WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IEEE CEC

A Hybrid Estimation of Distribution Algorithm for Solving the Multi-objective Multiple Traveling Salesman Problem

V. A. Shim
Department of Electrical and Computer Engineering
National University of Singapore
Singapore
g0800438@nus.edu.sg

K. C. Tan
Department of Electrical and Computer Engineering
National University of Singapore
Singapore
eletankc@nus.edu.sg

K. K. Tan
Department of Electrical and Computer Engineering
National University of Singapore
Singapore
eletankk@nus.edu.sg

Abstract—**The multi-objective multiple traveling salesman problem (MmTSP) is a generalization of the classical multi-objective traveling salesman problem. In this paper, a formulation of the MmTSP, which considers the weighted sum of the total traveling costs of all salesmen and the highest traveling cost of any single salesman, is proposed. An estimation of distribution algorithm (EDA) based on restricted Boltzmann machine is used for solving the formulated problem. The EDA is developed in the decomposition framework of multi-objective optimization. Due to the limitation of EDAs in generating a wide range of solutions, the EDA is hybridized with the evolutionary gradient search. Simulation studies are carried out to examine the optimization performances of the proposed algorithm on MmTSP with different number of objective functions, salesmen and problem sizes.**

Keywords- Decomposition; estimation of distribution algorithm; evolutionary gradient search; hybrid multi-objective optimization; multiple traveling salesman problem, restricted Boltzmann machine

I. INTRODUCTION

The multiple travelling salesman problem (mTSP) is a generalization of the classical travelling salesman problem (TSP). In the mTSP, m salesmen are involved in a routing to visit n cities ($m < n$) in order to achieve a common goal. In the routing, all the salesmen will start from and end at the single depot after visiting the ordered cities. Each city can only be visited once, and the total cost for all salesmen is required to be minimized. The cost can be defined as distance, time, expense, risk, etc. The complexity of the mTSP is higher than the TSP since it is required to allot a set of cities to each salesman in an optimal ordering while minimizing the total travelling cost for all salesmen. Furthermore, the mTSP is more appropriate for real life scheduling or logistic problems than the TSP because more than one salesman is usually involved. The mTSP is closely related to the school bus routing problem [1], design of global navigation satellite system [2], interview scheduling [3], hot rolling scheduling [4] and mission planning [5], among others. Over the past few decades, research on the TSP has attracted a great deal of attention. However, the mTSP has not received the same amount of research effort compared to the TSP.

Many real life scheduling problems also involve several conflicting objectives that have to be optimized simultaneously. In a multi-objective optimization problem (MOP) [6-7], no single point is an optimal solution. Instead, the optimal solution is a set of non-dominated solutions, which represents the tradeoff between the multiple objectives. In this case, fitness assignment to each solution in the evolutionary multi-objective evolutionary optimization is an important feature for the assurance of the survival of fitter and less crowded solutions to the next generation. Much research has been carried out over the past few decades to address this issue, and fitness assignment based on the domination approach is one of the most popular approaches. The algorithms that use the domination approach for solving MOPs are non-dominated sorting genetic algorithm-II (NSGA-II) [8] and strength Pareto evolutionary algorithm2 (SPEA2) [9], among others. However, the fitness assignment in the domination approach is less efficient in solving many-objective problems. This is because the strength of the domination among the solutions in a population is weakened with the increase in the number of objective functions. This phenomenon results in poor decision making in the selection of promising solutions.

Recently, the classical method for multi-objective optimization based on decomposition has been re-formularized into a population-based approach [10-11]. The decomposition approach decomposes an MOP into several subproblems and subsequently optimizes all the subproblems concurrently. Under this approach, it is not required to differentiate the domination behaviours of the solutions. Instead, the subproblems are constructed using any aggregation approach, and the superiority of the solutions is determined using the aggregated values.

Evolutionary computation (EC) is a computing paradigm inspired from biological evolution. In EC, it is commonly viewed that the reproduction process, from a biological perspective, may involve the creation of offspring based on the recombination between two parents (crossover) or random perturbation in a parent (mutation). An alternative to this approach is to estimate the probability distribution of the parent solutions in the decision space so as to predict the distribution of the unknown solutions. This approach is commonly known as estimation of distribution algorithm (EDA) [12-14]. In an EDA, a probabilistic model is constructed from the parent solutions by any modelling paradigm including statistical methods [15], probability mechanisms [16-17] or machine learning approaches [18-20]. Subsequently, children solutions are generated through the sampling of the constructed model.

This paper aims to study the performance of an EDA in solving the multi-objective mTSP (MmTSP). Particularly, an EDA using restricted Boltzmann machine (REDA) as its modelling approach is employed. REDA is developed in the decomposition framework of multi-objective optimization. Due to the limitation of EDAs in generating a set of diverse solutions, REDA is hybridized with a local search based on the evolutionary gradient search (EGS). A new formulation of the objective function for the MmTSP is proposed. The formulation considers the weighted sum of the total traveling costs of all salesmen and the highest traveling cost of any single salesman. The proposed hybrid algorithm (hREDA) is then used to solve the formulated problem. Simulation studies are carried out on various instances of the problems with different number of objective functions, salesmen and cities. hREDA are then rigorously compared with several state-of-the-art evolutionary multi-objective optimizers.

The remaining parts of the paper are organized as follows. Section II presents a literature review on the application of EC in mTSP, as well as a brief introduction of REDA and the EGS that will be studied in this paper. Section III describes the problem formulation of the MmTSP. The proposed algorithm is highlighted in Section IV. Section V presents the experimental results. Section VI draws the conclusion of this paper.

II. Literature Review

This section presents a literature review that focusing on the application of evolutionary approaches to mTSP. A more rigorous review of the works that involved non-evolutionary techniques can be found in [21]. A brief introduction on REDA and the EGS will also be presented.

