Proceeding of the 11th World Congress on Intelligent Control and Automation
Shenyang, China, June 29 - July 4 2014

# A Probability Model based Evolutionary Algorithm with Priori and Posteriori Knowledge for Multiobjective Knapsack Problems

Yang Li
Department of Computer Science
and Technology
East China Normal University
Shanghai, China, 200241
Email: yangli@student.ecnu.edu.cn

Aimin Zhou
Department of Computer Science
and Technology
East China Normal University
Shanghai, China, 200241
Email: amzhou@cs.ecnu.edu.cn

Guixu Zhang
Department of Computer Science
and Technology
East China Normal University
Shanghai, China, 200241
Email: gxzhang@cs.ecnu.edu.cn

***Abstract*—Most evolutionary algorithms utilize the posteriori knowledge learned from the running process to guide the search. It is arguable that the priori knowledge about the problems to tackle can also play an important role in problem solving. To demonstrate the importance of both priori and posteriori knowledge, in this paper, we proposes a *decomposition based estimation of distribution algorithm with priori and posteriori knowledge (MEDA/D-PP)* to tackle *multiobjective knapsack problems (MOKPs)*. In MEDA/D-PP, an MOKP is decomposed into a number of single objective subproblems and those subproblems are optimized simultaneously. A probability model, which incorporates both priori and posteriori knowledge, is built for each subproblem to sample new trail solutions. The proposed method is applied to a variety of test instances and the experimental results show that the proposed algorithm is promising. It is demonstrated that priori knowledge can improve the search ability of the algorithm and posteriori knowledge is helpful to guide the search.**

## I. Introduction

Knapsack problem is widely studied in the field of combinational optimization. *Multiobjective knapsack problem (MOKP)* is one of its variants. In an MOKP, $m$ knapsacks and $n$ items are given and each of the items has different weights and profits in different knapsacks. The goal is to select as many items into the knapsacks as possible so that each knapsack gets as much profit as possible. Note that once an item is chosen, it will be placed in all of the knapsacks. Many real world applications can be modeled as MOKP, such as cargo loading, cutting stock, bin-packing, budget control and financial management [1]. MOKP is also a good performance benchmark to assess algorithms for *multiobjective optimization problems (MOPs)*.

*Multiobjective evolutionary algorithm (MOEA)* is a promising algorithm to tackle multiobjective optimization problems since it can obtain a set of optimal solutions over a single run. A variety of MOEAs have been proposed for tackling MOPs. Among them, *decomposition based multiobjective evolutionary algorithm (MOEA/D)* [2] has shown promising results. The basic idea of MOEA/D is to decompose an MOP into a set of single objective subproblems, so that these subproblems can be solved simultaneously to obtain the solutions of the MOP. Recently, many works have been done to solve MOKP within the framework of MOEA/D. Kafafy et al. [3] used MOKP as the benchmark to study the effect of the hybridization of different metaheuristics within MOEA/D framework. A new version of MOEA/D with uniform design for solving MOKP is proposed in [4]. Ishibuchi et al. [5] examined the relation between the neighborhood size and the performance of MOEA/D on MOKP instances and two different neighborhoods, one for local mating and the other for local replacement, are introduced.

It advocates that the integration of machine learning techniques into heuristics and self-tunes algorithm behaviors based on online information collected during the search [6]. *Estimation of distribution algorithms (EDAs)* [7] are among the promising evolutionary computation paradigms which adopt this principle. Unlike traditional *evolutionary algorithms (EAs)*, which use crossover and/or mutation operators to generate new solutions, EDAs explicitly extract the population distribution information by probabilistic models and then sample new solutions from the models. The EDA method is used as the reproduction operator in our previous work [8] which builds an probability model based on the neighboring solutions of subproblems to do the reproduction within the MOEA/D framework.

