

Evolutionary Multi-objective Simulated Annealing with Adaptive and Competitive Search Direction

Hui Li and Dario Landa-Silva

Abstract—In this paper, we propose a population-based implementation of simulated annealing to tackle multi-objective optimisation problems, in particular those of combinatorial nature. The proposed algorithm is called Evolutionary Multi-objective Simulated Annealing Algorithm (EMOSA), which combines local and evolutionary search by incorporating two distinctive features. The first feature is to tune the weight vectors of scalarizing functions (i.e., search directions) for selection during local search using a two-phase strategy. The second feature is the competition between members of the current population with similar weight vectors. We compare the proposed algorithm to three other multi-objective simulated annealing algorithms and also to the Pareto archived evolutionary strategy (PAES). Experiments are carried out on a set of bi-objective travelling salesman problem (TSP) instances with convex or nonconvex Pareto-optimal fronts. Our experimental results demonstrate that the two-phase tuning of weight vectors and the competition between individuals make a significant contribution to the improved performance of EMOSA.

I. INTRODUCTION

Multi-objective optimisation problems (MOPs) arise in many real-life applications, such as engineering, finance, logistics, etc. The optimal solutions to MOPs are the best trade-offs among the objectives, known as Pareto optimal solutions. The number of Pareto-optimal solutions might be huge but in practice, decision-makers are usually interested in a set of finite Pareto-optimal solutions with good spread in the objective space (i.e., a good approximation to the Pareto-optimal front). Over the past twenty years, metaheuristics have gained increasing popularity in multi-objective optimisation [1]. One of the main advantages of these metaheuristics is that multiple Pareto-optimal solutions can be found in a single simulation run.

Simulated annealing (SA) is a stochastic local search based metaheuristic for global optimisation [2]. It is inspired by the physical process of heating and controlled cooling of a material. During local search, non-improving moves are allowed with some probability when the temperature is not zero. As a result, this method has a chance to escape local optima and find promising areas of the search space. As the temperature goes to zero, the probability of accepting only improving moves increases so that search in the promising areas is intensified.

A number of multi-objective simulated annealing algorithms (MOSAs) have also been developed to tackle MOPs [3][4][5][6][7][8][9]. Unlike single objective simulated annealing algorithms, the acceptance functions in these MOSAs

are based on either the aggregation of objectives (i.e., scalarization) or Pareto dominance. In order to find a diverse set of nondominated solutions, some MOSAs optimise multiple distinct weighted scalarizing functions, of which the weight vectors are either fixed or dynamically changed during the search procedure. Although these MOSAs maintain a population of solutions and optimise multiple scalarizing functions at the same time, each scalarizing function is optimised by one solution separately without cooperation with others.

A multi-objective evolutionary algorithm based on decomposition, called MOEA/D, was proposed in [10]. It optimises multiple subproblems (scalarizing functions) simultaneously by using an evolutionary algorithm. Each subproblem is associated with one solution, which is the best solution found previously regarding the related scalarizing function. The offspring solution for a certain subproblem is produced by recombining those solutions with similar weight vectors. During the replacement, the algorithm updates not only the current solution of this subproblem but also those neighbouring solutions. This is so-called cooperation between subproblems in MOEA/D, which has been proved to be very effective for solving benchmark MOPs with continuous or discrete search space [10].

Inspired by the strategies for maintaining a diverse set of weight vectors in previous MOSAs and cooperation between individuals with similar weight vectors in MOEA/D, we propose a multi-objective simulated annealing algorithm called Evolutionary Multi-objective Simulated Annealing (EMOSA). It combines the strengths of both local and evolutionary search. The main contributions of this paper are as follow.

- We propose a two-phase strategy for maintaining the diversity of search directions in EMOSA. This strategy uses both fixed (first phase) and adaptive (second phase) search directions during the search in order to examine the multi-objective space more effectively.
- We compare the performance of EMOSA and three other MOSAs and PAES [11] on a set of bi-objective TSP instances with convex or nonconvex Pareto-optimal fronts. We also assess the effectiveness of the proposed adaptive and competitive search direction mechanism and demonstrate that it significantly helps to improve the performance of the proposed algorithm.

The remainder of this paper is organised as follows. Section II introduces some basic definitions of multi-objective optimisation. In Section III, simulated annealing approaches for multi-objective optimisation are described. Section IV

The authors are with the Automated Scheduling, Optimisation and Planning Research Group, School of Computer Science, University of Nottingham, United Kingdom. (email: hzl@cs.nott.ac.uk; jds@cs.nott.ac.uk)

presents the proposed EMOSA algorithm. In the following section, multi-objective travelling salesman problems are formulated. In Section VI, the experimental results are presented and discussed. Finally, Section VII concludes the paper.