A. *Multiple traveling salesman problems (mTSP)*

In [22], the first implementation of genetic algorithm (GA) to solve the mTSP was carried out. The authors used a simple GA with natural representation to schedule the multiple teams of photographers to visit a large number of elementary and secondary schools. The objective is to minimize both the total distance traveled and the time consumed, such that the time constraints are satisfied and each team must be able to visit at least two schools daily. Unfortunately, the authors did not elaborate on how they manipulated multiple teams in their chromosome representation.

In [4], Tang *et al.* studied a hot rolling scheduling problem in Shanghai Baoshan Iron and Steel Complex. The problem considers actual production constraints and aims to schedule multiple turns within the same shift. The authors first modeled the hot rolling scheduling problem as an mTSP. Next, the mTSP was converted into a classical TSP through the proposal of a one-chromosome representation. The selection operator was modified such that the best solutions obtained so far were always selected to be one of the parent chromosomes to undergo crossover operation.

In [23], the author studied the vehicle scheduling problem using GA. Since there are multiple vehicles involve in the routing, the problem can basically be classified as an mTSP. In the paper, a two-chromosome representation was proposed. The first chromosome locates the cities while the second chromosome indicates which vehicle is to be assigned to visit the city specified in the first chromosome. In [24], a two-part chromosome representation was proposed for the mTSP. Under this scheme, the chromosome for a gene is divided into two distinct parts. The first part of the chromosome allots the permutation of the cities, while the second part of the chromosome determines the number of cities to be visited by each salesman. Thus, there will be an additional of m genes in the chromosome for m salesmen. In this two-part chromosome representation, the solution space is smaller than the two-chromosome representation. The results demonstrated that the proposed representation is able to produce better results than the one-chromosome representation under most of the test instances.

Zhao *et al.* [25] proposed a grouping GA, which utilized the one-chromosome representation, to solve the mTSP. The algorithm employed a pheromone-based crossover operator which utilized information of edge lengths, adjacency relations and pheromone levels to construct new solutions. Several local search strategies (relocation, exchange and 2-opt) were also used to facilitate the search. The grouping GA was also used by Singh and Baghel [26] who proposed a replacement policy to reduce problem redundancy. In their work, two different objective functions were considered – minimizing total distance traveled by all salesmen, and minimizing the maximum distance traveled by any salesman.

Recently, multi-chromosome representation of mTSP solutions was proposed in [27] where the route assigned to each salesman was represented in a chromosome. Therefore, each solution has m associated chromosomes. The crossover and mutation operators designed to deal with the representation were also proposed.

B. *Restricted Boltzmann machine based estimation of distribution algorithm (REDA)*

EDAs are a class of evolutionary computing techniques. EDAs imitate all properties of GA including survival of the fittest, elitism and generation of new solutions from parent solutions. The only difference is that the reproduction process

```
Begin
  1. Initialization: Generate $S$ solutions with a marginal probability of 0.5. Store the solutions in $Pop$.
  Do while ("Stopping criterion is not met")
    2. Evaluation: Get the fitness $F(x)$ of all solutions in $Pop$.
    3. Selection: Select $S1 \leq S$ solutions from $Pop$ using any selection strategy.
    4. Probabilistic modeling:
       4.1 Network training: Train the RBM using the contrastive divergence (CD) [29] training method in order to obtain a set of trained weights ($w$), biases ($b, c$) and hidden states ($h$) of the network.
       4.2 Modeling: Estimate the probability distribution of the selected solutions. In the binary representation, the marginal probability distribution of $x_i$ is constructed as follows.

           $p_g(x_i = 1) = \frac{\sum_{j=1}^{S1}\delta^+(x_i^j)}{Z_i = \sum_{j=1}^{S1}\delta^+(x_i^j) + \sum_{j=1}^{S1}\delta^-(x_i^j)}$

           where

           $\delta(x_i^j) = \begin{cases}\delta^+(x_i^j)\ if\ x_i^j = 1\\ \delta^-(x_i^j)\ otherwise\end{cases}$

           $\delta^+(x_i^j) = \sum_{k=1}^{H} e^{-E(x_i^j=1,h_k^j)}$

           $\delta^-(x_i^j) = \sum_{k=1}^{H} e^{-E(x_i^j=0,h_k^j)}$

           $E(x,h) = -\sum_{i=1}^{n}\sum_{k=1}^{H} x_i h_k w_{ik} - \sum_{i=1}^{n} x_i b_i - \sum_{k=1}^{H} h_k c_k$

           $x_i$ is the $i^{th}$ decision variable, $n$ is the number of decision variables, $Z_i$ is the normalizing constant, $H$ is the number of hidden units, $E$ is the energy function of a RBM, $w_{ik}$ is the weight connecting visible unit $i$ and hidden unit $j$, $b_i$ is the bias of visible unit $i$, $c_k$ is the bias of hidden unit $k$, and $h_k$ is the state of hidden unit $k$.
    5. Sampling: Sample $S$ offspring from $p_g(x)$ using simple sampling technique as follows.

           $x_i = \begin{cases}1\ if\ random\ (0,1) \leq p_g(x_i)\\ 0\ otherwise\end{cases}$

  End Do
End
```

Fig. 1. Pseudo code of REDA

in GA is based on biological recombination, while that in EDA is carried out by constructing a probabilistic model from the parent solutions, and the children solutions are produced by sampling the corresponding model [12-13]. One of the recently developed EDAs is REDA that its modeling approach is based on a restricted Boltzmann machine (RBM) [28]. RBM is an energy-based binary stochastic neural network. The network consists of two layers of neurons, which are a visible layer and a hidden layer. The visible layer is the input layer of the network while the hidden layer determines the capability of the network in modeling the probability distribution of the input stimuli. In REDA, the input stimuli are the alleles of the chromosomes. The pseudo code of a binary REDA is presented in Fig. 1.

