, then $p_j = (n-s)/n$, with Equ.(7)

$$P(y_j = 0 | x_j^* = 1) = 1 - \frac{n-s}{n} = \frac{s}{n} \tag{12}$$

Therefore, the probability that a bit sampled differs from original value is s/n. And because the independent between variables, the number of bits changed, denoted as N, is a random variable follows binomial distribution of which the parameters are n and s/n, i.e. $N \sim B(n, s/n)$. Then the expected value of N is s. ■

The probability that new solution having at least one bit differing from the original is

$$P_{change} = 1 - (1 - \frac{s}{n})^n \tag{13}$$

when $n \gg s$ (the problem size is often far greater than parameter s, so this condition always be true), the probability P_{change} almost not varying with n. The proof is as follows:

Proof: Let $\varepsilon = s/n$ and it is close to 0 and then Equ.(13) becomes the following form

$$\begin{aligned} P_{change} &= 1 - (1 - \frac{s}{n})^n && (14) \\ &= 1 - (1 - \varepsilon)^{s/\varepsilon} && (15) \end{aligned}$$

Compute the Taylor series expansion of $P_{change}(\varepsilon)$ with respect to ε around the expansion point 0 yields

$$\begin{aligned} P_{change}(\varepsilon) &= 1 - (1 - \varepsilon)^{s/\varepsilon} && (16) \\ &= 1 - e^{-s} + o(\varepsilon^2) && (17) \\ &\simeq 1 - e^{-s} && (18) \end{aligned}$$

where $o(\varepsilon^2)$ is the second order small quantity. The probability P_{change} almost not varying with n. ■

Brief discussions:

1) Why use Equ.(6) but not use ξ directly?: If using ξ directly, the operator could promote the diversity, but it is sensitive with the size of the problem. And ξ do not have clear physical interpretation.

2) What is the difference between the proposed operator and mutation operator?: Mutation operator can used beside the probability vector to improve the performance as PBIL[15]. In the view of estimation of distribution, mutation operator is a genetic operator, so we integrated the mutation operation into the model as a single step in the algorithm.

3) What if we set $s = 0$?: If s is set to be 0, then ξ in Equ.(6) is 0, and the operator is the same as that proposed in [8].

IV. Experiments

The algorithm is tested on the 0/1 multi-objective knapsack problem. This section presents the test problem, evaluation setup, performance metrics and comparison results with MOEA/D. At last, the main parameter s of the proposed operator is studied.

TABLE I. Comparison between s-MEDA/D and MOEA/D

Weighted Sum Decomposition

Problem	m	C-Metric		Hypervolume Metric	
		C(s-MEDA/D, MOEA/D)	C(MOEA/D, s-MEDA/D)	s-MEDA/D	MOEA/D
KN-250	2	**0.877179 ± 0.067145**	0.060381 ± 0.052836	(9.832523 ± 0.010875)E+07	(9.833248 ± 0.006205)E+07
	3	**0.894277 ± 0.032293**	0.036038 ± 0.015913	**(9.276245 ± 0.009366)**E+11	(9.240465 ± 0.006721)E+11
	4	**0.841459 ± 0.037736**	0.028743 ± 0.010869	**(8.006844 ± 0.009728)**E+15	(7.936617 ± 0.006874)E+15
KN-500	2	**0.983875 ± 0.013797**	0.000279 ± 0.001044	**(4.061472 ± 0.002780)**E+08	(4.053955 ± 0.002866)E+08
	3	**0.994918 ± 0.005147**	0.000397 ± 0.000909	**(7.665099 ± 0.008639)**E+12	(7.581944 ± 0.006951)E+12
	4	**0.989432 ± 0.004501**	0.000226 ± 0.000322	**(1.335180 ± 0.002071)**E+17	(1.306883 ± 0.002032)E+17
KN-750	2	**0.996841 ± 0.008467**	0.000000 ± 0.000000	**(8.900160 ± 0.006867)**E+08	(8.853172 ± 0.007248)E+08
	3	**0.999755 ± 0.001035**	0.000000 ± 0.000000	**(2.693014 ± 0.002851)**E+13	(2.649068 ± 0.003116)E+13
	4	**0.999262 ± 0.000478**	0.000000 ± 0.000000	**(7.029037 ± 0.011975)**E+17	(6.778892 ± 0.016631)E+17