II. BASIC DEFINITIONS IN MULTI-OBJECTIVE OPTIMISATION

A multi-objective optimisation problem (MOP) with minimisation objectives can be stated as:

$$\begin{aligned} \text{minimize} \quad & F(x) = (f_1(x), \dots, f_m(x)) \\ \text{s.t.} \quad & x \in \Omega \end{aligned} \quad (1)$$

where x is the vector of decision variables, Ω is the feasible region of search space, $F \in \mathbb{R}^m$ is the vector of m objective functions $f_i : \Omega \rightarrow \mathbb{R}, i = 1, \dots, m$. If Ω consists of a discrete set of solutions, then the MOP in (1) is called a multi-objective combinatorial optimisation problem (MOCOP). MOCOPs can be very challenging due to huge search space, complex constraints, and many local optimal solutions.

For any two objective vectors u and v in \mathbb{R}^m , u is said to dominate v , denoted by $u \prec v$, if and only if $u_i \leq v_i$ for all $i \in \{1, \dots, m\}$ and there exists at least one index $j \in \{1, \dots, m\}$ satisfying $u_j < v_j$. For any two solutions x and y in Ω , x is said to dominate y if $F(x)$ dominates $F(y)$. A solution x^* is said to be Pareto-optimal (efficient) if no solution in Ω dominates x^* . The set of all Pareto-optimal solutions in Ω is called Pareto-optimal set. The objective vectors of all solutions in the Pareto-optimal set is called Pareto-optimal front, denoted by PF .

A Pareto-optimal solution x^* is supported if it is the unique global minimum of the following problem [12]

$$\begin{aligned} \text{minimize} \quad & g^{(ws)}(x, \lambda) = \sum_{i=1}^m \lambda_i f_i(x) \\ \text{s.t.} \quad & x \in \Omega \end{aligned} \quad (2)$$

where $\lambda = (\lambda_1, \dots, \lambda_m)$ is a normalized weight vector satisfying $0 \leq \lambda_i \leq 1, i = 1, \dots, m$ and $\sum_{i=1}^m \lambda_i = 1$. Optimising the problem in (2) is also called weighted sum approach. It is a commonly-used method for fitness assignment in many multi-objective approaches. However, the weighted sum approach cannot solve MOPs with nonconvex Pareto-optimal fronts [13]. In contrast, the weighted min-max approach can overcome this weakness [12]. It needs to find the optimal solution of the following problem.

$$\begin{aligned} \text{minimize} \quad & g^{(mm)}(x, \lambda) = \max_{i \in \{1, \dots, m\}} \lambda_i f_i(x) \\ \text{s.t.} \quad & x \in \Omega \end{aligned} \quad (3)$$

where λ is the same as above. Note that the property of the weighted min-max approach is similar to that of weighted Tchebycheff approach, which needs a reference point. Here, we call the functions in (2) and (3) weighted scalarizing functions. Many other scalarizing functions are also available [12].

III. MULTI-OBJECTIVE SIMULATED ANNEALING ALGORITHMS

The basic idea in simulated annealing is to reduce the possibility of getting trapped in local optima by allowing local search moves from a current solution to its inferior neighbours. In multi-objective optimisation, the acceptance functions are often chosen as weighted scalarizing functions $g(x, \lambda)$, such as weighted sum function $g^{(ws)}$ in (2) or weighted min-max function $g^{(mm)}$ in (3). Given a current solution x and its neighbourhood $N(x)$, the probability of moving x to its neighbour $x' \in N(x)$ can be defined by:

$$P(x, x', \lambda, T) = \begin{cases} 1 & \text{if } \Delta g(x, x', \lambda) < 0 \\ e^{-\frac{\Delta g(x, x', \lambda)}{T}} & \text{otherwise.} \end{cases} \quad (4)$$

where

- $T > 0$ is the temperature level.
- $\Delta g(x, x', \lambda) = g(x', \lambda) - g(x, \lambda)$ is the difference of acceptance function values between x and x' .

The temperature T changes as the search progresses. Initially, T is set to a high value. At this stage, many inferior neighbours could be accepted because the probability P is high. Then, T is decreased gradually to a low value during the search. Thus, the probability of allowing non-improving moves is also reduced gradually. At the final temperature value, nearly all inferior neighbours are rejected.

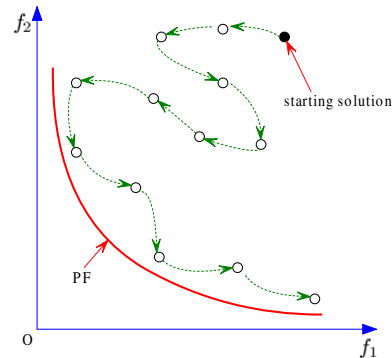


Fig. 1. Random search directions in Serafini's MOSA

Serafini [7] developed a simulated annealing approach for solving MOCOPs. Like many single objective simulated annealing algorithms, Serafini's MOSA is a single-point method. At each step, it optimises one weighted scalarizing function, which acts as the acceptance function in local search moves. To diversify the nondominated solutions found, the weight vector of scalarizing function is modified slightly and randomly during the search. Fig 1 illustrates the search direction in Serafini's MOSA.

