Evolutionary Multi-objective Optimization Algorithms with Probabilistic Representation Based on Pheromone Trails

Hui Li, Dario Landa-silva, and Xavier Gandibleux

Abstract— **Recently, the research on quantum-inspired evolutionary algorithms (QEA) has attracted some attention in the area of evolutionary computation. QEA use a probabilistic representation, called Q-bit, to encode individuals in population. Unlike standard evolutionary algorithms, each Q-bit individual is a probability model, which can represent multiple solutions. Since probability models store global statistical information of good solutions found previously in the search, QEA have good potential to deal with hard optimization problems with many local optimal solutions. So far, not much work has been done on evolutionary multi-objective (EMO) algorithms with probabilistic representation. In this paper, we investigate the performance of two state-of-the-art EMO algorithms - MOEA/D and NSGA-II, with probabilistic representation based on pheromone trails, on the multi-objective travelling salesman problem. Our experimental results show that MOEA/D and NSGA-II with probabilistic presentation are very promising in sampling high-quality offspring solutions and in diversifying the search along the Pareto fronts.**

I. Introduction

Evolutionary algorithms are a class of very popular optimization search techniques for tackling hard optimization problems from real-world applications. To design an effective evolutionary algorithm, the choice of proper solution representation and the development of appropriate search operators must be considered. This is because representation of solutions could affect some properties of the search landscap, such as redundancy, neighborhood structure, and ruggedness [1]. Some well-known examples of solution representations include: binary, real-value vector, messy encodings, permutation, and tree structures.

In the past two decades, evolutionary multi-objective (EMO) algorithms have attracted a lot of interest from researchers [2][3]. Fitness assignment and diversity maintenance are two major research issues in evolutionary multi-objective optimization. However, little work has been done on designing multi-objective oriented representation of solutions. In [4], a hybrid representation was studied for multi-objective optimization. That scheme uses both binary and real-valued representations to encode individuals in the population. In [5], the influence of binary and order-based representations on the performance of EMO algorithms was investigated for multi-objective knapsack problems.

Currently, the mainstream state-of-the-art EMO algorithms are based on Pareto dominance (e.g. NSGA-II [6] and SPEA-II [7]) or on aggregating functions (e.g. MOGLS [8] and MOEA/D [9]). The research literature shows that Pareto-based EMO algorithms are very effective to tackle continuous multi-objective optimization problems. In contrast, for tackling multi-objective combinatorial optimization problems, EMO algorithms based on aggregating functions seem to be more suitable. The main reason for this is that the later approaches directly use local search to intensify the exploration of promising regions in the search space. Moreover, these algorithms also have advantages in dealing with many-objective optimization problems.

In [10], a quantum-inspired evolutionary algorithm (QEA) was developed to solve combinatorial optimization problems. Unlike other evolutionary algorithms, a QEA uses Q-bit representation to encode individuals. Since each Q-bit individual encodes multiple solutions, a QEA has the ability to provide good diversity in the population. Essentially, each Q-bit individual can be regarded as a simple probabilistic model. Therefore, a QEA is also a kind of multi-model estimation of distribution algorithm (EDA) [11], which samples offspring solutions from probabilistic models. Therefore, there are no crossover and mutation operators used in QEA. In [12], a QEA has been extended to solve the multi-objective knapsack problem.

In order to find a set of diverse non-dominated solutions, many EMO algorithms incorporate mechanisms to encourage diversity of the population in the objective space. However, maintaining diversity in the decision space is also crucial to the performance of EMO algorithms. Based on this, we develop variants of NSGA-II and MOEA/D with probabilistic solution representation. Each probabilistic individual corresponds to one pheromone matrix, which is a probability model commonly used in ant colony optimization (ACO) algorithm [13]. We also compare the performance of NSGA-II and MOEA/D with probabilistic representation to their original versions on the multi-objective travelling salesman problem.

The rest of this paper is organized as follows. Section II introduces some basic definitions in multi-objective optimization. Section III briefly overviews ant colony optimization. Section IV presents the new versions of NSGA-II and MOEA/D with probabilistic presentation. Section V describes the pheromone trails and heuristics in the multi-objective travelling salesman problem. Experimental results are reported and discussed in Section VI while the final section concludes this paper.

Hui Li and Dario Landa-Silva are with the Automated Scheduling, Optimisation and Planning (ASAP) Research Group, in the School of Computer Science, University of Nottingham, United Kingdom (email: hzl@cs.nott.ac.uk and dario.landasilva@nottingham.ac.uk).

Xavier Gandibleux is with the Laboratoire d'Informatique de Nantes Atlantique (LINA), in the Department of Computer Science, University of Nantes, France (email: xavier.gandibleux@univ-nantes.fr).

II. Multi-objective Optimization

Mathematically, a multi-objective optimization problem (MOP) can be stated as:

$$\begin{aligned} \text{minimize} \quad & F(x) = \{f_1(x), \ldots, f_m(x)\} \\ & x \in \Omega \end{aligned} \tag{1}$$

where x is the vector of decision variables, Ω is the feasible area in the decision search space and $F(x) \in R^m$ is a vector of objective functions.