Tchebycheff Decomposition

Problem	m	C-Metric		Hypervolume Metric	
		C(s-MEDA/D, MOEA/D)	C(MOEA/D, s-MEDA/D)	s-MEDA/D	MOEA/D
KN-250	2	**0.850588 ± 0.084157**	0.090291 ± 0.056651	(9.829682 ± 0.014494)E+07	(9.833269 ± 0.007742)E+07
	3	**0.890765 ± 0.040440**	0.039502 ± 0.020306	**(9.277022 ± 0.009184)**E+11	(9.239961 ± 0.006990)E+11
	4	**0.835567 ± 0.033432**	0.031279 ± 0.013262	**(8.005976 ± 0.010006)**E+15	(7.936030 ± 0.010029)E+15
KN-500	2	**0.978685 ± 0.024043**	0.000976 ± 0.002797	**(4.060804 ± 0.003585)**E+08	(4.054188 ± 0.003719)E+08
	3	**0.995862 ± 0.004070**	0.000273 ± 0.000551	**(7.666164 ± 0.007918)**E+12	(7.585415 ± 0.008352)E+12
	4	**0.988791 ± 0.005239**	0.000429 ± 0.001172	**(1.334330 ± 0.002023)**E+17	(1.306300 ± 0.001688)E+17
KN-750	2	**0.998460 ± 0.003947**	0.000000 ± 0.000000	**(8.902155 ± 0.005378)**E+08	(8.853976 ± 0.006809)E+08
	3	**0.999687 ± 0.000850**	0.000000 ± 0.000000	**(2.693431 ± 0.002328)**E+13	(2.649361 ± 0.003369)E+13
	4	**0.999144 ± 0.000685**	0.000000 ± 0.000000	**(7.028353 ± 0.010077)**E+17	(6.780390 ± 0.018409)E+17

A. 0/1 Multi-objective Knapsack Problem

The classical 0/1 knapsack problem is a well-known combinatorial optimization problem [34]. It can be extended for an arbitrary number of knapsacks. Given n items and m knapsacks. Each item has a weight and a profit while each knapsack has a capacity. Select items and place them into every knapsacks. The objective is to maximize the sum of the values of each knapsack under the condition that the sum of the weights must be less than the knapsack's capacity. The formalize definition of 0/1 multi-objective knapsack problem is as follows.

$$\begin{cases} \text{max} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^{\mathrm{T}} \\ & f_i(\mathbf{x}) = \sum\limits_{j=1}^{n} p_{ij} x_i, i = 1, 2, ..., m \\ \text{subject to} & \sum\limits_{j=1}^{n} w_{ij} x_j \leq c_i, i = 1, 2, ..., m \end{cases} \tag{19}$$

where p_{ij} and w_{ij} are the profit and the weight of jth item according to ith knapsack respectively. The c_i is the capacity of ith knapsack. A solution, $\mathbf{x} = (x_1, x_2, ..., x_n)^{\mathrm{T}}, x_j \in \{0, 1\}$ for all $j \in (1, 2, ...n)$, is a binary vector where $x_j = 1$ if the jth item is selected and $x_j = 0$ otherwise. If a solution is not feasible, it should be repaired by some repair method. In this paper, we used the same instances and the same greedy repair method which have been used in MOGLS[35] and MOEA/D [6]. The main idea of this repair method is that the item with the heavy weights and small profits is more likely to be removed. It repairs the solution by removing the item one by one until the solution is feasible [6]. Nine instances (3 objectives × 3 sizes) are used to test the algorithms. An instance is denoted as KN-n-m where n is the number of items and m is the number of objectives. For example, instance that has bi-objectives and 500 items is denoted as KN-500-2.