C. *Evolutionary Gradient Search (EGS)*

The EGS is a local search that utilizes the gradient information of the trajectory of the solutions to predict the favorable movements in the search space [30-32]. The pseudo code of the EGS is presented in Fig. 2. Firstly, initial step size σ_0 is predetermined. The step size σ is used to govern the degree of mutation applied to generate local neighbors and children solutions. Then, select an individual from the population as the initial solution. L local neighbors are generated by perturbing the initial solution using mutation with normal distribution of zero mean and σ^2 variance. The global gradient direction is subsequently estimated from the local neighbors (Step 5). Next, a gradient child solution is generated (Step 6) and a factor ε is used to control the mutation step size as shown in Step 7. The initial solution is updated in Step 8 if the gradient child solution is fitter than the initial solution. The process is repeated until the stopping criterion is met.

```
Begin
  1. Input: Define initial step size $\sigma_0$. $t = 1$.
  Do while ("Stopping criterion is not met")
    For $j = 1 : K$(Number of solutions undergoing local search)
      2. Initial solution: Select a solution $x^j$ from selection pool.
      3. Reproduction: Create $L$ local neighbors $y^i$, $i \in (1,2,\ldots,L)$ by perturbing $x^j$ using normal mutation $N(0,\sigma_t^2)$.
      4. Evaluation: Calculate the fitness value of $y^i$, $F(y^i)$.
      5. Direction: Estimate the global gradient direction as follows.

         $\hat{v} = \frac{\sum_{i=1}^{L}[F(y^i) - F(x^j)](y^i - x^j)}{\|\sum_{i=1}^{L}[F(y^i) - F(x^j)](y^i - x^j)\|}$

      6. Create offspring:

         $g = x^j - \sigma_t\hat{v}$

      7. Update mutation step size $\sigma_{t+1}$:

         $\sigma_{t+1} = \begin{cases}\sigma_t\varepsilon\ if\ F(g) < F(x^j)\\ \sigma_t/\varepsilon\ \ \ otherwise\end{cases}$ , $\varepsilon = 1.8$

      8. Update solution: if $F(g) < F(x^j)$
                          then $x^j = g$
      9. Output: Output $x^j$ with the best fitness.
    End for $j$
    $t = t + 1$.
  End do
End
```

Fig. 2. Pseudo code of the evolutionary gradient search (EGS)

III. PROBLEM FORMULATION

The mTSP is the generalization of the classical TSP (single salesman), where m salesmen are involved in the routing. The aim is to minimize the total traveling cost of all the salesmen under the condition that each city must be visited strictly once by any salesman, and all the salesmen must return to the starting depot after visiting their final ordered city. The traveling cost, which is the objective function of the problem, could be defined as the traveling distance, traveling time, traveling expense, traveling risk, etc incurred. Each salesman will have his own route and there should be no repeated visit on any city in the route of the salesman.

In the literature, the aim of the mTSP is specified to be either minimizing the total traveling cost of all salesmen or the highest traveling cost incurred by any single salesman [33]. In this paper, the focus is tailored specifically for the mTSP with single depot; considering the minimization of the total traveling cost and the balancing of the workload among all salesmen. This is achieved by formularizing the objective function to be the weighted sum of the total traveling cost of all salesmen and the highest traveling cost of any single salesman. In the context of multi-objective optimization, more than one objective is subject to be minimized, which can be formulated as follows.

Minimize:

$$F(x) = (F_1(x), \dots, F_P(x))$$
$$F_1(x) = w_1 * TC^1(x) + w_2 * MC^1(x)$$
$$\vdots$$
$$F_P(x) = w_1 * TC^P(x) + w_2 * MC^P(x)$$

where

$$TC^k(x) = \sum_{j=1}^{m} IC_j^k(x)$$
$$MC^k(x) = \max_{1 \le j \le m} (IC_j^k(x))$$
$$IC_j^k(x) = \sum_{i=1}^{n_{j-1}} D^k(a_{i,j}, a_{i+1,j}) + D^k(a_{n_{j,j}}, a_{1,j})$$

In the above formulation, $x \in \phi$, ϕ is the decision space, $a_{i,j}$ is the i^{th} visiting city by salesman j, P is the number of objective functions, w_1 and w_2 are the weights to balance between total cost and highest cost (w_1+w_2=1.0), TC is the total traveling cost of all salesmen, MC is the highest traveling cost of any single salesman, IC is the individual traveling cost, m is the number of salesmen, n_j is the number of cities traveled by salesman j, $D^k(a_{i,j}, a_{i+1,j})$ is the traveling cost (for the k^{th} objective function) between cities at locations i and $i+1$ for salesman j. In a chromosome, two conditions should be met, which are all the cities must be visited exactly once and each salesman must be assigned at least one city in his traveling route.

IV. Proposed Algorithms

The proposed algorithm, named as hybrid REDA (hREDA), consists of four main mechanisms. They are chromosome representation, decomposition, modeling and local search. The hREDA is developed in the decomposition multi-objective framework suggested in [11]. In the implementation stage, one-chromosome representation [4] is utilized to represent the order of the cities to be traveled by m salesmen. This scheme introduces $m-1$ pseudo cities (integer values < 0) to the chromosome. These pseudo cities represent the same initial city where all the salesmen will start their routes. Therefore, each chromosome may consist of $n+m-1$ genes. An example of the chromosome representation with nine cities and three salesmen is illustrated in Fig. 3. The sequence of travel is as follows. The first salesman starts from the initial city 0 then visits cities 2, 5 and 7 in that order. The second salesman again starts from the initial city then visits cities 1 and 8 in that order. The last salesman visits cities 6, 4 and 3 in that order.

