Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 55 (2008) 439–449
computers & industrial engineering
www.elsevier.com/locate/caie

A memetic random-key genetic algorithm for a symmetric multi-objective traveling salesman problem

Funda Samanlioglu [a,*], William G. Ferrell Jr. [b], Mary E. Kurz [b]

[a] Kadir Has University, Department of Industrial Engineering, Kadir Has Campus, 34083 Cibali, Istanbul, Turkey
[b] Clemson University, Department of Industrial Engineering, 110 Freeman Hall, Clemson, SC 29634-0920, USA

Received 3 May 2007; received in revised form 6 January 2008; accepted 9 January 2008
Available online 17 January 2008

Abstract

This paper proposes a methodology to find weakly Pareto optimal solutions to a symmetric multi-objective traveling salesman problem using a memetic random-key genetic algorithm that has been augmented by a 2-opt local search. The methodology uses a "target-vector approach" in which the evaluation function is a weighted Tchebycheff metric with an ideal point and the local search is randomly guided by either a weighted sum of the objectives or a weighted Tchebycheff metric. The memetic algorithm has several advantages including the fact that the random keys representation ensures that feasible tours are maintained during the application of genetic operators. To illustrate the quality of the methodology, experiments are conducted using Euclidean TSP examples and a comparison is made to one example found in the literature.

Keywords: Multi-objective traveling salesman problem; Random-key; Memetic algorithms; Hybrid algorithms; Genetic algorithms

1. Introduction

The Traveling Salesman Problem (TSP) is a widely studied problem in the combinatorial optimization literature. The goal of the TSP is to find a tour that begins in a specific city, visits each of the remaining cities exactly once, and returns to the initial city such that some objective function is optimized, typically involving minimizing a function such as total distance traveled, total time, or total cost. In the multi-objective TSP (mo-TSP), simultaneous optimization of distances, costs, times, or other relevant objectives is required. In this research, the symmetric case is considered where the distance, time, or cost between cities is known and symmetric.

Determining whether a solution is inefficient is a non-deterministic polynomial-time complete (NPC) task for many multi-objective combinatorial optimization problems. Single-objective shortest path problem and

* Corresponding author. Tel.: +90 212 533 6532.
E-mail addresses: fsamanlioglu@khas.edu.tr (F. Samanlioglu), fwillia@ces.clemson.edu (W.G. Ferrell Jr.), mkurz@clemson.edu (M.E. Kurz).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2008.01.005

minimum spanning tree problem are both polynomially solvable problems. However, multi-criteria versions of these problems belong to the class of NPC problems even for two objectives (Hansen, 2000). TSP is NP-hard even with single objective. The multi-criteria version has the difficulty of the TSP itself and the difficulty of multiple objectives (Ehrgott, 2000). Therefore, for mo-TSP, heuristic methods which provide sub-optimal solutions are widely used. The literature contains many such approaches for addressing the mo-TSP such as tabu search (Hansen, 2000) and genetic algorithms (GA) (Jaszkiewicz, 2002). It is noted, particularly for larger TSP problems, that combining approaches can enhance the ability to find good solutions. Methods that combine domain-specific local search and evolutionary algorithms have received special attention and are called memetic algorithms (Merz & Freisleben, 2001). In this research, we combine a 2-opt local search (2-opt) with a GA having been inspired by Jaszkiewicz (2002). In the research, GA is allowed to work in the search space of local minima (with respect to the local search implemented) and diversity is preserved via different genetic operators to avoid trapping.