Usually, most MOEAs use problem-independent methods to search within the domain of the problems. But as a specific problem, knapsack problem has its own properties, one of which is that each item has a pseudo utility considering its profit and weight. This special property is seen as priori knowledge in this paper. Priori knowledge can fasten the search speed and improve the search quality if it is used properly. Many works have been done to find and use priori knowledge to improve the performances of algorithms, such as [6], [9]. As it usually happens in heuristics, the more priori knowledge about a problem is used, the better the results it would get. In this paper, we propose an algorithm called MEDA/D-PP which uses an EDA model to incorporate both

978-1-4799-5825-2/14/$31.00 ©2014 IEEE

priori and posteriori knowledge within MOEA/D framework to solve MOKP.

The rest of the paper is organized as follows. Section II describes MOKP in details. The priori knowledge of MOKP used is given. The proposed MEDA/D-PP is presented in Section III. In Section IV, we compare the proposed algorithm with the MEDA/D without using priori knowledge (MEDA/D-N) and an ant colony optimization (ACO) based algorithm called MOEA/D-ACO. Section V concludes this paper.

## II. MOKP AND ITS PRIORI KNOWLEDGE

Mathematically, an MOKP can be formulated as

$$\begin{aligned} \max \quad & F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \\ & f_i(x) = \sum_{j=1}^{n} p_{ij} x_j, i = 1, 2, .., m \\ \text{s.t.} \quad & \sum_{j=1}^{n} w_{ij} x_j \leq c_i, i = 1, 2, \ldots, m, \\ & x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n \end{aligned}$$

where $m$ is the number of objectives, $n$ is the number of items, $p_{ij} \geq 0, w_{ij} \geq 0$ are the profit and weight of the $j^{th}$ item in the $i^{th}$ knapsack respectively, $c_i$ is the capacity of the $i^{th}$ knapsack, $x$ is an $n$-dimensional binary vector with each element indicating the corresponding item is chosen ($x_j = 1$) or not ($x_j = 0$).

The priori knowledge of an MOKP used in this paper is from *multidimensional knapsack problem (MKP)*. An MKP can be formulated as

$$\begin{aligned} \max \quad & F(x) = \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} w_{ij} x_j \leq c_i, i = 1, 2, \ldots, m, \\ & x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n \end{aligned} .$$

Contrast to MOKP, in an MKP, each item has only one profit value for all knapsacks.

The pseudo utility is used as the priori knowledge of the MKP, and there are several ways to calculate it [10][11][12]. In this paper, the surrogate duality approach of [11] is adopted to determine the pseudo utility. The surrogate relaxation problem of the MKP (denoted by SR-MKP) can be defined as:

$$\begin{aligned} \max \quad & F(x) = \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} (\sum_{i=1}^{m} \gamma_i w_{ij}) x_j \leq \sum_{i=1}^{m} \gamma_i c_i, \\ & x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n \end{aligned}$$

where $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$ is a set of surrogate multipliers (or weights) of some positive real numbers. An upper bound on the original MKP can be obtained by solving SR-MKP [13]. Since SR-MKP is essentially a single constraint knapsack problem, the pseudo utility for the $j^{th}$ variable of the solution of the MKP, based on the surrogate constraint coefficient, is simply $u_j = p_j / \sum_{i=1}^{m} \gamma_i w_{ij}$ [14].

In this paper, we decompose an MOKP into $N$ subproblems, whose solutions are then the solutions of the MOKP, using $N$ weight vectors $\lambda^1, \ldots, \lambda^N$ where $\lambda^k = (\lambda^k_1, \ldots, \lambda^k_m)$. Each $\lambda$ can define an MKP. The $k^{th}$ MKP which is defined by $\lambda^k$ can be formulated as:

$$\begin{aligned} \max \quad & F(x) = \sum_{i=1}^{m} \lambda^k f_i(x) = \sum_{j=1}^{n} (\sum_{i=1}^{m} \lambda^k p_{ij}) x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} w_{ij} x_j \leq c_i, i = 1, 2, \ldots, m, \\ & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, \\ & x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n \end{aligned}$$

. By solving the surrogate relaxation problems of the $N$ MKPs, we can obtain an upper bound on the original MOKP and the pseudo utility for the $j^{th}$ variable of the $k^{th}$ subproblem $u^k_j = \sum_{i=1}^{m} \lambda^k p_{ij} / \sum_{i=1}^{m} \gamma_i w_{ij}$, which will be used as the priori knowledge of MOKP in this paper.