In the decomposition framework, the fitness assigned to each solution can be based on any classical aggregation approach. In this paper, the Tchebycheff approach [11], [34] is used and hREDA is described according to this approach. A set of evenly distributed weight vectors $\lambda^1, \dots, \lambda^S$ and the reference point z^* are generated, where S is the number of subproblems. The algorithm decomposes the population into S scalar optimization subproblems according to the Tchebycheff formulation and the fitness value of the j^{th} subproblem is defined as:

$$gt(x|\lambda^j, z^*) = max_{1 \le i \le P}\{\lambda_i^j |F_i(x) - z_i^*|\} \quad (1)$$

The pseudo code of hREDA is presented in Fig. 4. In step 1, the Q neighbors (denoted as $B(i) = \{i_1, \dots, i_Q\}$.) that are nearest, in terms of Euclidean distance, to each weight vector are determined. Then, initial chromosomes in the form as indicated in Fig. 3 are randomly generated. Objective values $F_1(x), \dots, F_P(x)$ are calculated based on the formulation in section III, and the reference point z^* is set to the minimum objective value of the initial population. In Step 2, RBM is trained using the CD training mechanism [29] in order to obtain the trained weights, biases and hidden states of the network. Then, the probability model is built by calculating the probability of existence of each city in each permutation location in the chromosome. In the model, a $N \times N$ probabilistic matrix $Pr_g(x_i)$, where $N = n + m - 1$, is constructed (Step 2a). The children solutions are subsequently generated by sampling the constructed probabilistic model (Step 2b).

Since the sampling is carried out marginally, the existing cities in a chromosome are not taken into consideration by the sampling mechanism. Therefore, some cities may appear more than once while some cities may not even be included in a chromosome. The chromosome is repaired according to the heuristic approach proposed in [35] to ensure that there is no repetition of any city. In this approach, those repeated and unallocated cities are determined. The first unallocated city is inserted to the location of the repeated city if that city has the smallest cost (e.g. distance, times, charge, risk, etc) to the adjacent cities in that location. Let takes Fig. 3 as an example and let city 5 as the repeated city and cities 10 and 11 as the unallocated cities. The traveling costs of visiting cities in an order of 2, 10, 7 and 2, 11, 7 are calculated. If it is in the condition that the traveling cost of visiting cities in the order of 2, 11, 7 is smaller than the cost of visiting cities in the order of 2, 10, 7, then city 11 is inserted to the location of the city 5.

In the event that some salesmen are not being assigned any city in their routes, there will be a penalty added to the objective values of the solutions by multiplying the original values with a constant value $\mathcal{L}$. This is done to weaken the solutions since the workloads assigned to some of the salesmen in those solutions are unevenly distributed. In the implementation stage, $\mathcal{L} = 10$ is applied. Step 3 updates z^* followed by FV. For z^*, it is the reference point used in the Tchebycheff approach and is updated by taking the minimum value of the objective functions. It is emphasized that Step 3b is one of the important features in the decomposition framework whereby the fitness of the solutions are assigned according to (1). All solutions sampled from the probabilistic model will also be updated one by one to all neighboring

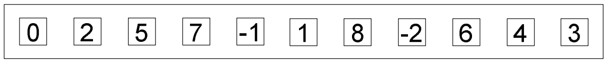


Fig. 3. One-chromosome representation

```
Step 1: Initialization:
    a) Compute the Euclidean distance between all the weight vectors and then group the $Q$ closest weight vectors $B(i) = \{i_1, .., i_Q\}$, $i \in [1, S]$, to each weight vector. $S$ is the population size or number of subproblems.
    b) Randomly generate the initial population $x^1, \ldots, x^S$ in integer number from $[1-m, n-1]$. No integer number is repeated. Set $FV^i = F(x^i)$.
    c) Initialize $z^* = (z_1^*, \ldots, z_P^*)$ by setting $z^*$ according to the minimum objective value of the initial population.
Step 2: Reproduction based on REDA:
    a) Decode the integer representation of the cities into the binary representation. Train the network. Compute the $\delta(x_i^j)$ as shown in Step 4 of Fig. 1. Encode the binary representation of $\delta(x_i^j)$ into integer representation. Construct the probabilistic model $Pr_g(x)$ by computing the marginal probability of each city $(c_1, .., c_N)$, where $N = n + m - 1$, in each permutation location as follows.

        $$Pr_g(x) = \begin{bmatrix} Pr_g(x_1 = c_1) & \ldots & Pr_g(x_N = c_1) \\ \vdots & \ddots & \vdots \\ Pr_g(x_1 = c_N) & \ldots & Pr_g(x_N = c_N) \end{bmatrix}$$

        $$Pr_g(x_i = c_j) = \frac{\sum_{k=1}^{S} \delta_k(x_i = c_j) + Z_i/(N \times S)}{Z_i + Z_i^2/(N \times S)}$$

       where $Pr_g(x_i)$ is the probability distribution of the cities at generation $g$, $Pr_g(x_i = c_j)$ is the probability of city $j$ to be located at the $i^{th}$ position of the chromosome, $c_j$ is the city $j$ $(c_1 = 1 - m, \ldots, c_N = n - 1)$ and $Z_i$ is the normalizing constant as shown in Step 4 of Fig. 1.
    b) Sample $Pr_g(x)$ to generate $S$ children solutions as follows.

        $$y_i = \begin{cases} C_1 \; if \; random\,(0,1) \leq Pr_g(x_i = c_1) \\ C_2 \; if \; Pr_g(x_i = c_1) < random\,(0,1) \leq \sum_{j=1}^{2} Pr_g(x_i = c_j) \\ \vdots \\ C_N \; if \; \sum_{j=1}^{N-1} Pr_g(x_i = c_j) < random\,(0,1) \leq \sum_{j=1}^{N} Pr_g(x_i = c_j) \end{cases} \quad (6)$$

       where $y_i$ is a newly generated city at $i^{th}$ position of a chromosome.
    c) Improvement: Apply specific heuristic approach to repair the chromosomes to ensure that the conditions of routing are satisfied. Penalize the solution if any salesman is not assigned any city by multiplying the objective value with a constant $\mathcal{L}$, $F(y) = \mathcal{L} \times F(y)$.
Step 3: Update solution:
    For $i = 1, \ldots, S$, do
        a) Update of $z^*$: For $j = 1, \ldots, P$, if $z_j^* > F_j(y^i)$, then set $z_j^* = F_j(y^i)$
        b) Update of neighboring solutions: For $j \in B(i)$, if $gt(y^i|\lambda^j, z^*) \leq gt(x^j|\lambda^j, z^*)$, then set $x^j = y^i$ and $FV^j = F(y^i)$.
    End do
Step 4: Local search:
    a) Perform local search if local search is activated. Next, apply Step 3 to update the created children solutions.
Step 5: Stopping criterion: If stopping criterion is met, then stop. Else, go to Step 2.
```