To be more precise, the proposed methodology combines a random-keys genetic algorithm (RKGA) with 2-opt local search to find good solutions to the symmetric mo-TSP. RKGA was introduced by Bean (1994) and was found particularly useful in manufacturing cell formation problems (Goncalves & Resende, 2004) and sequencing and scheduling problems (Kurz & Askin, 2004; Norman & Bean, 2000). This, along with Snyder and Daskin's (2006) application to the generalized TSP problem, provided initial motivation for using an RKGA in the mo-TSP setting. Preliminary results were quite promising as reported in Samanlioglu, Kurz, Ferrell Jr., and Tangudu (2007) for the symmetric TSP and Samanlioglu, Kurz, and Ferrell Jr. (2006) for the mo-TSP. In addition to a much more extensive look at the problem, this preliminary work has led to a number of fundamental modifications in the methodology including: a set of swap mutations for the inversion, and a different 2-opt implementation – a randomly selected guide for the 2-opt in which a weighted sums or a weighted Tchebycheff metric is applied with equal likelihood. Details of these are provided when the mo-TSP and the methodology are discussed.

In summary, the proposed methodology utilizes a novel memetic algorithm to address the mo-TSP problem. To effectively explain this methodology, we now address the basic idea of the RKGA and identify the unique aspects that make it particularly amenable to the mo-TSP. We then provide a brief discussion of the critical ideas related to the mo-TSP that directly relate to the proposed methodology that is presented in Section 4. Finally, experimental results are shown and comparisons made with the literature.

2. Random-key genetic algorithm

GAs were introduced by Holland in the 1970s (Holland, 1975). These algorithms mimic the concepts of biological evolution to develop solutions to complex real world problems. GAs have been used in many applications of TSP and its extensions throughout the literature (Cheng & Gen, 1994; Cheng, Gen, & Sasaki, 1995; Kubota, Fukuda, & Shimojima, 1996).

The main idea behind GAs is to start with randomly generated solutions and implement the "survival of the fittest" strategy in order to evolve to increasing better solutions through generations. A typical GA process consists of initial population generation, fitness evaluation, chromosome selection, applying genetic operators such as mutation, immigration, inversion, and crossover for reproduction, and termination. A particularly nice introduction to GAs is given in Goldberg's book (Goldberg, 1989).

An important issue in designing a GA is the chromosomal representation of a solution. Chromosomes are strings of numbers which represent the solution of the problem or can be decoded to represent the solution. Sometimes these numbers are 0s and 1s but other possibilities exist like strings of non-negative integers. When applying GAs to sequencing problems like scheduling and TSP, one of the key genetic operators is crossover. The simplest form of crossover is to select two chromosomes from the population, randomly select a point to split each into two pieces, and splice the front end of the first chromosome with the complementary end from the other and vice versa to form two different chromosomes, each of correct length. To illustrate this in TSP, assume that we have two chromosomes randomly selected from the population representing two different tours of a salesman starting at a city and visiting six other cities.

These tours are $1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7$ and $5 \to 4 \to 3 \to 6 \to 1 \to 2 \to 7$ as seen in Fig. 1. Assume that each tour is split at a randomly selected point, for example after the third city visited, and the front end of

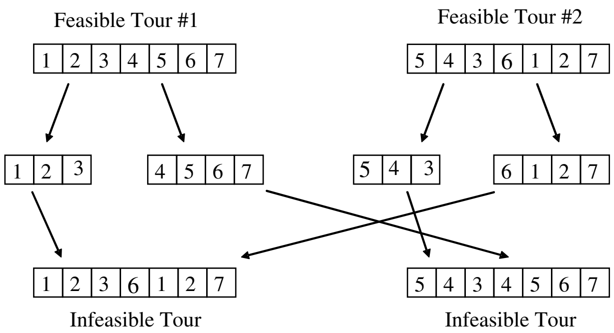

Fig. 1. Example of a simple crossover that creates infeasible tours.

the first tour is spliced with the complementary end of the second tour, and vice versa. The resulting tours are $1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 7$ and $5 \rightarrow 4 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$. These tours are infeasible since in each tour some cities are visited twice and others are not visited at all. This is a common occurrence for TSP; however feasibility can be maintained in a number of ways. Some have proposed "repair algorithms" to recreate feasible tours; however, repair algorithms can consume a considerable amount of time and can inhibit convergence (Michalewicz, 2000). A better alternative is to use an alternative chromosomal representation like the one introduced by Bean (1994) in which a random numbers encoding structure is used, resulting in the so-called Random Keys GA (RKGA). The structure proposed by Norman and Bean (1999) for a multiple machine scheduling problem was to assign a real number to each job. The part of the number to the left of the decimal was used to assign the machine and the part to the right of the decimal was used to assign the job sequence.