## III. THE PROPOSED MEDA/D WITH PRIORI KNOWLEDGE

In this section, we firstly describe the general MEDA/D-PP for solving MOKP and then introduce how to incorporate priori and posteriori knowledge with the EDA method to do reproduction.

### A. The Proposed MEDA/D-PP for MOKP

The proposed MEDA/D-PP shares the same framework with MOEA/D. The main difference between them is that the traditional EA based reproduction operator is replaced by an EDA based reproduction operator. MEDA/D-PP first decomposes an MOKP into $N$ single objective subproblems by $N$ preselected weight vectors $\lambda^1, \ldots, \lambda^N$. Subproblem $k$ has $T$ neighbors $B(k)$ whose weight vectors are the $T$ closest to $\lambda^k$. In this paper, we use the following Tchebycheff approach to define the subproblems:

$$\min g(x|\lambda^k, z^*) = \max_{1 \leq j \leq m} \lambda^k_j |f_j(x) - z^*_j|$$

where $\lambda^k = (\lambda^k_1, \cdots, \lambda^k_m)^T$ is a weight vector with the $k^{th}$ subproblem, $z^* = (z^*_1, \cdots, z^*_m)$ is a reference point. It is clear that the subproblems are differentiated by the weight vectors. If two vectors are close to each other, the corresponding subproblems should be similar and their optima should also be close in both the decision and objective spaces in most cases.

At each generation, an MEDA/D-PP with $N$ subproblems maintains the following:

- each subproblem is with
  - $S^k$, an EDA model,
  - $U^k$, a predetermined priori knowledge,
  - $x^k$, a decision objective vector and its objective vector $F^k = F(x^k)$;
- $z^* = (z^*_1, \cdots, z^*_m)$, a reference point which is an ideal point, i.e., $z^*_i = \max\{f_i(x)\}$;
- $EP$, an external population which stores the nondominated solutions found so far. In practice, $EP$ may be more useful for discrete MOPs than for continuous MOPs.

Algorithm 1 illustrates how MEDA/D-PP works. Details are shown in our previous work [8]. Next section describes the offspring reproduction in Line **6** and probability model updating in Line **11**.

**Algorithm 1:** Procedure of MEDA/D-PP

```
Set $EP = \emptyset$;
Initialize a set of subproblems: $\lambda^k$, $B^k$, $S^k$, $U^k$, $k = 1, \cdots, N$;
Initialize the reference point $z^*$ as $z_i^* = 0$, $i = 1, \cdots, m$;
while not terminate do
    for $k = 1 : N$ do
        Generate a solution $y$ for the $k^{th}$ subproblem using $S^k$ in a probabilistic way;
        Update $z^*$ with $y$;
        Update the $k^{th}$ subproblem's neighboring solutions with $y$;
        Update $EP$ with $y$;
    end
    Update all the subproblems' EDA models;
end
```

### B. Probability Model Updating and Offspring Reproduction

In this paper, the solution $x$ of an MOKP is a binary vector. In the proposed MEDA/D-PP, two kinds of knowledge are used to generate offspring solutions. The EDA model $S^k$ for the $k^{th}$ subproblem is composed of priori knowledge $U^k$ and posteriori knowledge $V^k$. $S^k, U^k$ and $V^k$ are computed as follows.

$U^k = (u_1^k, \cdots, u_n^k)$ is the priori knowledge of the $k^{th}$ subproblem of the decomposed MOKP:

$$u_j^k = \frac{\sum_{l=1}^{m} \lambda_l^k p_{l,j}}{\sum_{l=1}^{m} \gamma_l^k w_{l,j}}$$

where $j = 1, 2, \cdots, n$, which is described in Section II.

$V^k = (v_1^k, \cdots, v_n^k)$ is the posteriori knowledge based on neighboring solutions of the $k^{th}$ subproblem:

$$v_j^k = \frac{1}{T} \sum_{t=1}^{T} x_j^{k_t}$$

where $j = 1, 2, \cdots, n$. $v_j^k \in [0, 1]$ and it denotes the desirability that the $j^{th}$ element of $x^k$ is set to 1. We can find that the knowledge learned from the neighborhood solutions is used to build the model. The reason is that the neighboring solutions should be similar with each other and share similar patterns.