When objective functions are conflicting, solutions in the objective space are not completely ordered. For this reason, the optimal solutions of a MOP are trade-offs among objectives, known as Pareto-optimal solutions. For two objective vectors $u, v \in R^m$, u is said to dominate v if and only if $u_i \leq v_i$ for all $i \in \{1, \ldots, m\}$, and there exists at least one $j \in \{1, \ldots, m\}$ satisfying $u_j < v_j$. A solution $x^* \in \Omega$ is said to be Pareto-optimal if its objective vector is not dominated by the objective vector of any other solution in Ω. The set of objective vectors of all Pareto-optimal solutions is called Pareto-optimal front.

Traditional methods in mathematical programming often convert a MOP into one single objective optimization problem [14]. Among these traditional methods, weighted sum approach is one of the most commonly used. It minimizes the following weighted scalarizing function:

$$g(x, \lambda) = \sum_{i=1}^{m} \lambda_i f_i(x), \tag{2}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)$ is the weight vector with $\lambda_i \geq 0$ for all $i \in \{1, \ldots, m\}$ and $\sum_{i=1}^{m} \lambda_i = 1$. The optimal solution of g is called supported. Under some mild conditions, it is also Pareto-optimal to the MOP in (1).

III. Ant Colony Optimization

Ant colony optimization (ACO) [13] is a constructive meta-heuristic that mimics the behavior of ants when seeking a path between the ant colony and a source of food. In ACO, the priori information (i.e. problem-specific heuristic) on the structure of good solutions and the posteriori information (i.e. artificial pheromone trail) on the probabilistic distribution of previously obtained good solutions are used to sample offspring solutions in promising areas of the search space. Over the years, ACO algorithms have been applied to solve many combinatorial optimization problems [15] [16], such as travelling salesman problem and quadratic assignment problem.

Consider a combinatorial optimization problem (S, f, Ω) with the following characteristics:

- $\mathcal{C} = \{c_1, \ldots, c_{N_c}\}$ is a finite set of solution components;
- each state x of the problem can be represented by a sequence of solution components $\langle c_i, c_j, \ldots, c_k, \ldots \rangle$. The set of all states is denoted by S, the set of candidate solutions.
- $\Omega \subset S$ is the feasible set of the search space satisfying certain constraints and $f(s)$ is the cost function for each candidate solution $s \in S$.

In ACO, artificial ants build solutions by incrementally adding solution components to a partial solution, which is initially empty. This procedure is stopped until a complete solution is generated. The decision rule for accepting one component as new member of the partial solution is determined by both heuristic information and pheromone trail.

The ACO process for constructing solutions can also be understood as moving on a graph $G = (\mathcal{C}, \mathcal{L})$, where the vertices are the components in $\mathcal{C}$ and $\mathcal{L} = \{l_{ij}\}$ is the connection between components. Connections l_{ij} are associated with both pheromone trails τ_{ij}, storing global search information obtained from previous search, and heuristic values η_{ij}, representing problem-specific greediness. Assume that $x_r =< x_{r-1}, c_i >$ is the partial solution constructed so far and c_i is the current node, the next node c_j is chosen from the feasible neighborhood $\mathcal{N}_i$ of node c_i by applying some probabilistic decision rules. Then, the new partial solution x_{r+1} is set to $< x_r, j >$. The decision rule is determined by pheromone trails and heuristic values.

In the past twenty years, many variants of ACO algorithms have been proposed. The main differences among them lie in the design of pheromone representation (i.e. how to define solution components) and probabilistic decision rules. For example, in the first implementation of ACO algorithm for the travelling salesman problem, called ant system (AS) [15], the probability of moving from node c_i to node c_j was defined as:

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i} [\tau_{il}(t)]^{\alpha} [\eta_{il}]^{\beta}} \tag{3}$$

where

- $\eta_{ij} = 1/d_{ij}$ is the priori heuristic value and d_{ij} is the distance between cities i and j;
- α and β are two parameters representing the relative importance of heuristic information and pheromone information.
- $\mathcal{N}_i$ is the feasible neighborhood of city i not visited yet.

When $\alpha = 0$, the closest city is more likely to be selected as the new component of the current partial solution.

Once a complete tour x is generated, the pheromone trails need to be updated in two steps. First, the pheromone trail for each edge is modified by

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t)$$

where $\rho \in (0, 1]$ is the evaporation rate.

Second, for each edge (i, j) belonging to the tour x, the related pheromone trail is updated as follows:

$$\tau_{ij}(t + 1) = \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{4}$$

where $\Delta\tau_{ij}(t) = 1/L(t)(L(t)$ is the length of $x)$.

In [17], one of the successors of AS, called Ant Colony System (ACS), used a different probability decision rule. At each step, the node j with the maximal greediness was chosen with probability $q_0 \in [0, 1]$. That is,

$$j = \text{argmax}_{l \in \mathcal{N}_i} [\tau_{il}(t)]^{\alpha} [\eta_{il}]^{\beta}.$$

The ACS approach uses the same rules as the original AS with probability set to $1 - q_0$.

Like evolutionary algorithms, the application of ACO algorithms for multi-objective optimization has also attracted interest for the research community. In [18], an empirical analysis of multi-objective ACO algorithms was conducted on bi-objective TSP problems.

IV. MOEA/D AND NSGA-II WITH PROBABILISTIC REPRESENTATION

In this section, we first describe a quantum-inspired genetic algorithm (QEA), in which the probabilistic representation was first used. Then, the modified versions of MOEA/D and NSGA-II with probabilistic representation are presented.