Ulungu et al. [3] suggested a population-based MOSA, which optimises multiple scalarizing functions but each of them is optimized by a single SA run. To maintain the diversity of resultant nondominated solutions, a set of fixed evenly-distributed weight vectors are used. Unlike Serafini's

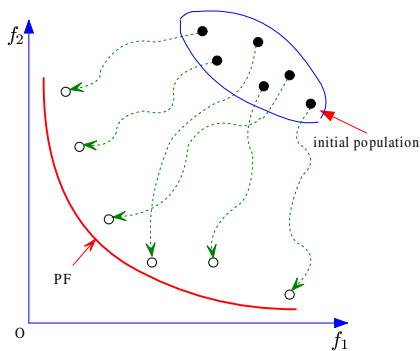


Fig. 2. Fixed search directions in Ulungu's MOSA

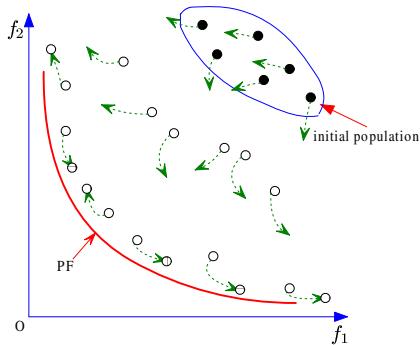


Fig. 3. Adaptive search directions in Czyzak's MOSA

MOSA, Ulungu's MOSA has an equal chance to optimise each weighted scalarizing function (see Fig 2).

Czyzak et al. [4] proposed a MOSA with adaptive search directions. It also uses a population of solutions to optimise multiple weighted scalarizing functions at the same time. To find the solutions in the unexplored area of the Pareto-optimal front, this approach adaptively tunes the weight vector of each solution during the search according to the closeness to its neighbours. Fig 3 illustrates the search direction in Czyzak's MOSA.

IV. THE PROPOSED EVOLUTIONARY MULTI-OBJECTIVE SIMULATED ANNEALING ALGORITHM

A. Motivations

Both Ulungu's and Czyzak's MOSAs are population-based algorithms and maintain multiple weight vectors during the search. The weight vectors in both algorithms are either fixed or adaptively changed during the whole search procedure. However, the strategies for maintaining multiple weight vectors in both algorithms have their weaknesses. Firstly, the optimal solutions of scalarizing functions with fixed weight vectors might not be able to cover the whole Pareto-optimal front well due to some reasons, such as the scalability of objectives, the shape of the Pareto-optimal front, and small population size. In this case, it is necessary to tune the search directions of solutions in order to find the nondominated

solutions in the unexplored area of the Pareto-optimal front. Secondly, it is not effective to tune the weight vectors locally when the current population is not close to the true Pareto-optimal front. The fluctuation of search directions in the early phase of the search may slow down the convergence speed. In this phase, more computational effort should be spent on finding solutions near the Pareto-optimal front instead of seeking a diverse set of solutions along the Pareto-optimal front.

It should be noted that no competition between members of the population is considered in both Ulungu's and Czyzak's MOSAs. That is, there is no cooperation between individuals during selection even if some of them are clearly superior to the others in terms of their related scalarizing functions. As studied in [10], the optimisation of a certain scalarizing function is helpful for that of its neighbouring scalarizing functions, and the competition between solutions with different scalarizing functions can benefit the convergence of the search. However, this issue was not considered in the MOSAs mentioned above.

B. The Description of EMOSA

Based on the above motivations, an evolutionary multi-objective simulated annealing algorithm (EMOSA) is proposed in this paper. A two-phase strategy for tuning the search direction of each solution is considered in EMOSA. The competition between members of population is also a key feature of the proposed algorithm. EMOSA maintains a population of solutions x^1, \dots, x^{pop} . Each solution x^i is associated with a weight vector λ^i . Here, we assume that the weighted sum approach is used in the selection of local search moves. The EMOSA approach is described in Fig 4.

In Step 2.1, a neighbouring solution x' is chosen from the neighbourhood $N(x^i)$ of x^i . It should be noted that the definition of neighbourhood is problem-specific. Step 2.2 updates NDS if x' is not dominated by the current solution x^i . All members in NDS dominated by x' are removed and x' is added to NDS if it is not dominated by any member of NDS. In Step 2.3, the neighbouring solution x' is accepted as the current solution with probability $P(x^i, x', \lambda^i, T)$. In Step 2.4, x' is also used to replace other similar members of the current population if x' is better with respect to the related scalarizing function. The similarity between solutions is measured by the Euclidean distance between their weight vectors. After local search moves are performed K times for each member of the population, the current temperature is modified in Step 3. Once the temperature has been reduced to the value below T_c , Step 4 modifies the weight vector of each solution in CS . The basic idea in this step is to move the members of CS away from their closest nondominated neighbours in the population. Maintaining the maximal size of NDS is optional. This is necessary when the number of Pareto-optimal solutions is too large or infinite (as in the case of continuous problems). The commonly-used strategies for maintaining the diversity of a set of solutions in multi-objective optimisation can be found in the literature [14].

Input : starting temperature value (T_0), final temperature value (T_{min}), temperature value for tuning weight vectors (T_c), temperature cooling rate (α), population size: (pop), number of local search moves for each solution between two consecutive temperature levels (K), radius of the neighbourhood of each weight vector (r_w).