$S^k = (s_1^k, \cdots, s_n^k)$ is computed with $U^k$ and $V^k$:

$$s_j^k = (v_j^k)^\alpha (u_j^k)^\beta$$

where $\alpha$ and $\beta$ are two parameters to balance the priori knowledge and the posteriori knowledge.

Algorithm 2 is used to construct a feasible solution for the $k^{th}$ subproblem using $S^k$. Here $rand()$ generates a random real number between 0 and 1, and $r$ is used to balance the two reproduction strategies.

**Algorithm 2:** Sampling procedure

```
Randomly select an item;
while there are items that do not violate the constraints left do
    if $rand() < r$ then
        Select the item with the largest $s$ value;
    else
        Select one item by the roulette wheel selection according to items' $s$ values;
    end
end
```

## IV. Experimental Results

In this section, we compare MEDA/D-PP with two algorithms. Then the sensitivity of parameter $\beta$ in MEDA/D-PP is investigated.

### A. Comparison Algorithms

We choose the MEDA/D without using priori knowledge (MEDA/D-N) [8] and MOEA/D-ACO [6] as the comparison algorithms to test our newly proposed algorithm.

In MEDA/D-N, the probability models based on neighboring solutions are directly used as EDA models to generate offspring solutions for the subproblems. New solutions might be infeasible, so a greedy repair method [15] is applied to make them feasible.

MOEA/D-ACO proposed in [6] works within the same framework as MEDA/D-PP does. To our best knowledge, MOEA/D-ACO can obtain the best non-dominated solutions for MOKP than other state-of-the-art algorithms. In MOEA/D-ACO, the MOKP is decomposed into a set of subproblems, and an ant is responsible for solving one subproblem. Then the ants are divided into several groups. Similar to EDA, ACO is also an probability model since it learns information on one group of ants and then use learned knowledge (i.e. pheromone matrix), priori knowledge (i.e. heuristic information matrix) and ants' solutions for constructing new offspring solutions for the ants in the group.

Here, MEDA/D-N is used to investigate the effect of priori knowledge on the performance of MEDA/D-PP. And MOEA/D-ACO is used to test the effect of EDA model on the performance of MEDA/D-PP.

### B. Experimental Settings

Nine test instances from [16] with 250, 500, or 750 items, and 2, 3, or 4 knapsacks (objectives) are used as the benchmark. All the three algorithms stop after 500 generations. To compare the results of different algorithms, each test instance is optimized over 30 independent runs. Table I gives the parameter settings of the three algorithms, and all these parameters of MOEA/D-ACO are the same as in [6].

TABLE I
PARAMETER SETTINGS OF THE THREE ALGORITHMS

| parameter | MEDA/D-N | MEDA/D-PP | MOEA/D-ACO |
|---|---|---|---|
| T | 10 | 10 | 10 |
| $\alpha$ | - | 1 | 1 |
| $\beta$ | - | 10 | 10 |
| r | - | 0.9 | 0.9 |
| $\rho$ | - | - | 0.95 |
| $\varepsilon$ | - | - | $1/(2n)$ |
| $\Delta$ | - | - | $0.05 \times \tau_{max}$ |

### C. Performance Metric

The hypervolume metric [16] is used to assess the performance of the algorithms in our experimental studies.

The hypervolume metric $I_H(P,z)$ denotes the volume covered by a set $P$ and a reference point $z$. $I_H(P,z)$ can measure both the diversity and convergence of a set. To have a high value of $I_H(P,z)$ for a given reference point $z$, the approximation set $P$ must be as close to the true PF and diverse as possible.

In the experiments, $z$ is set as $z = 7.0 \times 10^3(1,\cdots,m)$ for problems with 250 items, $z = 1.5 \times 10^4(1,\cdots,m)$ for problems with 500 items, $z = 2.2 \times 10^4(1,\cdots,m)$ for problems with 750 items.