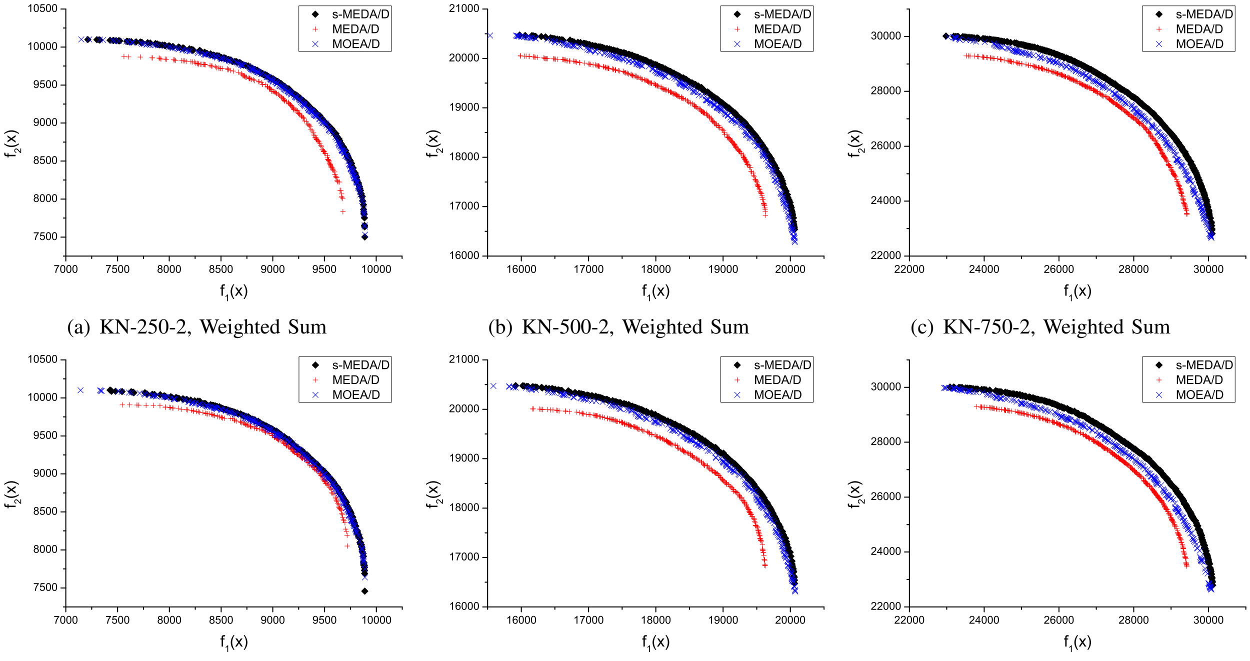
B. Evaluation Setup and Performance Metric

1) Evaluation Setup: In the MOEA/D framework, an integer parameter H controls the generating of weight vectors. For a weight vector, each element takes a value from

$$\{\frac{0}{H}, \frac{1}{H}, ..., \frac{H}{H}\} \tag{20}$$

The number of such vectors is $N = C_{H+m-1}^{m-1}$. For fair comparison, the weight vectors used in this paper is the same with that in MOEA/D and MEDA/D. That is, for three bi-objectives instances, the number of weight vectors are 150, 200 and 250 respectively. For instance having three objectives, the parameter H is set to be 25. For instance having four objectives, H is 12. Another important parameter in all these three algorithms is T which is the number of the weight vectors in the neighborhood of each weight vector. For all instances, T is set to be 10. MOEA/D uses genetic operators. The mutation probability is set to be 0.01. Parameter s in the proposed s-MEDA/D is set to be 0.4. The termination condition is the max call number of the evaluation function. These max call numbers are set to be the same with that in MOGLS[35] and MOEA/D [6]. Each instance was tested for 30 independent runs. All parameters are summarized in Table.II.

2) Performance Metric: Given A and B, which are two approximations to the Pareto front, the Coverage metric $C(A, B)$



(a) KN-250-2, Weighted Sum

(b) KN-500-2, Weighted Sum

(c) KN-750-2, Weighted Sum

(d) KN-250-2, Tchebycheff

(e) KN-500-2, Tchebycheff

(f) KN-750-2, Tchebycheff

Fig. 1. The approximations of 2 objective instances obtained by the three algorithms over 30 runs.