Output: A set of nondominated solutions - NDS

Step 1: Initialization

Step 1.1 Generate the initial population CS of x^1, \dots, x^{pop} randomly and evaluate them;

Step 1.2 Produce pop distinct weight vectors $\lambda^1, \dots, \lambda^{pop}$ with uniform spread;

Step 1.3 Form NDS by the nondominated members of the initial population CS and set $T := T_0$.

Step 2: Population Evolution

foreach $i \in \{1, \dots, pop\}$ **do**

$c := 0$;

repeat

Step 2.1 Generate a neighbor $x' \in N(x^i)$ and evaluate it. Set $c := c + 1$;

Step 2.2 Update the external population NDS if x^i does not dominate x' ;

Step 2.3 Replace the current solution x^i by x' with probability $P(x^i, x', \lambda^i, T)$;

until $c \geq K$;

Step 2.4 Compete with similar members in CS as follows:

foreach $j \in \{1, \dots, pop\}$ **do**

Set $x^j = x^i$ if $g^{(ws)}(x^i, \lambda^j) < g^{(ws)}(x^j, \lambda^j)$ and $d(\lambda^i, \lambda^j) < r_w$.

end

end

Step 3: Temperature Change

Decrease the temperature by setting $T := T - \alpha$. If $T < T_c$, go to **Step 4**; otherwise, go to **Step 5**.

Step 4: Search Direction Adaption

foreach $i \in \{1, \dots, pop\}$ **do**

Find the closest nondominated neighboring solution \bar{x} of x^i from CS .

foreach $k \in \{1, \dots, m\}$ **do**

if \bar{x} exists **then**

If $f_k(x^i) < f_k(\bar{x})$, $\lambda_k^i = \min\{1, \lambda_k^i + \frac{1}{pop}\}$;
otherwise, $\lambda_k^i = \max\{0, \lambda_k^i - \frac{1}{pop}\}$

else

λ_k^i equals to $\min\{1, \lambda_k^i + \frac{1}{pop}\}$ or $\lambda_k^i = \max\{0, \lambda_k^i - \frac{1}{pop}\}$ with the probability 0.5.

end

end

Normalize λ^i by setting $\lambda_k^i = \lambda_k^i / \sum_{s=1}^m \lambda_s^i$,
 $k = 1, \dots, m$.

end

Step 5: Stopping Criteria

If $T < T_{min}$, stop and return NDS; otherwise, go to **Step 2**.

Fig. 4. The framework of EMOSA

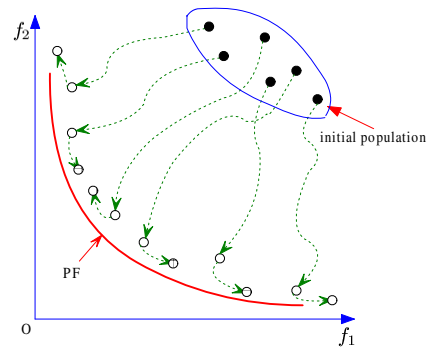


Fig. 5. Fixed-then-adaptive search directions in EMOSA

C. Connections Between EMOSA and Relevant Algorithms

- The main differences between EMOSA and other two population-based MOSAs (i.e., Ulungu's and Czyzak's MOSAs) lie in Step 2.4 and Step 4. Ulungu's MOSA can be regarded as a variant of EMOSA without Step 2.4 and Step 4. In Czyzak's MOSA, the change of weight vector is performed after each local search move instead of Step 4 and no Step 2.4 is needed.
- In EMOSA, a two-phase strategy for modifying weight vectors is adopted, which is the combination of the strategies in Ulungu's and Czyzak's MOSAs. In the first phase ($T \geq T_c$), all weight vectors remain unchanged. This is the same as in Ulungu's MOSA. In the second phase ($T < T_c$), weight vectors are adaptively changed according to the closeness to their nondominated neighbour in the population. This is the same as that in Czyzak's MOSA [15]. This two-phase strategy is illustrated in Fig 5.
- Similar to MOEA/D in [10], EMOSA also measures the similarity between x^i and x^j by the Euclidean distance between their weight vectors - λ^i and λ^j . That is, x^i is the neighbour of x^j if λ^j is the neighbour of λ^i (i.e., $d(\lambda^i, \lambda^j) < r_w$). EMOSA competes x^i with all x^j if their weight vectors λ^j are the neighbour of λ^i . However, all other MOSAs only compare x^i with x^j . In MOEA/D, the current solution of each subproblem is updated only if better offspring solutions are produced by recombining similar mating parents. In contrast, EMOSA generates new solution from the neighbourhood of current solution and accepts worse solution with some probability during the selection.