### D. Comparison Results

Table II presents the mean and standard deviation of the hypervolume values of the final approximations obtained by each algorithm among 30 runs for each test instance. We can see from Table II that MEDA/D-PP and MOEA/D-ACO perform better than MEDA/D-N on all the test instances. It also shows that the final approximations obtained by MEDA/D-PP are a little worse than those obtained by MOEA/D-ACO.

Fig. 1 shows the run time performances of MEDA/D-N and MEDA/D-PP on 2 and 3 objectives test instances. It shows that convergence speed of MEDA/D-PP is much faster than that of MEDA/D-N on all the test instances. We can also find that MEDA/D-PP can produce high quality solutions at the initial stage of the search. The reason might be that the use of priori knowledge makes contributions.

Fig. 2 plots the comparison between the final approximations of MEDA/D-N and MEDA/D-PP with the lowest hypervolume value among 30 runs for each 2-objective test instance. The 'Relaxed PF' is the upper approximations obtained by Jaszkiewicz [15]. From Fig. 2, we can find that there is significant difference between the final approximations by MEDA/D-N and MEDA/D-PP, MEDA/D-PP being much better than MEDA/D-N.

Fig. 3 plots the comparison between the final approximations of MOEA/D-ACO and MEDA/D-PP with the lowest hypervolume value among 30 runs for each 2-objective test instance. Fig. 3 shows that the final approximations by MEDA/D and MOEA/D-ACO are very close to the relaxed PF and there is little difference between the final approximations by MEDA/D and MOEA/D-ACO. It is difficult to distinguish them visually. Thus the proposed algorithm can obtain similar performance as MOEA/D-ACO while uses smaller number of parameters (see Table I). We can also see from Fig. 3 that the numbers of solutions in the two ends of the final approximations by MEDA/D-PP are smaller than those by MOEA/D-ACO and this might explain why MEDA/D is slightly worse than MOEA/D-ACO in terms of hypervolume metric statistics.

TABLE II
HYPERVOLUME METRIC VALUES OF THE APPROXIMATIONS FOUND BY MEDA/D-N, MEDA/D-PP AND MOEA/D-ACO ON THE NINE TEST INSTANCES OVER 30 RUNS