TABLE II. PARAMETERS SETTING

Instance	N	Max Evaluations ($\times 10^3$)	T	$P_{mutation}$ (MOEA/D)	s (s-MEDA/D)
KN-250-2	150	75	10	0.1	0.4
KN-500-2	200	100			
KN-750-2	250	125			
KN-250-3	351	100			
KN-500-3	351	125			
KN-750-3	351	150			
KN-250-4	455	125			
KN-500-4	455	150			
KN-750-4	455	175			

is defined as the percentage of vectors in B that are dominated by some vectors in A. Formal definition is as follows:

$$C(A,B) = \frac{|\{u \in B | \exists v \in A : v \text{ dominates } u\}|}{|B|} \quad (21)$$

Zitzler and Thiele use the size of the union of all rectangles covered by the Pareto-optimal solutions to evaluate a MOEA independent of the other algorithms [36]. The higher the hypervolume, the better the approximation. The hypervolume metric needs a reference point. In our experiments, the reference point is set to be the origin for all instances. Differences between these algorithms were evaluated by Wilcoxon signed-rank test, and considered statistically significant when $p < 0.05$.

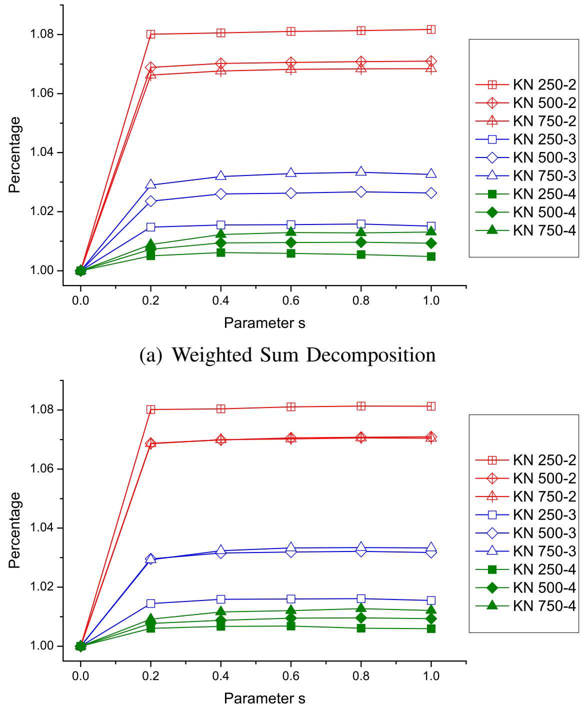
C. Comparison Results

The result of the experiment is shown in Table I. Results significantly better are highlight in bold. Table I shows that the final external population obtained by s-MEDA/D is better than that obtained by MOEA/D, in terms of both C-metric and hypervolume metric except instance KN-250-2 in which s-MEDA/D is slightly worse than MOEA/D in hypervolume metric. For KN-250-2, KN-250-3 and KN-250-4, the mean values of C(s-MEDA/D, MOEA/D) using both decomposition approaches are around 85% while the mean values of C(MOEA/D, s-MEDA/D) are less than 10%. Specifically, for instances of which the item number is 750, the mean values of C(s-MEDA/D, MOEA/D) are greater than 99.5% and the mean values of C(MOEA/D, s-MEDA/D) are zero. That means most solutions obtained by s-MEDA/D could not dominated by any solution found by MOEA/D. But all solutions found by MOEA/D are dominated by some found by s-MEDA/D. From Table I,we could also see that as the size of items increases, the mean values of C(s-MEDA/D, MOEA/D) increase. From the results of hypervolume metric we could see that, on all instances except KN-250-2, s-MEDA/D was statistically significantly superior to MOEA/D.

Fig. 1 plots approximations containing all nondominated solutions found by three algorithms in 30 runs for bi-objective instances. The three figures above are results obtained using weighted sum decomposition and figures below are that using Tchebycheff decomposition. These figures clearly indicate that for all the bi-objective instances, s-MEDA/D is more effective than MOEA/D and MEDA/D. They also indicate that as the size of instance increases, the advantage of s-MEDA/D over its competitors increase. Because the MEDA/D is a special case of s-MEDA/D when the parameter s is zero, these figures can partly confirm that the proposed reproduction operator is able to improve the performance.