V. MULTI-OBJECTIVE TRAVELLING SALESMAN PROBLEM

The travelling salesman problem is a well-known NP-hard combinatorial optimisation problem [16]. It has been widely used to test the performance of many metaheuristics, such as simulated annealing, tabu search, ant colony algorithm, and memetic algorithm (e.g. [17][18]). The objective in TSP is to find the shortest Hamiltonian circuit, which visits each city only once. Recently, some researchers have shown some

interest in the multi-objective version of TSP [19][20][21]. Given a set $\{c_1, \dots, c_n\}$ of n cities, each pair $\{c_i, c_j\}$ of cities corresponds to a vector $(d_{i,j}^1, \dots, d_{i,j}^m)$ of distances. The objectives of multi-objective TSP can be formulated as:

$$\text{minimize } f_i(\phi) = \sum_{j=1}^{n-1} d_{\phi(j), \phi(j+1)}^i + d_{\phi(n), \phi(1)}^i \quad (5)$$

$$i = 1, \dots, m.$$

where ϕ is a permutation of $\{1, \dots, n\}$. In this paper, we only consider the symmetric multi-objective TSP satisfying $d_{i,j}^k = d_{j,i}^k$ for $1 \leq i, j \leq n$ and $1 \leq k \leq m$.

According to the results reported previously [20][21], the shapes of the Pareto-optimal fronts for the known bi-objective TSP instances are convex¹. To study the performance of the proposed algorithm for solving MOCOPs with nonconvex Pareto-optimal fronts, we suggest the variant of bi-objective TSP problems with the following objective functions:

$$\bar{f}_1(x, \phi) = A_1 t(x) + |B_1 f_1(\phi) + C_1(t) f_2(\phi) + D_1| \quad (6)$$

$$\bar{f}_2(x, \phi) = A_2 S(t(x)) + |B_2 f_1(\phi) + C_2(t) f_2(\phi) + D_2| \quad (7)$$

where

- $x = (x_1, \dots, x_d) \in \{0, 1\}^d$ is a binary string and ϕ is the permutation of $\{1, 2, \dots, n\}$.
- $t: \{0, 1\}^d \rightarrow [0, 1]$ is the function of x and $S: [0, 1] \rightarrow \mathbb{R}$ is the function of t . In this paper, we use

$$t(x) = \frac{\sum_{i=1}^d x_i \cdot 2^{i-1}}{2^d - 1}.$$

- $A_i > 0, B_i, D_i, i = 1, 2$ are constants, and $C_i(t), i = 1, 2$, are functions of t .

In the above bi-objective TSP, $S(t(x))$ is used to control the shape of the Pareto-optimal front. The values of $|B_1 f_1(\phi) + C_1(t) f_2(\phi) + D_1|$ and $|B_2 f_1(\phi) + C_2(t) f_2(\phi) + D_2|$ in a Pareto-optimal solution should be as small as possible, ideally equal to zero. In this case, each Pareto-optimal solution corresponds to the intersection points of two straight lines in the objective space.

In this paper, we consider the following two examples of $S(t(x))$:

- Pareto-optimal front with nonconvexity

$$S^{(1)}(t) = \begin{cases} \frac{2}{3} + \sqrt{\frac{1}{9} - t^2} & \text{if } 0 \leq t < \frac{1}{3} \\ \frac{2}{3} - \sqrt{\frac{1}{9} - (t - \frac{2}{3})^2} & \text{if } \frac{1}{3} \leq t < \frac{2}{3} \\ \sqrt{\frac{1}{9} - (t - \frac{2}{3})^2} & \text{otherwise.} \end{cases} \quad (8)$$

- Pareto-optimal front with clusters (i.e., discontinuity)

$$S^{(2)}(t) = 1 - 0.5\sqrt{t} - 0.5t \sin(10\pi t) \quad (9)$$

¹In this paper, the Pareto-optimal front is said to be convex if the shape of the front formed by all supported solutions is convex.

VI. EXPERIMENTAL STUDIES

A. Performance Metric

We use two metrics - inverted generation distance (IGD) [22] and hypervolume [14] to evaluate a set of nondominated solutions found by the algorithms under consideration.

- Inverted Generational Distance (IGD)

Given a reference set A^* , the IGD value of a set $A \subset \mathbb{R}^m$ is defined as:

$$IGD(A, A^*) = \frac{1}{|A^*|} \sum_{v \in A^*} \{\min_{u \in A} d(u, v)\} \quad (10)$$

where $d(u, v)$ is the Euclidean distance between u and v in \mathbb{R}^m . The IGD metric measures the average distance from the reference set A^* to the nearest solution in A . The lower the value of IGD, the better the quality of A . Since the true Pareto-optimal fronts of the bi-objective TSP test instances used in this paper are unknown in advance, A^* is alternatively formed by all nondominated solutions found by all approaches collected over all runs.

- Hypervolume (Ψ)

Given a reference point $y^* \in \mathbb{R}^m$, the hypervolume of a set $A \subset \mathbb{R}^m$ can be defined by:

$$\Psi(A, y^*) = \Lambda \left(\bigcup_{u \in A} \{y | u \prec y \prec y^*\} \right) \quad (11)$$

where Λ is the Lebesgue measure of a set. This metric describes the size of the objective space that is dominated by the points in A and dominates y^* . In this paper, y^* is chosen as $y_i^* = \max_{u \in PF} u_i, i = 1, \dots, m$. The larger the value of hypervolume, the better the quality of A .