Fig. 4. Pseudo code of hREDA

TABLE I
PARAMETER SETTINGS FOR EXPERIMENTS

Population size or number of subproblem, $\boldsymbol{S}$	Number of cities, $\boldsymbol{n}$ [44]
Number of cities, $\boldsymbol{n}$	100, 300, 500
Number of salesmen, **m**	2,5,10 for 100 cities; 2,5,10,30 for 300 cities; 2,5,10,20,50 for 500 cities
Number of objective functions, $\boldsymbol{P}$	2 and 5
$\boldsymbol{Q}$	Population size, S
Number of local neighbors, $\boldsymbol{L}$	10
Computing budget in fitness evaluations	200 000 for 100 cities; 600 000 for 300 cities; 1 000 000 for 500 cities
Crossover and mutation rates in NSGA-II and MOEAD	0.8 and 0.005 [44]
Initial step size ($\boldsymbol{\sigma_0}$) and $\boldsymbol{\varepsilon}$ in EGS	300 and 1.8 [41]
Local search activation (in terms of number of fitness evaluations)	After 100,000 for 100 cities; 300,000 for 300 cities; 500,000 for 500 cities

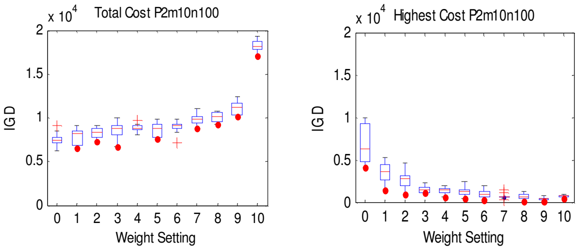
TABLE II
INDICES OF DIFFERENT WEIGHT SETTINGS

Index	0	1	2	3	4	5	6	7	8	9	10
W1	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0
W2	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0

solutions, and the superior solutions will replace the inferior ones.

Local search is performed if it is activated and it is applied every generation thereafter. The same procedure is carried out until the stopping criterion is met. The pseudo code of the EGS is presented in Fig 2. However, some modifications are required to adapt the EGS for the permutation-based problem. Firstly, each subproblem that undergoes local search will be perturbed to generate L local neighbors, which are created by simply swapping two genes in a chromosome. For each local solution, Step 3 is carried out to update the z^* and FV. The fitness of the solutions are aggregated according to the weighted sum approach by using available $\lambda^1, \ldots, \lambda^S$ values. This is different from the previous implementation [30-32] where the weights are randomly generated. The global gradient direction is estimated according to Step 5 of Fig. 2. In Step 6 of Fig. 2, we have x^j as a floating value. However, in MmTSP, x^j is the city which may not be suitable for creating an offspring. As such, the average cost between the local neighbors to the original chromosome is calculated and then assigned to x^j. Following which, g is updated. However, g is a floating value and it cannot be used to represent a city. Thus, we have to determine the candidate city to be the one that has the closest g cost value to the original city. For example, let $g = 100$, city 1 as the original city, and cities 2 and 3 as other cities that will be visited. If the traveling cost between cities 1 and 2 is 200 and the traveling cost between cities 1 and 3 is 150, then city 3 is the candidate city because the traveling cost is closer to g value. The mutation step size is updated according to Step 7 of Fig. 2, and the gradient solution is updated to the population according to Step 3 of Fig. 4. A shared mutation step size σ_t is used to generate the gradient offspring and is adaptively tuned based on the quality of the estimated gradient. $\varepsilon = 1.8$ is used as suggested in the previous work [30-32].

The diversity of the solutions is maintained through the preselected weight vectors. The idea is similar to that in classical aggregation algorithms whereby multiple weight settings are used to produce an estimated optimal solution for each weight setting. However, the decomposition MOEA maintains a set of solutions in each simulation run rather than carrying out multiple runs as seen in classical aggregation approaches. Even though there is no obvious elitism being applied, it is implicitly presented in Step 3b of Fig. 4, where Q nearest neighbors (parents) are updated by comparing their fitness values with those of the offspring.



(a) (b)

Fig. 5. IGD metric for a) total traveling cost of all salesmen and b) highest traveling cost of any single salesman under various weight settings for the MmTSP with two objective functions, 10 salesmen and 100 cities (P2m10n100)

V. Experimental Results

All the algorithms in this study were implemented in C++. The experimental settings are shown in Table I. MmTSP with two and five objectives are studied. A $n \times n$ cost matrix is randomly generated for each objective in the range of [0,1000] [35-37]. For experimental studies, the results are compared based on the performance metrics of inverted generational distance (IGD) [20] and Pareto front. IGD measures the average Euclidean distance between each solution on the Pareto optimal front to the nearest solution on an evolved front. A smaller IGD value implies better proximity and spread. Since the optimal solution set to the problem is unknown, the estimated optimal front is formed using the non-dominated solutions found from all the algorithms and all simulation runs.