A. QEA

QEA is an evolutionary algorithm paradigm based on the concepts of qubits and superposition of states in quantum mechanics. In this algorithm, solutions are encoded using a Q-bit representation system. Each Q-bit corresponds to two states '0' or '1'. The probabilities of finding these two states are denoted by α^2 and β^2 respectively. Normalization of states needs to guarantee $\alpha^2 + \beta^2 = 1$. An individual in a n Q-bit system can be defined as:

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \beta_1 & \beta_2 & & \beta_n \end{array} \right] \tag{5}$$

where n is the number of Q-bits, and $(\alpha_i, \beta_i), i = 1, \ldots, n$ is a pair of complex numbers specifying the probability amplitudes of states '0' and '1' of the i-th Q-bit, respectively. Essentially, each Q-bit individual is a simple probabilistic model, which can represent multiple states of Q-bits. In QEA, a population of probabilistic Q-bit individuals is maintained. It can also be viewed as a multi-modal estimation of distribution algorithm (EDA). Like classical EDA algorithms, QEA produces offspring solutions by sampling from a probabilistic model, i.e., observing Q-bit probabilistic individuals. Since each Q-bit individual stores global statistical information of good solutions obtained previously in the search, evolutionary algorithms with Q-bit representation have advantage in exploring promising areas of the search space. In [12], a multi-objective version of quantum-inspired evolutionary algorithm, called QMEA, was also proposed. That algorithm uses NSGA-II as its baseline algorithm and samples new solutions by observing Q-bit individuals.

B. MOEA/D

In [9], Zhang and Li proposed a decomposition-based evolutionary multi-objective optimization algorithm, called MOEA/D. It decomposes a multi-objective optimization problem into NP single objective subproblems with objective functions $g(x, \lambda^{(i)}), i \in \{1, \ldots, \mathrm{NP}\}$. The main goal of MOEA/D is to find the optimal solutions of NP subproblems simultaneously. The information from the optimization of one subproblem is used to assist the optimization of similar subproblems. MOEA/D provides a general framework, which is very flexible to integrate some single-objective metaheuristic search methods, such as differential evolution (DE) [19], simulated annealing (SA) [20] and greedy randomized adaptive search procedure (GRASP) [21].

```
Step 1: Initialization
    Step 1.1: Initialize NP weight vectors $\lambda^{(s)}, s = 1, \ldots, \mathrm{NP}$
    and set $t := 0$. For each weight vector $\lambda^{(s)}$, calculate
    its neighborhood $B(s)$, which includes the indexes of $K$
    closest weight vectors to $\lambda^{(s)}$;
    Step 1.2: Initialize pheromone trails $\tau_{ij}^{(s)}(t)$ and subproblem heuristics $\eta_{ij}^{(s)}$, $s = 1, \ldots, \mathrm{NP}$;
    Step 1.3: Initialize NP solutions in current population
    $x^{(s)}(t), s = 1, \ldots, \mathrm{NP}$ by sampling from probabilistic
    individuals. Set EP as an empty set.
Step 2: Reproduction and Update
for each subproblem $s \in \{1, \ldots, \mathrm{NP}\}$
    Step 2.1: Sample an offspring $y$ from the probability
    model based on $\tau_{ij}^{(s)}(t)$ and $\eta_{ij}^{(s)}$;
    Step 2.2: Improve $y$ by applying local search;
    Step 2.3: Update EP with $y$;
    Step 2.4: Compete $y$ with all neighbors.
    For each $k \in B(s)$, set $x^{(k)}(t) = y$ if $x^{(k)}(t)$ is worse
    than $y$ regarding $g(x, \lambda^{(k)})$.
end for
Step 3: Pheromone Update
for each subproblem $s \in \{1, \ldots, \mathrm{NP}\}$
    Step 3.1: Evaporate pheromone trails $\tau_{ij}^{(s)}(t)$;
    Step 3.2: Update pheromone trails $\tau_{ij}^{(s)}(t)$ with $x^{(k)}(t)$,
    $k \in B(s)$;
end for
Step 4: Termination
If the stopping conditions are satisfied, then output EP and stop;
otherwise set $t := t + 1$ and go to Step 2.
```

Procedure 1: MOEA/D

In the previous implementation of MOEA/D, each subproblem is only associated with one individual, which is its current best solution. As discussed in [19], the competition between neighboring solutions might cause loss of diversity. To avoid this weakness, more complex memory should be used to store the information about the history of the search for each subproblem. This memory could be a small population or a probability model.

So far, no work has yet been done on MOEA/D with probabilistic representation. In this paper, we investigate a new version of MOEA/D, in which each subproblem is associated with one probabilistic individual represented by pheromone trails. At each generation t, MOEA/D maintains the following data structures:

- A set of NP weight vectors - $\lambda^{(1)}, \ldots, \lambda^{(\mathrm{NP})}$;
- Current population - $\{x^{(1)}(t), \ldots, x^{(\mathrm{NP})}(t)\}$, where $x^{(s)}(t), s = 1, \ldots, \mathrm{NP}$ is the current best solution to $g(x, \lambda^{(s)})$;
- Pheromone population: $\{\tau_{ij}^{(1)}(t), \ldots, \tau_{ij}^{(\mathrm{NP})}(t)\}$, where $\tau_{ij}^{(s)}(t)$ is the pheromone trail for the subproblem associated with $g(x, \lambda^{(s)})$.
- Subproblem heuristics: $\{\eta_{ij}^{(1)}, \ldots, \eta_{ij}^{(\mathrm{NP})}\}$, where $\eta_{ij}^{(s)}$

is the heuristic for the subproblem associated with $g(x, \lambda^{(s)})$.