B. Experimental Settings

- Test Instances

In our experiments, we used two 50-city bi-objective TSP test instances (KROAB50 and KROBC50) and two 100-city bi-objective instances (KROAB100 and KROBC100)². Two more test instances were created by incorporating KROAB50 and the functions in (8) and (9) into (6) and (7). We call these new instances KROAB50-A and KROAB50-B respectively. The dimensionality d of the binary string is 8. The constants in these two test instances are set to $A_1 = A_2 = 80000, B_1 = B_2 = 2, C_1(t) = -1 - 3t, C_2(t) = 2, D_1 = 0$, and $D_2 = 160000$.

- Algorithms in Comparison

We compared EMOSA with four other algorithms - Serafini's MOSA (SMOSA), Ulungu's MOSA (UMOSA), Czyzak's MOSA (CMOSA), and PAES. All algorithms except PAES are SA-based multi-objective algorithms. PAES uses Pareto dominance for its selection in local search moves while the others use scalarizing functions.

²The data of these four test instances can be found at <http://www.idss.cs.put.poznan.pl/~jaszkiewicz/>

We include PAES in the comparison because this is an evolutionary algorithm but also based on local search and regarded as successful in many difficult MOCOPs.

- Neighbourhood in Local Search

In our experiments, a neighbouring solution at each step is generated by exchanging the positions of two randomly selected cities in the tour (i.e., 2H-opt exchange). For two TSP test instances with nonconvex Pareto-optimal fronts, we apply the same 2H-opt exchange to mutate ϕ and bit-flip to modify the binary string x in the solution with probability 0.01.

- Initialisation of Weight Vectors

In the three population-based MOSAs, the set Δ of pop normalized weight vectors with uniform spread is generated as follows. Firstly, we create a large set Γ of 1000 random weight vectors and add one of them into Δ chosen at random. Then, the weight vector unselected in Γ with the largest distance to the set Δ is added into Δ . Repeat this process until the size of Δ equals to pop .

- Parameter Settings

The starting (T_0) and final (T_{min}) temperature values are set to 100 and 5, respectively, in four MOSAs. The cooling rate (α) for lowering the temperature level is 5. The number K of local moves between two consecutive temperature levels in three population-based MOSAs is 500 for 50-city test instances and 2500 for 100-city instances. The population size (pop) in these algorithms is 50 for all test instances. To have a fair comparison, the two non-population based algorithms - SMOSA and PAES use the same number of function evaluations as the three population-based MOSAs for each test instance.

In the proposed EMOSA, the temperature value (T_c) for tuning weight vectors is 50 and the radius (r_w) of the neighbourhood of each weight vector is 0.2. In PAES, 20×20 grids are used to estimate the density of each member in the external population containing nondominated solutions found during the search.

C. Results and Discussion

TABLE I
THE MEANS OF IGD-METRIC VALUES ($\times 10^4$)

Instance	EMOSA	UMOSA	CMOSA	SMOSA	PAES
KROAB50	0.2843	0.3015	1.2370	0.3934	1.2513
KROBC50	0.2272	0.3082	1.5338	0.5803	1.9063
KROAB100	0.4056	0.5443	4.6600	2.6470	2.1171
KROBC100	0.3617	0.5159	4.5885	2.6938	2.0313
KROAB50-A	0.3466	0.4331	3.2475	0.7883	2.0072
KROAB50-B	0.4442	0.4637	1.0885	0.3962	2.2717

1) *TSP Instances with Convex PFs*: The mean IGD values of the nondominated solutions found by each algorithm on KROAB50, KROBC50, KROAB100, and KROBC100 are summarized in Table I. The mean values of hypervolume metric of these solutions are provided in Table II. Figs 6 and

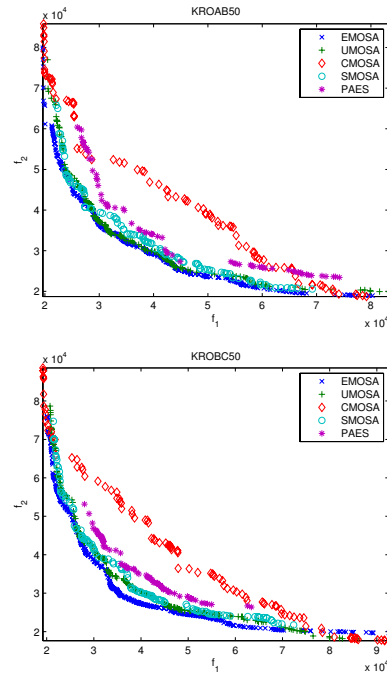


Fig. 6. The nondominated solutions found by all five algorithms in 20 runs on KROAB50 and KROBC50.