Five algorithms are put to comparison. hREDA is the proposed hybrid algorithm. REDA is a pure decomposition EDA using RBM. MOEA/D is a decomposition algorithm with GA proposed in [11] and used in [36]. UMGA is a synthesizing algorithm between an EDA and NSGA-II proposed in [35]. Lastly, NSGA-II is one of the most popular MOEAs based on the concept of domination proposed in [8] and used in [35]. The results presented have been averaged over 10 independent runs with different random number seeds. The test instances consist of 24 MmTSP with different number of objective functions (P), salesmen (m) and problem size (n). The problem is denoted in the form of P2m5n100, which refers to an MmTSP with two objective functions, five salesmen, and 100 cities.

A. *Effects of weight setting on optimization performance*

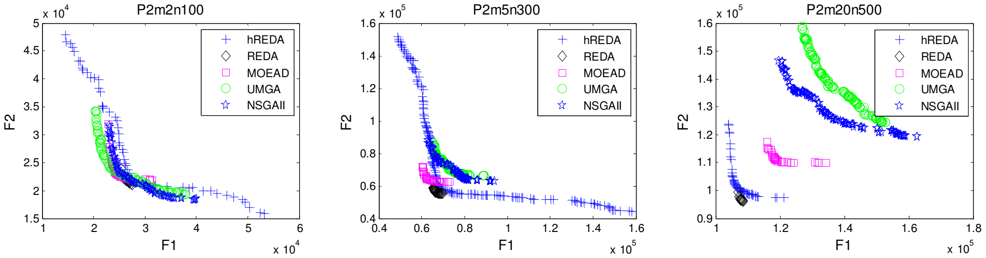
The formulation of the MmTSP in this paper takes into account the weighted sum of total traveling cost of all salesmen and the highest traveling cost of any single salesman. The weight setting is dependent on the preference of the manager whether he wants to achieve the lowest total traveling cost of all salesmen or he wants to achieve the balancing of workload of all salesmen. If the aim is to obtain the lowest total traveling cost, the weights will be set to $w_1 = 1.0, w_2 = 0.0$. On the other hand, if the final objective is to balance the workload of all salesmen, the weights will then be set to $w_1 = 0.0$, $w_2 = 1.0$. However, if the aim is to achieve tradeoff between the two aims, then different weight settings should be employed.

Simulations were carried out to study the performance of hREDA under various weight settings. Fig. 5 shows the IGD metric for a) total traveling cost of all salesmen and b) highest traveling cost of any single salesman under various weight settings for the MmTSP with two objective functions, 10 salesmen and 100 cities (P2m10n100). The indices of the weight settings are illustrated in Table II. From Fig. 5a, it is observed that the algorithm is able to produce solutions with minimum total traveling cost of all salesmen under the weight setting of $w_1 = 1.0, w_2 = 0.0$. This is expected since the weight setting deems solutions with smaller total traveling cost as superior. However, Fig. 5b shows that the weight setting causes imbalance workload for the salesmen. On the other hand, if the focus is to minimize the highest traveling cost of any single salesmen (index 10), it is then observed that the highest cost is far smaller than that of index 0, but this leads to the solutions having the highest total traveling cost. This observation suggests that there is a tradeoff between both aims. Striking a balance between both aims can be achieved by setting the weights to intermediate values, which will lead to the ability to produce routes with smaller total cost of all salesmen and highest cost of any single salesman. For the rest of the experimental studies, the weight setting $w_1 = 0.5, w_2 = 0.5$ is used.

B. *Results for two objective functions*

Simulations were carried out to study the performance of the five algorithms applied on the MmTSP with different number of objective functions, salesmen, and cities. Fig. 6a shows the evolved Pareto front of total traveling cost generated by the algorithms applied to the MmTSP with two objective functions, five salesmen, and 100 cities. It is observed that hREDA is able to produce a set of diverse solutions but it is slightly inferior in terms of proximity to the other algorithms. Similar observations can also be made for the MmTSP with two objective functions, five salesmen, and 300 cities (Fig. 6b). The Pareto front for the MmTSP with 20 salesmen and 500 cities are shown in Fig. 6c. From the figure, it is observed that the decomposition algorithms (hREDA, REDA and MOEA/D) achieve better Pareto front than the domination algorithms (UMGA and NSGA-II). For all the decomposition algorithms, hREDA generates a better set of diverse solutions than REDA. However, the solutions generated by REDA have a better proximity than hREDA. From this observation, it can be concluded that the decomposition algorithms scale well with the increase in the number of decision variables compared to the algorithms using the concept of domination. REDA uses global distribution of the parent solutions to guide the search process and is shown to have good proximity results, but poor solution diversity. Introducing local information into the evolutionary process, which helps the algorithm to further explore and exploit the search space, rectifies this limitation of REDA.

For ease of visualization, the optimization results (mean ± standard deviation) for the different problem settings are



(a) (b) (c)

Fig. 6. Evolved Pareto front of total traveling cost generated by the various algorithms applied to the MmTSP with two objective functions (a) two salesmen and 100 cities (b) five salesmen and 300 cities and (c) 20 salesmen and 500 cities.