- External population EP which is used to store all non-dominated solutions found during the search.

The algorithmic framework of MOEA/D with probabilistic representation is described in **Procedure 1**.

Step 1.1 is the same as in the original MOEA/D. In Step 1.2, NP probabilistic individuals are initialized. This step needs problem-specific information. Step 1.3 samples the initial population with probabilistic individuals. Similarly, Step 2.1 generates an offspring solution for the current subproblem. Then, it is improved by local search in Step 2.2. The acceptance function in local search is the objective function of the current subproblem. In Step 2.3, the improved solution is used to update the external population EP. If y is not dominated by any member of EP, then add it into EP. If any member of EP is dominated by y, then it is removed from EP. In Step 2.4, y competes with all its neighbors. After sampling and updating the current population, the probabilistic individuals are updated in Step 3. For each subproblem, the related pheromone trails are evaporated in Step 3.1, and then updated by the current solutions of all neighboring subproblems in Step 3.2.

C. NSGA-II

```
Step 1: Initialization
    Step 1.1: Initialize NP weight vectors randomly and set $t := 0$;
    Step 1.2: Initialize $\{\tau_{ij}^{(1)}(t), \ldots, \tau_{ij}^{(\text{NP})}(t)\}$ (pheromone trails) and $\{\eta_{ij}^{(1)}, \ldots, \eta_{ij}^{(\text{NP})}\}$(priori heuristics);
    Step 1.3: Initialize $\{x^{(1)}(t), \ldots, x^{(\text{NP})}(t)\}$ randomly or by sampling from probabilistic individuals. Set EP as an empty set.
Step 2: Sampling
for $s = 1$ to NP
    Step 2.1: Sample an offspring $y^{(s)}(t)$ from the probability model based on $\tau_{ij}^{(s)}(t)$ and $\eta_{ij}^{(s)}$;
    Step 2.2: Improve $y^{(s)}(t)$ by applying local search;
    Step 2.3: Update EP with $y^{(s)}(t)$.
end for
Step 3: Nondominated Sorting
    Step 3.1: Rank the union of $\{x^{(1)}(t), \ldots, x^{(\text{NP})}(t)\}$ and $\{y^{(1)}(t), \ldots, y^{(\text{NP})}(t)\}$ using non-dominated sorting method;
    Step 3.2: Copy the set Q of the best NP solutions into the population into the next generation;
Step 4: Pheromone Update
for $s = 1$ to NP
    Step 4.1 Evaporate pheromone trails $\tau_{ij}^{(s)}(t)$;
    Step 4.2 Update $\tau_{ij}^{(s)}(t)$ by $y^{(s)}(t)$ and all solutions in $Q$ which dominates $y^{(s)}(t)$
Step 5: Termination
If the stopping conditions are satisfied, then output EP and stop; otherwise set $t := t + 1$ and go to Step 2.
```

Procedure 2: NSGA-II

Deb et al. proposed a fast non-dominated sorting genetic algorithm in [6], called NSGA-II. In this algorithm, the population is divided into a number of non-dominated fronts in terms of Pareto dominance. Within each front, all solutions are mutually non-dominated. The solutions in the front closer to the true Pareto front are preferred in selection and competition. To control the distribution of non-dominated solutions within each front, the density values of individuals are estimated based on crowding distance. The solutions in the less crowed part are preferred in selection.

Compared to MOEA/D, NSGA-II has no bias in searching any particular part of the Pareto front. All non-dominated solutions in the current population have equal chance to be selected for reproduction. However, this might not be efficient when sampling offspring solutions due to the following two reasons. First, the non-dominated solutions might have very different structures in the decision space. Therefore, the possibility of generating high-quality offspring solutions by recombining these solutions is low. Second, the design of recombination operators is often problem-dependent. Efficient recombination operators for some combinatorial optimization problems are not always readily available.

Here, we also investigate the version of NSGA-II with probabilistic representation, in which multiple probability models (pheromone trails) are maintained. Each probability model stores long-term search information. All offspring solutions are sampled from these probability models. Similar to MOEA/D in **Procedure 1**, NSGA-II also needs to maintain a set of weight vectors, current population, pheromone population, subproblem heuristics and external population. The details of NSGA-II with pheromone representation are described in **Procedure 2**.

Steps 1.1 and 1.2 initialize pheromone population, current population, and subproblem heuristics. These are similar to the related steps in MOEA/D. Step 2.1 samples offspring solutions from a probabilistic individual. Unlike the original NSGA-II, tournament selection and crossover are not needed here. In Step 2.2, a local search is applied to improve offspring solutions. The acceptance function used in local search is a scalar function $g(x, \lambda)$ with normalized weight vector generated randomly. The external population is updated in Step 2.3, which use the same rules as in MOEA/D. Step 3 ranks all solutions in the union of current and offspring populations by applying non-dominated sorting. The best NP solutions are copied into the population Q in the next generation. In Step 4.1, the pheromone trails of each probabilistic individual are evaporated. Then, they are updated by the offspring solution generated by this probabilistic individual as well as those members of Q that dominate the offspring solution.