In this research, the initial population is created by randomly generating N (population size) chromosomes, each consisting of T genes that correspond to T cities in the problem using a uniform distribution between A and B (in our experiments $A = 1$, $B = 3000$). The first random number created for a chromosome is for the first city, second number is for the second city... and nth number is for the nth city. These random numbers are then used as "sort keys" to decode the solution. To decode a chromosome, cities are sorted in the ascending order of their corresponding keys to indicate the travel order of a salesman. For example, for a seven-city ($T = 7$) problem, the random keys encoding of a chromosome $C_1 = \{2, 12, 15, 29, 30, 45, 52\}$ decodes as the tour $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ and another chromosome $C_2 = \{24, 32, 13, 10, 4, 22, 48\}$ decodes as the tour $5 \rightarrow 4 \rightarrow 3 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 7$. Note that, these tours are the same ones shown in Fig. 1. All genetic operators are then executed on the random keys encoding, not on the decoded tour, so that all resulting chromosomes decode to produce feasible solutions. For example, if we apply the same crossover operation

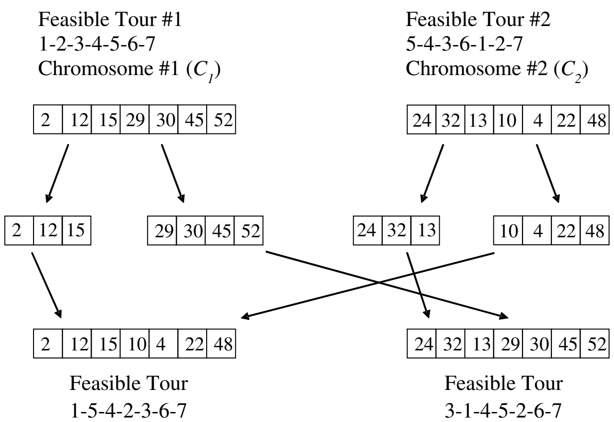

Fig. 2. Example of a simple crossover in RKGA.

implemented in Fig. 1 to C_1 and C_2, since crossover is implemented to random keys, not the decoded tours, the resulting tours will be feasible as seen in Fig. 2. Here, RKGA eliminates the infeasibilities in the algorithm without the usage of "repair algorithms."

3. Multi-objective TSP

A multi-objective program (MOP)

$$\begin{aligned} &\min f(x) = \{f_1(x), f_2(x), \ldots, f_k(x)\} \\ &st \qquad x \in X \end{aligned} \tag{1}$$

is assumed to have $k\ (k \geqslant 2)$ competing objective functions ($x \in \Re^n$, $f_i : \Re^n \rightarrow \Re$) that are to be minimized simultaneously.

Definition: A decision vector $x^* \in X$ is *efficient* (*Pareto optimal*) for MOP if there does not exist a $x \in X$, $x \neq x^*$ such that $f_i(x) \leqslant f_i(x^*)$ for $i = 1, \ldots, k$ with strict inequality holding for at least one index *i*. ($x^* \in X$ is efficient, $f(x^*)$ is non-dominated.)

Definition: A decision vector $x^* \in X$ is *weakly efficient* (*weakly Pareto optimal*) for MOP if there does not exist a $x \in X$, $x \neq x^*$ such that $f_i(x) < f_i(x^*)$ for $i = 1, \ldots, k$. ($x^* \in X$ is weakly efficient, $f(x^*)$ is weakly non-dominated.)