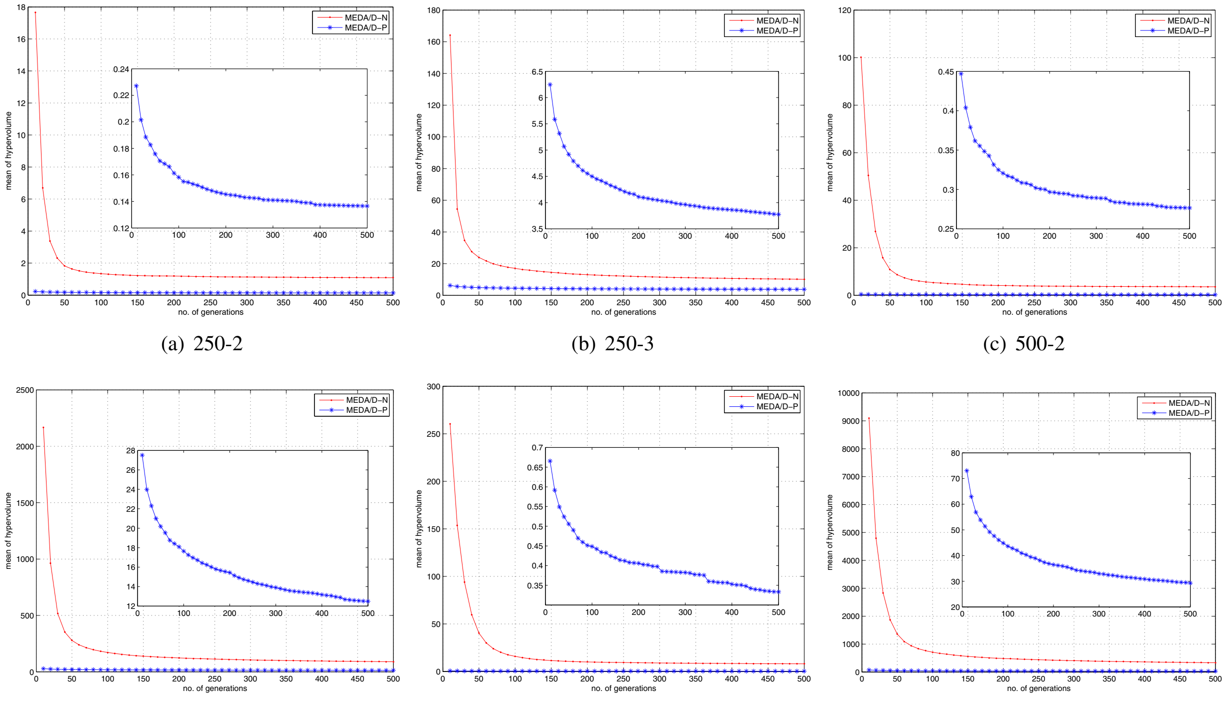
| instance | MEDA/D-N | | MEDA/D-PP | | MOEA/D-ACO | |
|---|---|---|---|---|---|---|
| $n-m$ | mean | std. | mean | std. | mean | std. |
| $250-2$ | 7.525 | 0.08 | 7.717 | 0.00 | 7.724 | 0.00 |
| $250-3$ | 13.81 | 0.04 | 14.94 | 0.00 | 14.97 | 0.00 |
| $250-4$ | 12.68 | 0.09 | 17.27 | 0.01 | 17.48 | 0.01 |
| $500-2$ | 23.45 | 0.18 | 24.13 | 0.00 | 24.14 | 0.00 |
| $500-3$ | 56.14 | 0.38 | 66.82 | 0.02 | 67.04 | 0.02 |
| $500-4$ | 42.41 | 1.03 | 93.64 | 0.16 | 95.64 | 0.07 |
| $750-2$ | 52.36 | 0.30 | 54 | 0.00 | 54 | 0.00 |
| $750-3$ | 328.5 | 1.37 | 374.4 | 0.07 | 375.3 | 0.05 |
| $750-4$ | 508 | 7.57 | 1030 | 0.96 | 1052 | 0.66 |

### E. Sensitivity of Algorithm Parameter $\beta$

In this section, we study the sensitivity of $\beta$ which is the major control parameter of the reproduction procedure in MEDA/D-PP. We test $\beta = 1, 5, 10, 15, 20$ on the nine test instances. The final mean hypervolume metric values are shown in Table III and the ranks of the $\beta$s on each instance are calculated to find the overall ranks of the $\beta$s. Table III shows that a middle value of $\beta$ is preferable since a low value ($\beta = 1$) performs the worst among the $\beta$s on most of the test instances while a high value ($\beta = 20$) performs badly on some test instances such as 500-2 and 750-2. We can infer that if the priori knowledge is too little, the algorithm will perform badly because it will mostly depend on the posteriori knowledge whose effect depends on the quality of the solutions. On the other hand, if too much priori knowledge is used, the algorithm will also perform badly since the posteriori knowledge will make little sense during the search.

## V. CONCLUSIONS

This paper proposed an algorithm using EDA with priori and posteriori knowledge within the MOEA/D framework to tackle multiobjective knapsack problems. The main idea is to use priori knowledge to improve the quality of the solutions and posteriori knowledge to guide the search. The priori knowledge of each MOKP instance is predefined and obtained before the search. In the proposed reproduction procedure, probability models are built using priori knowledge and posteriori knowledge which is from the neighboring solutions of