V. Pheromone Trails and Heuristics in Multi-objective travelling Salesman Problem

In this paper, we tested MOEA/D and NSGA-II with probabilistic representation on the multi-objective travelling salesman problem. A travelling salesman problem with m

objectives can be formulated as:

$$f_i^{(k)}(\pi) = \sum_{j=1}^{n-1} d_{\pi_i\pi_j}^{(k)} + d_{\pi_1\pi_n}^{(k)}, k = 1, \ldots, m \tag{6}$$

where

- $\pi = (\pi_1, \ldots, \pi_n)$ is a permutation of $\{1, \ldots, n\}$ where n is the number of cities.
- $d_{\pi_i\pi_j}^{(k)}$ is the cost between city π_i and city π_j regarding criterion k.

Like the single objective ACO algorithms for travelling salesman problem, solution components correspond to cities (nodes) in a graph while links between components correspond to edges between cities. In MOEA/D, the heuristic values of each subproblem $g(x, \lambda^{(s)})$ are defined as

$$\eta_{\pi_i\pi_j}^{(s)} = 1/\sum_{k=1}^{m} \lambda_k^{(s)} d_{\pi_i\pi_j}^{(k)}.$$

The definition of heuristic values in NSGA-II is almost the same as that in MOEA/D. The difference is that NSGA-II uses weight vectors generated randomly and changed at each construction step. In this paper, only bi-objective TSP instances are tested. MOEA/D uses a set of NP weight vectors with uniform distribution, which are generated by $\lambda_1^{(s)} = 1/(\text{NP-1})$ and $\lambda_2^{(s)} = 1 - \lambda_1^{(s)}, s = 1, \ldots, \text{NP}$.

The values of pheromone trails in both algorithms are initialized by:

$$\tau_{\pi_i\pi_j}^{(s)}(0) = m \,/ \sum_{k=1}^{m} \min_{x \in P} f_k(x)$$

where $P = \{x^{(1)}(0), \ldots, x^{(\text{NP})}(0)\}$.

For each probabilistic individual, the values $\tau_{\pi_i\pi_j}^{(s)}$ of pheromone trails are increased by a certain solution x as follows:

$$\Delta\tau_{\pi_i\pi_j}^{(s)}(t)|_x = \begin{cases} 1/g(x, \lambda^{(s)}) & \text{if edge } (\pi_i, \pi_j) \text{ belongs to } x \\ 0 & \text{Otherwise.} \end{cases}$$

The total amount of pheromone values changed is:

$$\Delta\tau_{\pi_i\pi_j}^{(s)}(t) = \sum_{x \in S} \Delta\tau_{\pi_i\pi_j}^{(s)}(t)|_x,$$

where S is the set of solutions selected for pheromone update.

VI. Computational Experiments

In this section, we compare the original MOEA/D and NSGA-II to their variants with probabilistic representation based on pheromone trails, which are denoted by MOEA/D-ACO and NSGA-II-ACO respectively.

A. Experimental Settings

In our experiments, we used two 50-city TSP instances (KROAB50 and KRCD50) as well as two 100-city TSP instances (KROAB100 and KROCD100) [22]. All four instances are bi-objective. They are constructed by combining two benchmark single objective TSP instances. The ACS probability decision rules are used in both algorithms. We also compared them with the original versions of MOEA/D and NSGA-II, in which cycle crossover is used for reproduction. The population size is set to 200 in all algorithms for all instances. Both α and β used in probabilistic decision rules are set to 1. The probability q_0 in ACS decision rule is 0.95 while the pheromone evaporation rate is 0.1. The neighborhood size in MOEA/D is 20. We also investigated the performance of MOEA/D and NSGA-II with local search. The 2-opt local move, which randomly exchanges two edges in the tour, is used to generate neighboring solutions. Each local search procedure is stopped after examining 100 neighbors. The total number of runs of each algorithm for each instance was set to 20. In each run, each algorithm is stopped after 50 seconds for the instances with 50 cities and 100 seconds for the instances with 100 cities. All algorithms were implemented in C++ and run in PC computer (Intel (R) Core (TM)2 CPU, 1.86 GHz, 2GB RAM) running Windows XP.

TABLE I

Average number of function evaluations used by MOEA/D and NSGA-II without local search

Instance	MOEA/D	NSGA-II	MOEA/D-ACO	NSGA-II-ACO
KROAB50	1487523	825694	95985	67450
KROCD50	1489063	862066	95948	68460
KROAB100	2395545	1449649	49068	43800
KROCD100	2379413	1437141	49218	43610

To measure the quality of the non-dominated solutions found by MOEA/D and NSGA-II, we use the inverted generation distance (IGD) metric in our experiments [9], which can be formulated as: $\text{IGD}(P, P^*) = \sum_{u \in P^*} \min_{v \in P} \text{dist}(u, v)/|P^*|$. The IGD metric computes the average distances of solutions in a reference set P^* to the set P of resultant solutions. To measure the range of objectives in P, we also calculate the following indicator values: $\text{R}^{(i)}(P) = |\max_{x \in P} f_i(x) - \min_{x \in P} f_i(x)|, i = 1, \ldots, m$.

B. Experimental Results and Analysis

1) MOEA/D and NSGA-II without Local Search: Fig. 1 and 2 plot the non-dominated solutions found by MOEA/D, NSGA-II, MOEA/D-ACO and NSGA-II-ACO after 20 runs. The average number of solutions examined by each algorithm are reported in Table I. Table II shows the mean and standard deviation IGD-metric values of non-dominated solutions found by the four algorithms. The average differences between the best and the worse values of the non-dominated solutions obtained by the four algorithms on each objective are summarized in Table III. From these results, we can make the following observations.