In practical mo-TSP applications, there might be several competing objective functions that correspond to cost factors related to distance, expenses, travel time, degree of risk, energy consumption, and other relevant considerations for the tour. The dimensionality of the objective space along with other factors such as comparability of solutions and hybridization with local search greatly affects the selection of evolutionary multi-objective optimization approach that will be implemented to find (weakly) Pareto optimal solutions to mo-TSPs. Existing evolutionary multi-objective optimization approaches can be generally classified as Pareto-based techniques and non-Pareto based techniques.

In non-Pareto based techniques like VEGA (Schaffer, 1984) and the Target Vector Approach (Coello Coello, 2001), the selection does not rely directly on the concept of Pareto dominance and Pareto ranking. In Pareto-based techniques such as NSGA II (Deb, Pratap, Agarwal, & Meyarivan, 2002) and SPEA II (Zitzler, Laumanns, & Thiele, 2001), the selection depends on Pareto dominance and ranking induced by dominance relation. This idea was first introduced by Goldberg (1989). A basic example to these techniques is the multi-objective GA of Fonseca and Fleming (1993) where the rank of a given solution is equal to the number of solutions that dominate it. The main advantages of Pareto ranking based methods are to be able to avoid the necessity of normalizing objective functions, setting reference points and specifying weighting coefficients (w_i) which represent relative importance given to each objective function. However, there are also disadvantages of Pareto based techniques as mentioned in Jaszkiewicz's (2002) and Knowles and Corne's (2004) research. One disadvantage is that Pareto ranking is not well suited for hybridization with local search since many local moves may not influence the rank of a solution. In some cases, change of a rank of a solution may require significant changes in the objective function values, and this might not be possible with local moves. Also, for solutions which are already ranked 1 (efficient solutions), local improvement is not possible (Jaszkiewicz, 2002). Another disadvantage is about the comparability of solutions. In 2004, Knowles and Corne (2004) suggested that in a MOP if there are only two or three objective functions, the dominance-ranking-based methods may be the most appropriate; however, if the number of objectives is four or more, Pareto selection may cause problems since many solutions will be incomparable.

In this research RKGA is hybridized with local search (2-opt). Moreover, presented memetic algorithm is intended for up to five or more objective-TSPs. Based on these facts, this research uses a non-Pareto based approach that does not directly rely on Pareto dominance or Pareto ranking called – Target Vector Approach.

In Target Vector Approach, each objective vector is assigned a goal or target vector. The evolutionary algorithm seeks to minimize the "distance" between the generated solution and the target vector with distance defined by one of several different metrics. In this methodology, as the evaluation function of RKGA, a weighted Tchebycheff metric is used with the ideal point as the reference point or target vector.

If z_i'' is the reference point and we use the ideal point $z_i'' = \min_{x \in X} f_i(x)$ as the reference point, the general weighted L_p-metric ($1 \leqslant p < \infty$) is defined as

$$\begin{aligned} \min & \left(\sum_{i=1}^{k} w_i |f_i(x) - z_i''|^p \right)^{1/p} \\ st & \quad x \in X. \end{aligned} \tag{2}$$

We assume that $w_i \geqslant 0$ for all $i = 1, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$, where the w_i's are weighting coefficients provided by the decision maker. If $p = \infty$, problem (2) reduces to a "weighted Tchebycheff program" (Bowman, 1976)

$$\begin{aligned} \min & \max_{i=1,\ldots,k} [w_i |f_i(x) - z_i''|] \\ st & \quad x \in X. \end{aligned} \tag{3}$$

If the reference point is the global optimal solution for $f_i(x)$ then the absolute value signs in problem (3) can be removed (Miettinen, 1999) yielding

$$\begin{aligned} \min & \max_{i=1,\ldots,k} \{w_i (f_i(x) - z_i'')\} \\ st & \quad x \in X. \end{aligned} \tag{4}$$

The solution of problem (2) is non-dominated if the solution is unique or if all the weighting coefficients are positive (Yu, 1973). Note that problem (2) coincides with the "weighted sums" or "weighting" problem if $p = 1$. Convexity of the feasible objective space is needed in order to find all non-dominated solutions using problem (2). If the feasible objective space is not convex, with (2) only supported efficient solutions which lie in the convex hull of the Pareto front can be found, however non-supported efficient solutions which lie in the non-convex portions of the Pareto front can not be found.