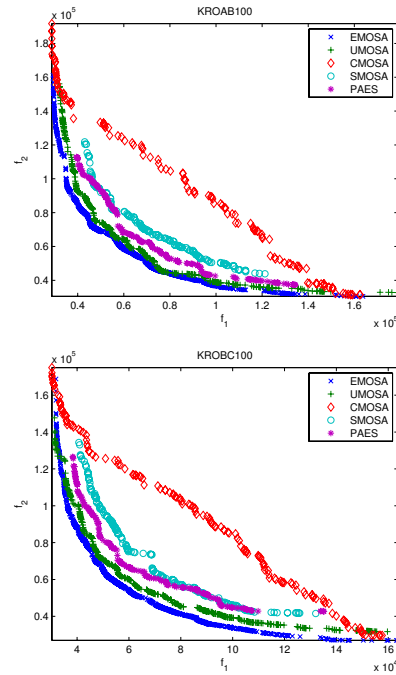


Fig. 7. The nondominated solutions found by all five algorithms 20 runs on KROAB100 and KROBC100.

TABLE II
THE MEANS OF HYPERVOLUME METRIC ($\times 10^{10}$)

Instance	EMOSA	UMOSA	CMOSA	SMOSA	PAES
KROAB50	0.2839	0.2827	0.2169	0.2743	0.1953
KROBC50	0.2809	0.2758	0.2085	0.2674	0.1771
KROAB100	1.9060	1.8641	1.3453	1.6081	1.5717
KROBC100	1.9392	1.8827	1.3986	1.6394	1.5758
KROAB50-A	0.1951	0.1932	0.0399	0.1731	0.1073
KROAB50-B	0.2139	0.2168	0.1645	0.2174	0.0931

7 show the nondominated solutions found by each algorithm after 20 runs on the 50-city and 100-city test instances.

The experimental results in Tables I and II show that EMOSA and UMOSA clearly perform better than the other algorithms in minimising the mean IGD and maximising the mean hypervolume. However, the performance of EMOSA is better than that of UMOSA on all test instances in terms of both metrics. The same observations can be made from the plots of nondominated solutions in Figs 6 and 7.

Among the four MOSAs, SMOSA and CMOSA are clearly the worst. The poor performance of SMOSA lies in the fact that it uses a single search direction at each step. It has no ability to approximate the whole Pareto-optimal front simultaneously. In contrast, the spread of nondominated solutions found by CMOSA on all test instances is satisfactory. However, CMOSA clearly performs worse than EMOSA and UMOSA in the convergence to the Pareto-optimal front. This might be caused by two reasons. Firstly, CMOSA changes the search direction for each member of the population at every local search move. We believe that this change of search directions is too frequent and unnecessary and it affects the convergence speed of the population towards the Pareto-optimal front. Secondly, individuals in the population do not compete amongst them during the search even if one individual is better than others regarding the related scalarizing functions. Since EMOSA takes the above factors into account during the search, its performance is significantly superior to the other algorithms considered in this paper on all test instances.

2) *TSP Instances with Nonconvex Pareto-optimal Fronts:* Fig 8 plots the nondominated solutions found by the five algorithms on KROAB50-A and KROAB50-B after 20 runs. It is evident that EMOSA and UMOSA are able to find the solutions in both the convex and the nonconvex parts of the Pareto-optimal fronts. This is mainly because both algorithms use an external population to store all nondominated solutions found along the search. Therefore, the nondominated solutions in the nonconvex part of the Pareto-optimal fronts found previously, can be retained in the external population. Fig 8 and the results in Table I and Table II, show that EMOSA and UMOSA also perform better than the other three algorithms on these two test instances.

3) *The Impact of Two-phase Strategy and Competition in EMOSA:* Do the two-phase strategy for tuning search directions and the competition between individuals in the population have a significant impact on the performance of MOSAs? To answer this question, we compared EMOSA

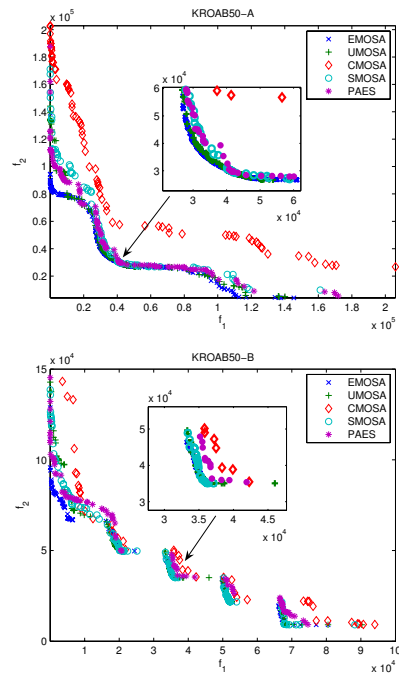


Fig. 8. Comparison of MOSAs with the weighted sum approach on KROAB50-A and KROAB50-B.

with its other variants on the KROAB50 instance. These variants are the versions of EMOSA without Step 2.4 (no competition) or Step 4 (fixed search directions). The weighted sum function was used in the selection for local search moves. All parameters remain same as before.

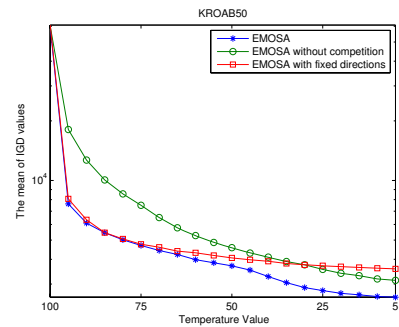


Fig. 9. Comparison of three versions of EMOSA on KROAB50.