- Table I shows that the average number of solutions examined by MOEA/D-ACO and NSGA-II-ACO is much less than the number of solutions by MOEA/D and NSGA-II. This can be explained by the fact that the computational complexity of constructive heuristics in MOEA/D-ACO and NSGA-II-ACO is higher than that of crossover and mutation operators in MOEA/D and

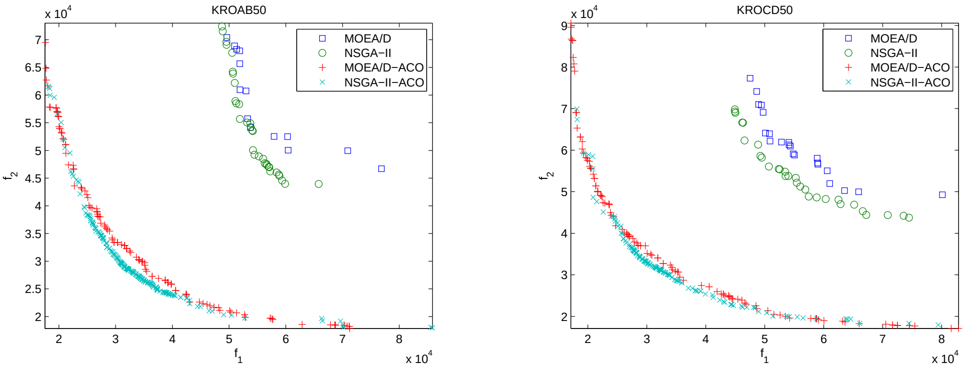

Fig. 1. Plots of the non-dominated solutions found by MOEA/D and NSGA-II on KROAB50 and KROCD50

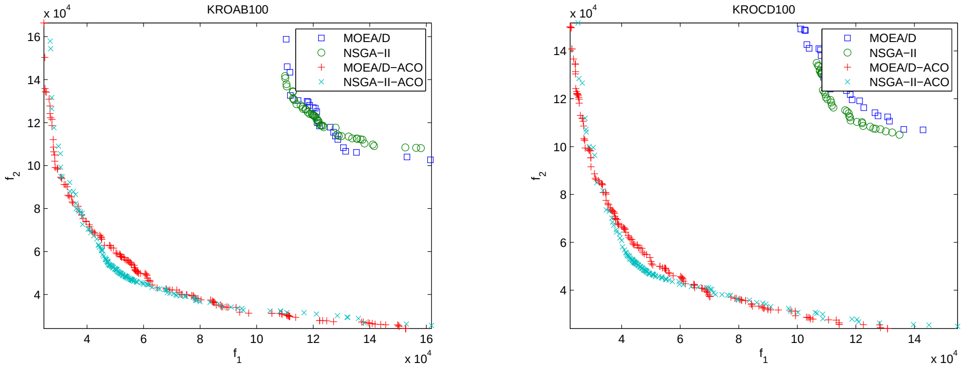

Fig. 2. Plots of the non-dominated solutions found by MOEA/D and NSGA-II on KROAB100 and KROCD100

TABLE II

MEAN AND (STANDARD DEVIATION) OF IGD-METRIC VALUES FOR THE NON-DOMINATED SOLUTIONS FOUND BY MOEA/D(ACO) AND NSGA-II(ACO) WITHOUT LOCAL SEARCH

Instance	MOEA/D	NSGA-II	MOEA/D-ACO	NSGA-II-ACO
KROAB50	41607.1 (4112.9)	37510.0 (2714.1)	3200.6 (190.9)	2104.1 (230.4)
KROCD50	39180.4 (2396.8)	34695.0 (2370.6)	2890.4 (97.1)	2530.7 (519.1)
KROAB100	104531.7 (5587.2)	100496.0 (3144.3)	6106.5 (257.6)	5146.6 (367.4)
KROCD100	105132.8 (4759.5)	101922.1 (4746.6)	5835.5 (351.8)	4392.3 (411.7)

TABLE III

AVERAGE RANGES ($R^{(1)}, R^{(2)}$) OF OBJECTIVES IN THE NON-DOMINATED SOLUTIONS FOUND BY MOEA/D AND NSGA-II WITHOUT LOCAL SEARCH

Instance	MOEA/D	NSGA-II	MOEA/D-ACO	NSGA-II-ACO
KROAB50	(18119, 18819)	(22391, 22052)	(58120, 49713)	(54800, 44947)
KROCD50	(19892, 18093)	(20684, 21157)	(57416, 62977)	(47386, 53867)
KROAB100	(21882, 27155)	(28099, 27216)	(135466, 129729)	(123297, 114071)
KROCD100	(26880, 27700)	(25900, 24171)	(124089, 120009)	(114902, 110945)

NSGA-II. These results also indicate that the problem-specific constructive heuristics are more efficient than the standard crossover and mutation operators when sampling offspring solutions.