(a) 250-2

(b) 250-3

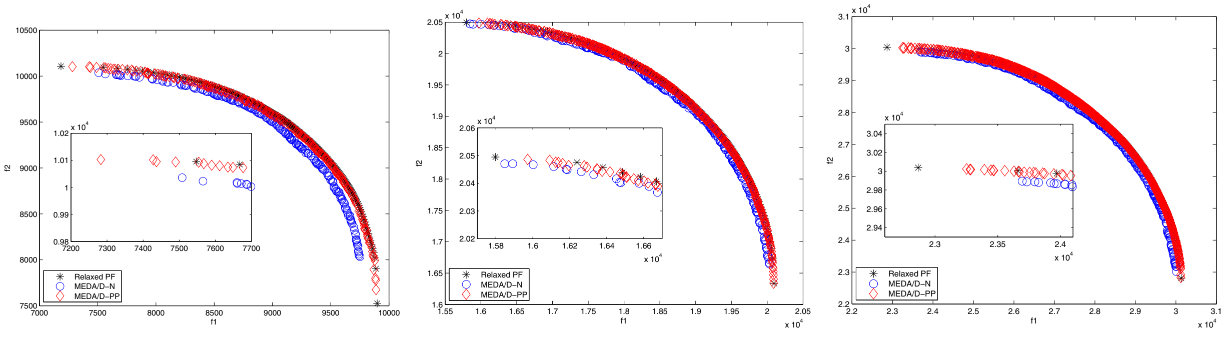
(c) 500-2

(d) 500-3

(e) 750-2

(f) 750-3

Fig. 1. The mean of hypervolume-metric values versus the number of generations by the two algorithms over 30 runs, the red '.' is with MEDA/D-N and the blue '*' is with MEDA/D-PP

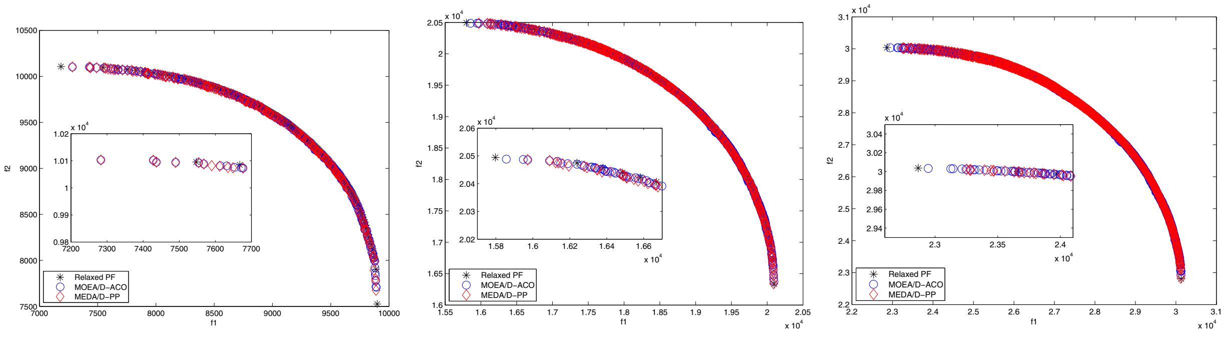


(a) 250-2

(b) 500-2

(c) 750-2

Fig. 2. The best runs obtained by the two algorithms for the three 2-objective instances according to the hypervolume metric. The black 'star' is with the relaxed front, the blue 'circle' is with MEDA/D-N and the red 'diamond' is with MEDA/D-PP



(a) 250-2

(b) 500-2

(c) 750-2

Fig. 3. The best runs obtained by the two algorithms for the three 2-objective instances according to the hypervolume metric. The black 'star' is with the relaxed front, the blue 'circle' is with MOEA/D-ACO and the red 'diamond' is with MEDA/D-PP

TABLE III

HYPERVOLUME METRIC VALUES OF THE APPROXIMATIONS FOUND BY THE ALGORITHMS WITH $\beta = 1, 5, 10, 15, 20$ ON THE NINE TEST INSTANCES OVER 30 RUNS AND THE RANKS THE $\beta$S