Fig 9 plots the evolution of the mean IGD values of the nondominated solutions found by three versions of EMOSA vs the temperature levels. As we can see from this figure, EMOSA with fixed directions performs worst in minimising IGD values. This means that tuning the search directions in the second stage (i.e. after $T < T_c$) is very necessary in EMOSA. It is also evident that the two versions of EMOSA with competition between individuals converge much faster early in the search. It is clear that EMOSA with both features

performs best among the three versions. Therefore, we can conclude that both the competition between the members of population and the two-phase strategy for tuning the search directions help to improve the performance of the proposed EMOSA algorithm.

VII. CONCLUSIONS

We have proposed an evolutionary multi-objective simulated annealing algorithm, called EMOSA, which incorporates two distinctive features. One feature is a two-phase strategy for tuning search directions and the other feature is a competition scheme between individuals with similar search directions. We also compared the performance of the proposed EMOSA to other four algorithms (three MOSAs and PAES) on four benchmark bi-objective TSP test instances. Our experimental results demonstrated that EMOSA clearly outperforms all other algorithms on the TSP test instances with convex Pareto-optimal fronts. We have also suggested two variants of bi-objective TSP instances with nonconvex Pareto-optimal fronts. From our experimental results, we observed that EMOSA with the weighted sum approach solves the TSP instances with nonconvex Pareto front well. This is mainly because the solutions in the nonconvex part of the Pareto-optimal front are retained in an external population during the search. Furthermore, we showed that the two-phase strategy for tuning weight vectors and the consideration of competition between solutions do improve the performance of MOSAs substantially. In the proposed EMOSA, two parameters T_c and r_w were introduced. In future work, we intend to investigate the influence of these parameters on the performance of EMOSA and apply our method to solve many-objective optimization problems.

Acknowledgement- The work was funded by the UK's EPSRC, under grant EP/E019781/1. The authors would like to thank anonymous reviewers for their helpful and constructive comments.

REFERENCES

- [1] X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, *Metaheuristics for Multiobjective Optimisation*. Springer, 2004.
- [2] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [3] E. Ulungu, J. Teghem, P. Fortemps, and D. Tuytens, "MOSA method: A tool for solving multiobjective combinatorial optimization problems," *Journal of Multi-Criteria Decision Analysis*, vol. 8, no. 4, pp. 221–236, 1999.
- [4] P. Czyzak and A. Jaskiewicz, "Pareto simulated annealing - a metaheuristic technique for multiobjective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–37, Dec 1998.
- [5] E. K. Burke and J. D. Landa-Silva, "The influence of the fitness evaluation method on the performance of multiobjective search algorithms," *European Journal of Operational Research*, vol. 169, no. 3, pp. 875–897, 2006.
- [6] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *Journal of the Operational Research Society*, vol. 57, no. 10, pp. 1143–1160, 2006.
- [7] P. Serafini, "Simulated annealing for multiple objective optimization problems," in *Proceedings of the 10th International Conference on Multiple Criteria Decision Making*, vol. 1. Berlin: Springer-Verlag, 1994, pp. 283–294.
- [8] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Transactions on Evolutionary Computation*, 2008, in press.
- [9] K. Smith, R. Everson, J. Fieldsend, C. Murphy, and R. Misra, "Dominance-based multiobjective simulated annealing," *IEEE Transactions on Evolutionary Computation*, 2008, in press.
- [10] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [11] J. Knowles and D. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [12] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, USA: Kluwer Academic Publishers, 1999.
- [13] I. Das and J. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multi-criteria optimization problems," *Structural Optimization*, vol. 14, no. 1, pp. 63–69, 1997.
- [14] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK: John Wiley & Sons, 2001.
- [15] A. Jaskiewicz, "Multiple objective metaheuristic algorithm for combinatorial optimization," Ph.D. dissertation, Poznan University of Technology, 2001.
- [16] E. L. Lawler, J. K. Lenstra, A. Kan, and D. B. Shmoys, *The Travelling Salesman Problem*. Chichester, UK: John Wiley & Sons, 1985.
- [17] M. Dorigo and L. Gambardella, "Ant colony system: A cooperative learning approach to the travelling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [18] C. Voudouris and E. Tsang, "Guided local search and its application to the traveling salesman problem," *European Journal of Operational Research*, vol. 113, no. 2, pp. 469–499, 1999.
- [19] A. Jaskiewicz, "Genetic local search for multi-objective combinatorial optimization," *European Journal of Operational Research*, vol. 137, no. 1, pp. 50–71, 2002.
- [20] L. Paquete and T. Stützle, "A two-phase local search for the biobjective traveling salesman problem," in *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, Faro, Portugal, 2003, pp. 479–493.
- [21] C. Garcia-Martinez, O. Cordon, and F. Herrera, "A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP," *European Journal of Operational Research*, vol. 180, no. 1, pp. 116–148, 2007.
- [22] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.