- It can be seen from Fig. 1 and 2 that the non-dominated solutions found by MOEA/D-ACO and NSGA-II-ACO are much better than those found by MOEA/D and NSGA-II on all 4 instances. However, both EMO algorithms with probabilistic representation have similar performance in approximating the Pareto front. NSGA-II-ACO found better solutions in the middle parts of the Pareto fronts while MOEA/D found better extreme solutions close to the minimum of each objective and achieved better distribution along the Pareto fronts.
- The mean IGD-metric values in Table II show the MOEA/D-ACO and NSGA-II-ACO found better IGD-values than MOEA/D and NSGA-II. These results show that the non-dominated solutions found by two EMO algorithms with probabilistic representation are closer to the Pareto fronts than those found by the original versions. We can also observe that NSGA-II-ACO performed better than MOEA/D-ACO on all 4 instances in terms of the IGD-metric.
- The results in Table III show that MOEA/D-ACO

TABLE IV
AVERAGE NUMBER OF FUNCTION EVALUATIONS USED BY MOEA/D AND NSGA-II WITH LOCAL SEARCH

Instance	MOEA/D	NSGA-II	MOEA/D-ACO	NSGA-II-ACO
KROAB50	35436936	36149497	8039799	5978060
KROCD50	33732260	34546771	7932084	5880482
KROAB100	58859145	62359764	4631526	4156110
KROCD100	59422778	62902484	4631292	4156555

performed better than NSGA-II-ACO in diversity. The ranges of non-dominated fronts found by MOEA/D-ACO are wider than those of non-dominated solutions obtained by NSGA-II-ACO. The reason is that MOEA/D-ACO specifically spends equal computational effort in searching different parts of the Pareto fronts. In contrast, NSGA-II-ACO does not have any bias to search any part of the Pareto fronts.

2) MOEA/D and NSGA-II with Local Search: Fig. 3 and 4 plot the non-dominated solutions found by the four algorithms using local search in 20 runs. The average numbers of solutions examined by each algorithm are reported in Table IV. The mean and standard deviation IGD-metric values of non-dominated solutions found by the four algorithms are shown in Table V. The average differences between the best and the worse values of the non-dominated solutions obtained by the four algorithms for each objective are summarized in Table VI. From these results, the following observations can be made.

- The results in Fig. 3 and 4 show that all four algorithms with local search are able to find non-dominated solutions close to the Pareto fronts. Again, NSGA-II-ACO performed better than MOEA/D-ACO in approximating the middle parts of the Pareto fronts. From the subfigures, we can see that NSGA-II performed better than MOEA/D in convergence towards the middle parts of the Pareto fronts no matter if probabilistic representation is used or not. However, it is also evident that MOEA/D has better performance in diversity.
- The mean and standard deviation of IGD-metric values in Table V show that MOEA/D outperformed NSGA-II on KROAB50, KROCD50, and KROAB100, and performed slightly worse on KROCD100 in minimizing the closeness to the Pareto fronts. This is due to the advantage of MOEA/D in diversifying the search along the Pareto fronts. We can also note that MOEA/D-ACO is inferior to MOEA/D on all instances in terms of the IGD-metric. This suggests that MOEA/D with local search can have excellent performance in converging towards the Pareto fronts. However, it needs to consume more number of function evaluations (see Table IV).
- Again, the differences between the best and worse values of the non-dominated solutions show the MOEA/D-ACO found wider ranges of non-dominated fronts than MOEA/D. This indicates that the use of probabilistic representation in MOEA/D is beneficial in diversifying the search along the Pareto fronts.

VII. CONCLUSION

In this paper, we investigated the performance of two state-of-the-art evolutionary multi-objective (EMO) algorithms, namely MOEA/D and NSGA-II, when using probabilistic representation based on pheromone trails. In both algorithms, an individual is encoded by a probability model, i.e. pheromone trails. Offspring solutions are sampled from the probabilistic individuals and problem-specific heuristics. We compared the four algorithms without local search - MOEA/D, NSGA-II, MOEA/D-ACO and NSGA-II-ACO on the multi-objective travelling salesman problem. Our experimental results showed that MOEA/D-ACO and NSGA-II-ACO performed much better than the original MOEA/D and NSGA-II without probabilistic representation. MOEA/D-ACO performed worse than NSGA-II in finding solutions in the middle parts of Pareto fronts but it found more diverse Pareto fronts. We also compared the four algorithms when using local search. Our results showed that MOEA/D found better IGD-metric values than MOEA/D-ACO. However, it needs more number of function evaluations. We conclude that probabilistic representation can improve the efficiency of EMO algorithms in sampling high-quality solutions by constructive heuristics and in diversifying the search along Pareto fronts by pheromone trails. In the future, we intend to apply EMO algorithms with probabilistic representation to other challenging multi-objective combinatorial optimization problems.

REFERENCES

[1] F. Rothlauf, *Representations for Genetic and Evolutionary Algorithms*, 2nd ed. Berlin: Springer, 2006.

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

[3] C. Coello Coello, G. Lamont, and D. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. New York: Springer, 2007.

[4] T. Okabe, “Evolutionary multiobjective optimisation with a hybrid representation,” in *CEC2003*, 2003, pp. 2262–2269.

[5] G. Colombo and C. L. Mumford, “Comparing algorithms, representations and operators for the multi-objective knapsack problem,” in *CEC2005*, 2005, pp. 1268–1275.

[6] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm- NSGA-II,” *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[7] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization,” in *EUROGEN 2002*, 2002, pp. 95–100.

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

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

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

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

[12] Y. Kim, J.-H. Kim, and K.-H. Han, “Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 knapsack problem,” in *CEC2006*, 2006, pp. 9151–9156.

[13] M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, 2004.