The solution of problem (3) is guaranteed weakly non-dominated for positive weights and at least one non-dominated solution is also guaranteed. If the solution is unique, then it is non-dominated; however, if the solution is not unique, then it might be weakly non-dominated (Wierzbicki, 1986). The main advantage of implementing weighted Tchebycheff function (3) or (4) over (2) or specifically "weighted sums" function is to be able to generate non-supported efficient solutions as well as supported efficient solutions in the mo-TSP since the feasible objective space is not necessarily convex.

The weighted Tchebycheff function has been used as an evaluation function for mo-TSP in Hansen (2000) and it has been used as part of a quality measure in Jaszkiewicz (2002). Hansen (2000) combined tabu search and local search to solve mo-TSPs and claimed that better results are obtained when using weighted Tchebycheff functions if the local search is guided by a weighted sums function rather than a weighted Tchebycheff metric. Jaszkiewicz (2002) presented a multi-objective genetic local search algorithm for mo-TSP combining distance preserving crossover with local search. He referred to Hansen's results and implemented a weighted sums evaluation function while guiding the local search also with a weighting problem.

In this research, we also use a weighted Tchebycheff function as the evaluation function in RKGA, however, the way 2-opt is implemented is different from existing research. Here, 2-opt search is guided by random selection of weighted sums or a weighted Tchebycheff function with equal chances of implementation. Random selection occurs every time 2-opt is implemented. If weighted Tchebycheff function is selected as a result of random selection, 2-opt uses weighted Tchebycheff function for evaluation, so 2-opt moves are acceptable iff they provide a tour with a better weighted Tchebycheff function. On the other hand, if weighted sums function is selected as a result of the random selection, 2-opt uses weighted sums function for evaluation, so 2-opt moves is acceptable iff they provide a tour with a better weighted sums function. The reason for this approach is that we speculate each function will guide the search differently. The weighted Tchebycheff function will keep the search generally aligned with the RKGA's weighted Tchebycheff evaluation function and increase the chance of finding non-supported efficient solutions. Weighted sums function, on the other hand, will be easier to optimize and will have higher chance for finding new potentially efficient solutions as a result of mo-TSPs relatively smooth non-dominated set as mentioned in Jaszkiewicz (2002). Note, however, that we do not claim this approach will work the best for other multi-objective combinatorial optimization problems.

However, in general, we speculate that if we have no information or partial information about the global shape of the non-dominated set of a MOP, the presented random selection method might be a good alternative to find part of the non-dominated set.

4. Methodology

The genetic operators used in this research are elitist reproduction, immigration, parameterized uniform crossover with tournament selection and swap mutation, similar to those in Bean's (1994), Norman and Bean's (1999), Ryan, Azad, and Ryan's (2004) and Samanlioglu, Kurz, Ferrell Jr., and Tangudu's (2007) research. A set of swap mutations is used in this research versus the inversion operator used in our previous research (Samanlioglu, Kurz, & Ferrell Jr., 2006; Samanlioglu et al., 2007) since this provides better exploration of the solution space and diversification in the population. Inversion is simply cutting out a sequence and replacing in reverse order so only two edges are replaced. In the classic swap mutation operation (Ryan, Azad, & Ryan, 2004), two cities are selected at random along the chromosome and their keys are swapped. In our implementation, s cities are selected at random along the chromosome, all the possible permutations of the keys of these s cities are created (implementing 2-opt after each permutation separately), and the best found chromosome is kept. We repeat this procedure k times for the same chromosome.