D. Influence of Parameter

This subsection reports the influence of the main parameter of the proposed algorithm. As discussed in section III-B, the parameter s has clear physical meaning, which can directly



(a) Weighted Sum Decomposition

(b) Tchebycheff Decomposition

Fig. 2. The influence of parameter s. The y-axis is the relative values to the results obtained when $s = 0$.

reflect the number of changed bits of new solution when all the neighbouring solutions are the same in expectation. But s should not be too large. Even a small value of s could greatly improve the diversity. For example, when $s = 0.4$ and the size of the problem is $n = 500$, the probability that new solution having at least a bit differing from original ones is

$$p_{change} = 1 - (1 - \frac{0.4}{500})^{500} > 32\% \tag{22}$$

It should be noted that this is extreme case which all neighbouring solutions are same. In general, neighbouring solutions are not the same, and the probability producing a different new solution would become larger.

For each value of s, the algorithm is executed on an instance for 30 independent runs. Here we use hypervolume because it could reflects the performance of an algorithm independent of other algorithms. We consider the situation $s \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Fig. 2 shows how the performance of s-MEDA/D varies over the change of s. Algorithms in Fig. 2(a) use weighted sum decomposition, while that in Fig. 2(b) use Tchebycheff decomposition. We select the origin as the reference point in calculating the hypervolume metric. The range of the result is very large (from 1E+7 to 1E+18) and normalization is carried out. All results are divided by the result from the same instance when $s = 0$, that is exactly the MEDA/D. The y-axis means the relative values expressed in percentages. Therefore, for all instances, the results are begin with 1.0. First, we can see from the figures that the proposed reproduction operator is effective. No matter what instance is, the hypervolume metric of s-MEDA/D is greater than that of MEDA/D. For instances with bi-objectives, the hypervolume increases as the s increases. For instances with three objectives, s-MEDA/D attains its maximum performance when $s = 0.8$ and overlarge s results in performance regression. Overall, s-MEDA/D is not sensitivity to its parameter when $0.2 \le s \le 1.0$.

V. Conclusion and Future Work

In this paper, we reported a new algorithm based on MOEA/D framework and estimation of distribution algorithm method, referred as s-MEDA/D. In s-MEDA/D, we proposed a new reproduction operator containing a small value which could prevent the probability vector from convergence. We have tested the s-MEDA/D on nine instances of the 0/1 multi-objective knapsack problem. Experimental results show that on average, s-MEDA/D outperforms the established algorithm, MOEA/D. A study on the parameter s of s-MEDA/D has been carried out. The results indicate s-MEDA/D generate high quality solutions than MEDA/D and it is found to be insensitive to the parameter s. The proposed algorithm could handle only binary coded problems and the probabilistic vector does not capture relationship between variables. In the future, we will incorporate more complex probability model into s-MEDA/D. And also, we would like to apply s-MEDA/D to more benchmark problems and real-world problems.

Acknowledgment

The authors would like to thank Dr. Qingfu Zhang and Dr. Hui Li for making their MOEA/D codes available. This work is supported by National Basic Research Program of China (973 Program) (Grant No: 2012CB316301), National Natural Science Foundation of China (Grant No: 61175110), National S&T Major Projects of China (Grant No: 2011ZX02101-004) and National Banking Information Technology Risk Management Projects of China.

References

[1] C. A. Coello Coello, "Evolutionary multi-objective optimization: a historical view of the field," *Computational Intelligence Magazine, IEEE*, vol. 1, no. 1, pp. 28–36, 2006.

[2] 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.

[3] C. Erbas, S. Cerav-Erbas, and A. D. Pimentel, "Multiobjective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 3, pp. 358–374, 2006.

[4] M. A. Abido, "Multiobjective evolutionary algorithms for electric power dispatch problem," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 3, pp. 315–329, 2006.

[5] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

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

[7] M. Pelikan, D. E. Goldberg, and F. G. Lobo, “A survey of optimization by building and using probabilistic models,” *Computational optimization and applications*, vol. 21, no. 1, pp. 5–20, 2002.