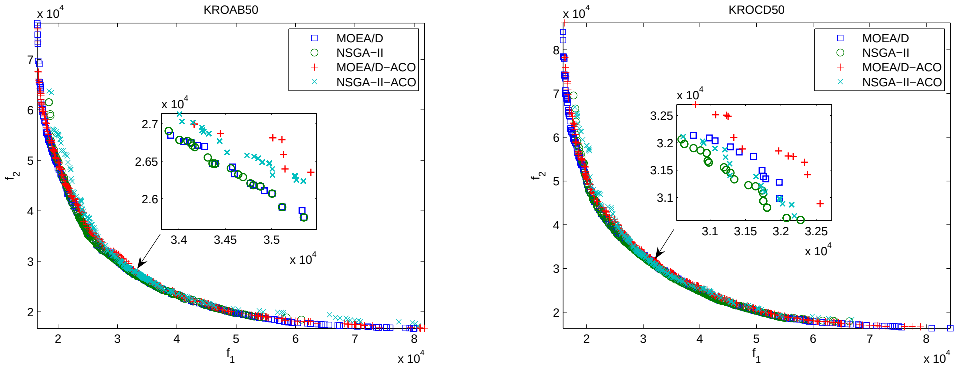

Fig. 3. Plots of the non-dominated solutions found by MOEA/D and NSGA-II with local search on KROAB50 and KROCD50

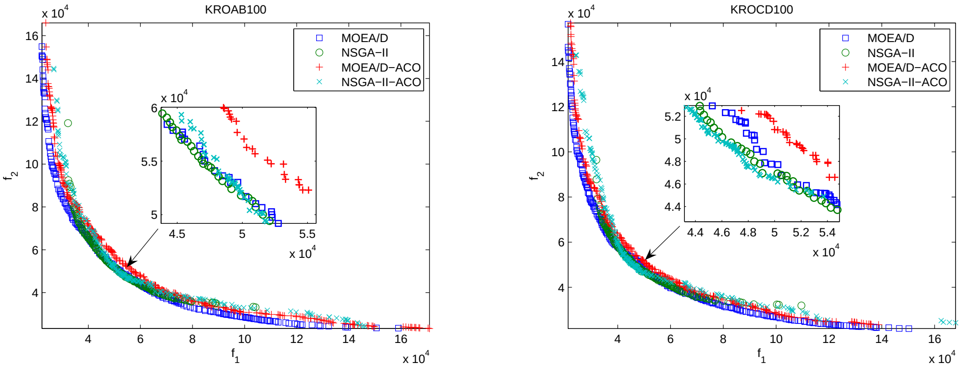

Fig. 4. Plots of the non-dominated solutions found by MOEA/D and NSGA-II with local search on KROAB100 and KROCD100

TABLE V

MEAN AND (STANDARD DEVIATION) OF IGD-METRIC VALUES FOR THE NON-DOMINATED SOLUTIONS FOUND BY MOEA/D AND NSGA-II WITH LOCAL SEARCH

Instance	MOEA/D	NSGA-II	MOEA/D-ACO	NSGA-II-ACO
KROAB50	498.3 (140.6)	1588.4 (188.3)	1031.7 (110.0)	1519.5 (158.3)
KROCD50	661.0 (185.4)	1599.6 (392.0)	993.8 (62.3)	2023.7 (183.4)
KROAB100	2354.9 (406.4)	8638.4 (1471.0)	4333.7 (213.6)	4707.9 (463.8)
KROCD100	2483.5 (437.1)	7607.4 (1172.5)	4092.9 (243.1)	3830.8 (336.2)

TABLE VI

AVERAGE RANGES $(R^{(1)}, R^{(2)})$ OF OBJECTIVES IN THE NON-DOMINATED SOLUTIONS FOUND BY MOEA/D AND NSGA-II WITH LOCAL SEARCH

Instance	MOEA/D	NSGA-II	MOEA/D-ACO	NSGA-II-ACO
KROAB50	(61186, 57877)	(41911, 35549)	(63521, 55024)	(57331, 46695)
KROCD50	(61526, 65937)	(43270, 45173)	(62285, 66320)	(49998, 56085)
KROAB100	(127631, 125954)	(63346, 59000)	(140117, 132510)	(123078, 116986)
KROCD100	(121372, 122164)	(65424, 60443)	(126139, 128444)	(120894, 113204)

[14] K. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

[15] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Trans. on Systems, Man, and Cybernetics - Part B*, vol. 26, no. 1, p. 2941, 1996.

[16] L. M. Gambardella, E. Taillard, and M. Dorigo, "Ant colonies for qap," IDSIA, Lugano, Switzerland, IDSIA 97-4, Tech. Rep., 1997.

[17] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, p. 5366., 1997.

[18] G. García-Martínez, O. Cordón, and F. Herrera, "A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for bi-criteria tsp," *European Journal of Operational Research*, vol. 180, no. 1, pp. 116–148, 2007.

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

[20] H. Li and D. Landa-Silva, "Evolutionary multi-objective simulated annealing with adaptive and competitive search direction," in *CEC2008*, 2008, pp. 3311–3318.

[21] ——, "An Elitist GRASP Metaheuristic for the Multi-objective Quadratic Assignment Problem," in *EMO2009*, 2009, pp. 481–494.

[22] A. Jaszkiewicz, "Genetic local search for multi-objective combinatorial optimization," *European Journal of Operational Research*, vol. 137, no. 1, pp. 50–71, 2002.