| instance | $\beta = 1$ | $\beta = 5$ | $\beta = 10$ | $\beta = 15$ | $\beta = 20$ |
|---|---|---|---|---|---|
| $n - m$ | mean(rank) | mean(rank) | mean(rank) | mean(rank) | mean(rank) |
| $250 - 2$ | 0.1605(4) | 0.1769(5) | 0.1362(2) | **0.1278**(1) | 0.1435(3) |
| $250 - 3$ | 5.1778(5) | 3.8777(2) | **3.8487**(1) | 3.9633(3) | 4.0769(4) |
| $250 - 4$ | 73.2475(5) | 63.3849(4) | 50.3389(2) | 56.1183(3) | **50.0254**(1) |
| $500 - 2$ | 0.5113(4) | 0.3200(3) | **0.2511**(1) | 0.2810(2) | 0.6371(5) |
| $500 - 3$ | 128.3453(5) | 13.0680(4) | 11.2543(3) | **9.1892**(1) | 9.2386(2) |
| $500 - 4$ | 1022.1355(5) | 543.1491(4) | **344.0319**(1) | 363.8055(2) | 364.0735(3) |
| $750 - 2$ | 0.6806(4) | 0.3872(3) | **0.1929**(1) | 0.2143(2) | 1.8668(5) |
| $750 - 3$ | 62.3365(5) | 35.5267(4) | 25.9950(2) | 26.9766(3) | **21.6238**(1) |
| $750 - 4$ | 6037.1583(5) | 3107.8742(4) | 1805.5381(3) | 1613.5445(2) | **1197.4751**(1) |
| rank statistics | 4.6667(5) | 3.6667(4) | **1.7778**(1) | 2.1111(2) | 2.7778(3) |

the subproblems. At each generation, for each subproblem, an probability model is used to produce a new solution. The newly produced solutions are then evaluated and used to update the neighborhood of the subproblems. The proposed MEDA/D with priori and posteriori knowledge (MEDA/D-PP) was compared with MEDA/D without priori knowledge (MEDA/D-N) and MOEA/D-ACO.

Experimental results indicates that MEDA/D-PP outperforms MEDA/D-N, which means that priori knowledge can improve the search ability. MEDA/D-PP has similar performance with MOEA/D-ACO, which shows that EDA is an simple yet efficient method since it has fewer parameters than ACO does whose performance on solving MOKP is the best in the literature.

There are a variety of directions worth exploring. For example, other priori knowledge of MOKP to can be utilized to improve the search. Another direction is combine the priori knowledge with other heuristic techniques.

## ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No.61273313).

## REFERENCES

[1] S. Fidanova, *Large Scale Scientific Computing*, ser. Lecture Notes in Computer Science. Springer, 2008, vol. 4818, ch. Probabilistic Model of Ant Colony Optimization for Multiple Knapsack Problem, pp. 545–552.

[2] Z. Qingfu and L. Hui, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[3] A. Kafafy, A. Bounekkar, and S. Bonnevay, “Hybrid metaheuristics based on MOEA/D for 0/1 multiobjective knapsack problems: A comparative study,” in *2012 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2012, pp. 1–8.

[4] Y. Tan and Y. Jiao, “MOEA/D with uniform design for solving multiobjective knapsack problems,” *Journal of Computers*, vol. 8, no. 2, pp. 302–307, 2013.

[5] H. Ishibuchi, N. Akedo, and Y. Nojima, “Relation between neighborhood size and MOEA/D performance on many-objective problems,” in *Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 459–474.

[6] L. Ke, Q. Zhang, and R. Battiti, “MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1845–1859, 2013.

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

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

[9] 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.

[10] R. Loulou and E. Michaelides, “New greedy-like heuristics for the multidimensional 0-1 knapsack problem,” *Operations Research*, vol. 27, no. 6, pp. 1101–1114, 1979.

[11] H. Pirkul, “A heuristic solution procedure for the multiconstraint zero-one knapsack problem,” *Naval Research Logistics (NRL)*, vol. 34, no. 2, pp. 161–172, 1987.

[12] Y. Toyoda, “A simplified algorithm for obtaining approximate solutions to zero-one programming problems,” *Management Science*, vol. 21, no. 12, pp. 1417–1427, 1975.

[13] B. Gavish and H. Pirkul, “Efficient algorithms for solving multi-constraint zero-one knapsack problems to optimality,” *Mathematical Programming*, vol. 31, no. 1, pp. 78–105, 1985.

[14] P. C. Chu and J. E. Beasley, “A genetic algorithm for the multidimensional knapsack problem,” *Journal of heuristics*, vol. 4, no. 1, pp. 63–86, 1998.

[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. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.