[8] Y. Li, A. Zhou, and G. Zhang, “A decomposition based estimation of distribution algorithm for multiobjective knapsack problems,” in *Natural Computation (ICNC), 2012 Eighth International Conference on*. IEEE, 2012, pp. 803–807.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[10] H. Fang, Q. Wang, Y.-C. Tu, and M. F. Horstemeyer, “An efficient non-dominated sorting method for evolutionary algorithms,” *Evolutionary computation*, vol. 16, no. 3, pp. 355–384, 2008.

[11] Y. Yusoff, M. S. Ngadiman, and A. M. Zain, “Overview of nsga-ii for optimizing machining process parameters,” *Procedia Engineering*, vol. 15, pp. 3978–3983, 2011.

[12] A. Santiago, H. J. F. Huacuja, B. Dorronsoro, J. E. Pecero, C. G. Santillan, J. J. G. Barbosa, and J. C. S. Monterrubio, “A survey of decomposition methods for multi-objective optimization,” in *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*. Springer, 2014, pp. 453–465.

[13] 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.

[14] N. Beume, B. Naujoks, and M. Emmerich, “Sms-emoa: Multiobjective selection based on dominated hypervolume,” *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[15] S. Baluja, “Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning,” DTIC Document, Tech. Rep., 1994.

[16] G. R. Harik, F. G. Lobo, and D. E. Goldberg, “The compact genetic algorithm,” *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 287–297, 1999.

[17] H. Mühlenbein and G. Paaß, “From recombination of genes to the estimation of distributions i. binary parameters,” in *Parallel Problem Solving from Nature ł PPSN IV*, ser. Lecture Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 1996, vol. 1141, pp. 178–187.

[18] K.-H. Han and J.-H. Kim, “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 6, pp. 580–593, 2002.

[19] M. D. Platel, S. Schliebs, and N. Kasabov, “Quantum-inspired evolutionary algorithm: A multimodel eda,” *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 6, pp. 1218–1232, December 2009.

[20] B. Wang, H. Xu, and Y. Yuan, “Quantum-inspired evolutionary algorithm with linkage learning,” in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 2467–2474.

[21] J. S. De Bonet, C. L. Isbell, P. Viola *et al.*, “Mimic: Finding optima by estimating probability densities,” *Advances in neural information processing systems*, pp. 424–430, 1997.

[22] G. Harik, “Linkage learning via probabilistic modeling in the ecga,” *Urbana*, vol. 51, no. 61, p. 801, 1999.

[23] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz, “Linkage problem, distribution estimation, and bayesian networks,” *Evolutionary computation*, vol. 8, no. 3, pp. 311–340, 2000.

[24] A. Johnson and J. Shapiro, “The importance of selection mechanisms in distribution estimation algorithms,” in *Artificial Evolution*. Springer, 2002, pp. 91–103.

[25] K.-H. Han and J.-H. Kim, “Quantum-inspired evolutionary algorithms with a new termination criterion, H_ϵ gate, and two-phase scheme,” *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 2, pp. 156–169, 2004.

[26] J. Yang, H. Xu, Y. Cai, and P. Jia, “Effective structure learning for eda via l1-regularizedbayesian networks,” in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010, pp. 327–334.

[27] A. Zhou, Q. Zhang, and G. Zhang, “A multiobjective evolutionary algorithm based on decomposition and probability model,” in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.

[28] V. A. Shim, K. Tan, and C. Cheong, “A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 5, pp. 682–691, 2012.

[29] A. Zhou, F. Gao, and G. Zhang, “A decomposition based estimation of distribution algorithm for multiobjective traveling salesman problems,” *Computers & Mathematics with Applications*, vol. 66, no. 10, pp. 1857–1868, 2013.

[30] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[31] A. Zhou, Q. Zhang, and Y. Jin, “Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm,” *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 5, pp. 1167–1189, 2009.

[32] K. Miettinen, *Nonlinear multiobjective optimization*. Springer, 1999, vol. 12.

[33] A. P. Wierzbicki, “The use of reference objectives in multiobjective optimization,” in *Multiple criteria decision making theory and application*. Springer, 1980, pp. 468–486.

[34] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990.

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

[36] E. Zitzler and L. Thiele, “Multiobjective optimization using evolutionary algorithmsła comparative case study,” in *Parallel problem solving from naturełPPSN V*. Springer, 1998, pp. 292–301.