TABLE III
IGD METRIC FOR TOTAL TRAVELING COST FOR ALL SALESMEN OF SOLUTIONS OBTAINED BY VARIOUS ALGORITHMS FOR THE MMTSP WITH TWO OBJECTIVES, M SALESMEN AND N CITIES

	hREDA	REDA	MOEA/D	UMGA	NSGA-II
P2m2	**3607**	10795	7349	5689	5881
n100	±793	±469	±952	±738	±651
P2m5	**4964**	11253	8180	7376	6581
n100	±1041	±785	±675	±776	±656
P2m10	8741	12332	8818	9428	**8288**
n100	±627	±501	±746	±840	±671
P2m2	**5779**	28319	26565	24327	25337
n300	±757	±648	±1093	±1474	±1455
P2m5	**8010**	28898	27709	26189	27109
n300	±1805	±805	±1125	±1480	±2244
P2m10	**17509**	29828	27667	32132	27582
n300	±4213	±1703	±761	±1523	±1757
P2m30	**31149**	35198	32554	42952	38103
n300	±1823	±1035	±1534	±1292	±2314
P2m2	**8358**	57196	59346	54436	56002
n500	±2812	±685	±1135	±2261	±1929
P2m5	**12114**	58022	58811	54424	57047
n500	±2319	±869	±1942	±1424	±2008
P2m10	**33680**	58910	59622	61855	57599
n500	±13163	±746	±1760	±24274	±1609
P2m20	**52195**	59336	61076	66323	62250
n500	±1928	±648	±1575	±2159	±1021
P2m50	**58870**	65826	69039	82341	77062
n500	±2753	±1456	±1610	±2635	±1298

TABLE IV
IGD METRIC FOR TOTAL TRAVELING COST FOR ALL SALESMEN OF SOLUTIONS OBTAINED BY VARIOUS ALGORITHMS FOR THE MMTSP WITH FIVE OBJECTIVES, M SALESMEN AND N CITIES

	hREDA	REDA	MOEA/D	UMGA	NSGA-II
P2m2	**8306**	14045	8801	14170	13939
n100	±689	±885	±238	±549	±821
P2m5	**10007**	13824	10623	17370	17364
n100	±1180	±630	±677	±800	±861
P2m10	**12987**	15075	14098	23298	23027
n100	±950	±424	±900	±1347	±1118
P2m2	**16279**	44048	45971	55795	54134
n300	±2203	±1124	±1574	±2397	±1532
P2m5	**21666**	44013	46255	56964	56959
n300	±3350	±7616	±2229	±3343	±3782
P2m10	**29192**	43987	51566	61475	59917
n300	±1492	±1361	±1985	±4005	±2703
P2m30	**46069**	52505	73765	95708	93817
n300	±1989	±1111	±4077	±2438	±3599
P2m2	**24134**	62362	95107	95930	93413
n500	±2105	±785	±448	±2845	±3099
P2m5	**23939**	62041	96631	95532	94692
n500	±5273	±1812	±4331	±3843	±3449
P2m10	**37310**	62770	102610	102940	97700
n500	±1217	±1024	±2816	±3760	±4709
P2m20	**42030**	63930	108060	107810	100330
n500	±2803	±1404	±1912	±3999	±6145
P2m50	**69190**	79290	147620	145400	143650
n500	±3972	±2941	±4644	±4995	±5158

presented in table form. In each table, the best result (mean) for each problem setting is highlighted in bold. Table III presents the total traveling cost for all salesmen of solutions generated by the algorithms for the MmTSP with two objective functions, m salesmen and n cities. The results show that hREDA generate the best solutions in most of the settings. This is caused by the usage of gradient information for local exploitation, which enhances the ability of the algorithm to search for more diverse solutions. From the table, it is also observed that the total traveling cost increases with the increase in the number of salesmen. This is because when more salesmen are involved, the task gets more difficult since the algorithms need to determine the route for each salesman while maintaining the minimum total traveling cost at the same time. Since all salesmen need to return to the home city and the final assigned city could be far from the depot, additional traveling cost may be incurred. For hREDA, the gradient information weakens with the increase in the number of salesmen, resulting in the algorithm not being able to exploit the information as effectively. However, its performance is still the best compared to the other four algorithms.

C. *Results for five objective functions*

Table IV shows the IGD metric for the total traveling cost of all salesmen of solutions obtained by the algorithms for the MmTSP with five objective functions, different number of salesmen and cities. Generally, the performances of algorithms using the decomposition framework (hREDA, REDA and MOEA/D) are superior to those of the algorithms based on the concept of domination (UMGA and NSGA-II) in most of the problem settings. The superiority of the decomposition algorithms is attributed to the aggregation principle used for fitness assignment. The tournament could be carried out by simply comparing the aggregated fitness values of solutions. Solutions with higher fitness values will always be selected to survive and reproduce. On the other hand, the concept of domination (NSGA-II and UMGA) requires that fitness be assigned to each solution based on their rank of domination. In many-objective problems, most of the solutions are non-dominated and are given lower ranks. This may prevent the tournament process from selecting promising solutions to survive.

VI. CONCLUSIONS

This study proposed a hybrid EDA based on RBM for solving the MmTSP. The objective function of the MmTSP has been designed in the form of weighted sum of the total traveling cost of all salesmen and the highest traveling cost of any single salesman. The proposed hybrid algorithm took into account the global information of the probability distribution of the cities and the local information in terms of trajectory of movements to perform the search. Furthermore, the utilization of the decomposition framework of multi-objective optimization succeeded in generating a set of promising tradeoff solutions in most of the instances of the MmTSP. This paper also showed that the decomposition algorithms scale well in both decision space and objective space. Comparative studies were carried out between the proposed algorithms and four state-of-the-art MOEAs, and the results indicated that REDA was able to achieve better results in larger problems and the hybridization with the EGS improved its performance.

References

[1] R. D. Angle, W. L. Caudle, R. Noonon, and A. Whinston, "Computer assisted school bus scheduling", *Management Science*, vol. 18, no. 6, pp. 278-288, 1972.