Since RKGA is an iterative procedure, a stopping criterion is utilized: a specified number of generations (G). The basic steps are:

1. Randomly generate the initial population of N chromosomes representing N feasible tours and apply 2-opt to all chromosomes.
2. Perform an elitist reproduction strategy whereby $e\%$ of the best chromosomes is copied to the next generation. The number of tours in the population of the next generation is now $e\% * N$.
3. Take duplicates of the tours in step 2 and perform a set of swap mutations to s genes for k times (applying 2-opt after each permutation of these s genes separately). Add these to the population of the next generation which now has $2e\% * N$ tours.
4. Perform parameterized uniform crossover with tournament selection (crossover probability $c\%$) and 2-opt to $p\%$ of the previous population. Copy these chromosomes to the next generation which now has $(2e\% + p\%)^* N$ tours.

In parameterized uniform crossover with tournament selection and 2-opt, two chromosomes are randomly selected from the population. For each gene corresponding to a city for the TSP problem, a random number (between 0 and 1) is generated. If the value is less than or equal to the crossover probability ($c\%$), the gene from the first chromosome is copied to the first new chromosome and the gene from the second chromosome is copied to the second new chromosome. If the random number is greater than the crossover probability, the genes are swapped. 2-opt is applied to both new chromosomes and as a result of the tournament selection, the one with the better evaluation function is included in the population of the next generation. The random keys representation ensures that feasible tours are constructed during the application of particularly this genetic operator, so no "repair" algorithm is needed to move back to the search space.

5. Use immigration to generate the remaining $i\%$ of the population associated with the next generation. Immigration is used to ensure diversity. Here, new chromosomes are randomly generated, implementing the same technique while initializing the population. Apply 2-opt to these chromosome(s) and copy them to the population of the next generation. The next generation is now complete with N tours since $2e\% + p\% + i\% = 100\%$.
6. If the stopping criterion is satisfied, go to step 7, otherwise, go to step 2.
7. Present the best tour and terminate the algorithm.

Note that, 2-opt is applied to all chromosomes at the initialization process and also after implementation of each genetic operator.

5. Computational experiments

The algorithm is implemented in C++ and tested using Condor High Throughput Computing Software (Condor), a distributed system that delivers the ideas of grid computing by harnessing the power of any available computer throughout the campus, and matching these resources to jobs through a flexible mechanism. It enables scientists to easily perform large-scale computations, however for CPU time comparisons it is not suitable since each experiment is done in a different available computer that might have different specifications depending on the computer availability. Based on this limitation, CPU times of the computational experiments are not presented.

The data for the computational experiments are taken from TSPLIB (Reinelt, 1995) and involve five problems, each containing 100 cities (kroA100, kroB100, kroC100, kroD100, and kroE100) with Euclidean distances. We created 2-, 3-, 4-, and 5-objective problems by combining these five problems. For example, to create a 3-objective problem, we used kroA100, kroB100 and kroC100 (kroABC100) as the first, second, and third objectives, respectively. Optimal solutions for each problem (kroA100, kroB100, kroC100, kroD100, and kroE100) are listed in TSPLIB, so the ideal point was readily available for use in the created mo-TSPs.

We have compared our best results with that found in Hansen (2000) for 2-, 3-, and 5-objective problems. To duplicate the situation in Hansen's research, the computational experiments all use equal weighting coefficients (w_i in problems (2) and (4)). All computational results are obtained with parameters $s = 4$, $k = 10$, and (e%, p%, c%, i%) = (20%, 59%, 70%, 1%). To be able to compare with Hansen's research, in each experiment, population size (N) and stopping criterion (G) are adjusted to allow approximately 10,000 evaluations in the algorithm using the formulae $G = \frac{10000-N}{3.19N}$. Here, $10000 - N$ represents the total number of evaluations reduced by initial population evaluations. $3.19N$ represents swap mutation evaluations $(20\%^*(k = 10))^*N$ plus parameterized uniform crossover with tournament selection evaluations $(59\%^*(\text{tournament selection} = 2)^*N$ plus immigration evaluations $(1\%)^*N$.