[2] H. A. Saleh and R. Chelouah, "The design of the navigation satellite system surveying networks using genetic algorithms", *Engineering Application of Artificial Intelligence*, vol. 17, no. 1, pp. 111-122, 2004.

[3] K. C. Gilbert and R. B. Hofstra, "A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem", *Decision Science*, vol. 23, no. 1, pp. 250-259, 1992.

[4] L. Tang, J. Liu, A. Rong, and Z. Yang, "A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron and Steel Complex", *European Journal of Operational Research*, vol. 124, no. 2, pp. 267-282, 2000.

[5] Y. Zhong, J. Liang, G. Gu, R. Zhang, and H. Yang, "An implementation of evolutionary computation for path planning of cooperative mobile robots", *Proceedings of the fourth world congress on intelligent control and automation*, vol. 3, pp. 1798-1802, 2002.

[6] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*, Chichester John Wiley & Sons, 2001.

[7] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications*, Springer-Verlag, 2005.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-97, 2002.

[9] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Computer Engineering and Network Laboratory, TIK-Report 103, Swiss Federal Institute of Technology Zurich.

[10] H. Ishibuchi, T. Yoshida, and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling", *IEEE Transaction on Systems, Man, and Cybernetics*, *Part C: Applications and Reviews*, vol. 28, no.3, pp. 204-223, 1998.

[11] Q. Zhang, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition", *IEEE Transaction on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

[12] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Norwell: MA: Kluwer, 2001.

[13] J. A. Lozano, P. Larrañaga, E. Bengoetxea, *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, Springer-Verlag New York, Inc., 2006.

[14] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multiobjective estimation of distribution algorithm," *Scalable Optimization via Probabilistic Modeling*, pp. 223-248: Springer, 2006.

[15] X. Zhong and W. Li, "A decision-tree-based multi-objective estimation of distribution algorithm," in *Proceedings of the 2007 International Conference on Computational Intelligence and Security,* pp. 114-118, 2007.

[16] H. Muhlenbein and G. Paass, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pp. 178-187, 1996.

[17] P. A. N. Bosman, and D. Thierens, "Continuous iterated density estimation evolutionary algorithm within the IDEA Framework," in *Workshop at the Genetic and Evolutionary Computation Conference*, pp. 197-200, 2000.

[18] H. J. Tang, V. A. Shim, K. C. Tan, and J. Y. Chia., "Restricted Boltzmann machine based algorithm for multi-objective optimization," *IEEE Congress on Evolutionary Computation*, pp. 1-8, 2010.

[19] L. Marti, J. Garcia, A. Berlanga, and J. M. Molina, "Solving complex high-dimensional problems with the multi-objective neural estimation of distribution algorithm," in *Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference,* pp. 619-626, 2009.

[20] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41-63, 2008.

[21] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures", *The International Journal of Management Science*, vol. 34, no. 3, pp. 209-219, 2005.

[22] T. Zhang, W. A. Gruver, and M. H. Smith, "Team scheduling by genetic search", in *Proceedings of the 2nd International Conference on Intelligent Processing and Manufacturing of Materials*, pp. 829-844, 1999.

[23] Y. B. Park, "A hybrid genetic algorithm for the vehicle scheduling problem with due times and times deadlines", *International Journal of Production Economics*, vol. 73, no. 2, pp. 175-188, 2001.

[24] A. E. Carter and C. T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms", *European Journal of Operational Research*, vol. 175, no. 1, pp. 246-257, 2006.

[25] F. Zhao, J. Dong, S. Li, and X. Yang, "An improved genetic algorithm for multiple traveling salesman problem", *Second International Asia Conference on Informatics in Control, Automation and Robotics*, pp. 493-495, 2008.

[26] A. Singh and A. S. Baghel, "A new grouping genetic algorithm approach to the multiple traveling salesperson problem", *Soft Computing*, vol. 13, no. 1, pp. 95-10, 2009.

[27] A. Kiraly and J. Abonyi, "Optimization of multiple traveling salesman problem by a novel representation based genetic algorithm", *10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, pp. 315-326, 2010.

[28] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," *ACM International Conference Proceeding Series*. pp. 791-798, 2007.

[29] M. Carreira-Perpinan, and G. Hinton, "On Contrastive Divergence Learning," in 10th Int. Workshop on Artificial Intelligence and Statistics, 2005.

[30] R. Solomon, "Evolutionary algorithms and gradient search: similarities and differences", *IEEE Transaction on Evolutionary Computation*, vol. 2, no. 2, pp. 45-55, 1998.

[31] C. K. Goh, Y. S. Ong, K. C. Tan, and E. J. Teoh, "An investigation on evolutionary gradient search for multi-objective optimization", *IEEE Congress on Evolutionary Computation*, pp. 3741-3746, 2008.

[32] W. T. Koo, C. K. Goh, and K. C. Tan, "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment", *Memetic Computing*, vol. 2, no. 2, pp. 87-110, 2010.

[33] C. Okonjo, "An effective method of balancing the workload amongst salesman", *Omega*, vol 16, no. 2, pp. 159-163, 1998.

[34] K. Miettinen, *Nonlinear Multiobjective Optimization*, Norwell, MA: Kluwer, 1999.

[35] V. A. Shim, K. C. Tan, and J. Y. Chia, "Probabilistic based evolutionary optimizers in bi-objective traveling salesman problem", in *Proceedings of the 8th international conference on simulated evolution and learning*, pp. 588-592, 2010.

[36] W. Peng, Q. Zhang, and H. Li, "Comparison between MOEA/D and NSGAII on the multi-objective traveling salesman problem", in *Multi-objective Memetic Algorithm*, vol. 171, pp. 309-324, 2009.

[37] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, Ph.D, Swiss Federal Institute of Technology, Zurich, 1999.