In Tables 1 and 2 best results obtained using the proposed methodology are given for 2-, 3-, 4-, and 5-objective problems along with Hansen's results. Note that Hansen did not present results for the 4-objective problems. In Table 2, best results of all 4-objective combinations of these five problems (kroA100, kroB100, kroC100, kroD100, and kroE100) are presented. For the 2- objective problem (kroAB100) the proposed methodology matched that of Hansen so only one solution is presented in Table 1. The 3- objective problem (kroABC100) and the 5-objective problem (kroABCDE100), however, produced different results from Hansen's as reflected in Table 1. As seen in this table, the best results found using the proposed methodology for kroABC100 and kroABCDE100 were superior to those found by Hansen since the evaluation functions (4) were smaller in value.

Table 1
Best results of Hansen (Hansen, 2000) and the proposed methodology for the 2-objective problem (kroAB100), 3-objective problem (kroABC100) and the 5-objective problem (kroABCDE100)

	Problem					
	1-Objective problems	kroAB100	kroABC100		kroABCDE100	
		Proposed methodology[a]	Hansen	Proposed methodology	Hansen	Proposed methodology
Cost in kroA100	21,282	49,771	67,274	67,202	85,412	85,604
Cost in kroB100	22,141	50,652	68,054	68,073	86,254	86,536
Cost in kroC100	20,749		66,751	66,557	85,018	84,795
Cost in kroD100	21,294				85,753	85,734
Cost in kroE100	22,068				86,177	86,499
$w_i = 1/k$		(1/2, 1/2)	(1/3,1/3,1/3)		(1/5,1/5,1/5,1/5,1/5)	
Evaluation function (4)		14,256	15,334	15,311	12,892	12,888

[a] Same result of kroAB100 was obtained in Hansen's research (Hansen, 2000).

Table 2
Best results found with the proposed methodology for five 4-objective problems

Cost in	Problem				
	kroABCD100	kroABCE100	kroABDE100	kroACDE100	kroBCDE100
kroA100	78,246	78,407	77,961	77,401	
kroB100	78,870	79,240	78,915		78,503
kroC100	77,710	77,637		76,946	77,497
kroD100	78,197		78,090	77,526	77,813
kroE100		79,234	78,782	78,418	78,744
$w_i = 1/k$			(1/4, 1/4, 1/4, ¼)		
Evaluation function (4)	14,241	14,292	14,199	14,088	14,187

Relative excess over the best known solution is calculated as

$$\text{Relative Excess} = \frac{\text{Evaluation function of the RKGA tour} - \text{Evaluation function of the best known RKGA tour}}{\text{Evaluation function of the best known RKGA tour}}$$

In Table 3, average relative excess (ARE) in percent of 30 replications of the proposed methodology along with number of replications (R) best solutions are obtained (out of these 30 replications, the ones that have relative excess = 0) are presented for 2-, 3-, 4-, and 5-objective problems. Here, five different settings are used (N = 25, 50, 100, 500, 1000) for these experiments. Approximately 10000 evaluations are allowed so for each population size (N), stopping criterion (G) is calculated based on the formulae given before.

These results indicate that the proposed methodology is able to perform well on average for the tested 2-, 3-, 4-, and 5- objective problems. Average relative excess never exceeded 0.5021% and 0.7230% for the 2-objective (kroAB100) and 3-objective (kroABC1000) problems, respectively. In both of these problems and in two of the 4-objective problems (kroABCE100, kroACDE100) best known solutions were found more than once during the computational experiments. Maximum average relative excess was 0.8892%, and 1.0053%, respectively for the 4- and 5-objective problems. Consistent with intuition, in our experiments, we observed that maximum average relative excess increased as the number of objectives increased.

Hansen (2000) presented the median of deviation over 30 tabu search repetitions from his best found solutions in percent for 3 mo-TSP problems (kroAB100, kroABC100, and kroABCDE100). He allowed 10,000 neighborhood moves in each repetition. He used several substitute scalarizing functions to guide local search (2-opt) even though the evaluation function remained as weighted Tchebycheff function, and claimed that better results are obtained in that case. We compared our evaluation function results with Hansen's by guiding 2-opt with weighted sums ($p = 1$ in program (2)) and weighted Tchebycheff ($p = \infty$ in program (2)) functions. When two-opt is guided by a weighted Tchebycheff function, a 2-opt move is conducted only if this move is going to provide a tour with a better weighted Tchebycheff function than the previous tour. On the other hand, when 2-opt is guided by a weighted sums function, a 2-opt move is acceptable only if this move is going

Table 3
Average relative excess (ARE) of 30 replications of the proposed methodology in percent and number of replications best solution is found (R) for five different settings

Problem	$N = 25$ $G = 125$		$N = 50$ $G = 62$		$N = 100$ $G = 31$		$N = 500$ $G = 6$		$N = 1000$ $G = 3$	
	ARE	R	ARE	R	ARE	R	ARE	R	ARE	R
kroAB100	0.4520	3	0.4003	2	0.3328	6	0.3140	3	0.5021	0
kroABC100	0.7183	1	0.7230	0	0.5851	1	0.5744	0	0.6976	1
kroABCD100	0.6873	0	0.5597	0	0.5242	1	0.5284	0	0.6855	0
kroABCE100	0.7850	1	0.6460	0	0.5337	3	0.6968	0	0.8892	0
kroABDE100	0.7013	1	0.7415	0	0.7471	0	0.6955	0	0.8117	0
kroACDE100	0.7394	1	0.5829	1	0.5294	0	0.5646	0	0.6019	0
kroBCDE100	0.7131	0	0.5494	1	0.5806	0	0.6027	0	0.8194	0
kroABCDE100	0.9469	1	1.0027	0	0.9127	0	0.8985	0	1.0053	0

Table 4
Median of deviation over 30 replications of the proposed methodology from Hansen's (Hansen, 2000) best found solutions in percent and related Hansen's results

Problem	Hansen's tabu search		Proposed methodology	
	$p=\infty$ in program (2)	$p=1$ in program (2)	$p=\infty$ in program (2)	$p=1$ in program (2)
kroAB100	1.9	0.4	0.4	0.4
kroABC100	2.1	0.6	1.3	0.4
kroABCDE100	2.2	0.6	1.8	0.7

to produce a tour with a better weighted sums function than the previous tour. Here, the general evaluation function remains as a weighted Tchebycheff function but just to evaluate the acceptability of 2-opt moves, a weighted sums function is used. We allowed 10,000 evaluations to be consistent with Hansen's research and presented median of deviation over 30 replications of the proposed methodology from Hansen's best found solutions in percent in Table 4. Note that CPU times are not presented here because of the limitations due to grid computing.

As seen in this table, the results obtained using the proposed methodology for kroAB100 and kroABC100 were superior to or at least as good as those found by Hansen for two different strategies of guiding 2-opt: with weighted sums ($p=1$) and with weighted Tchebycheff ($p=\infty$) functions. Also, a better result was obtained with the proposed methodology than Hansen's for the 5-objective problem (kroABCDE100) when 2-opt was guided with weighted Tchebycheff ($p=\infty$) function. In general, we observed that better or at least as good results were obtained by guiding 2-opt with a weighted sums function ($p=1$) than a weighted Tchebycheff function ($p=\infty$) for the tested problems. This observation was consistent with Hansen's suggestions.

In all the experiments presented in this research, except the ones shown in Table 4, a different strategy of guiding 2-opt is implemented. This strategy of 2-opt includes random selection of a weighted sums function or a weighted Tchebycheff function with equal chances of implementation. Fig. 3 illustrates the average relative excess of 30 replications of the proposed methodology with three different 2-opt strategies (random selection, with weighted Tchebycheff ($p=\infty$), and with weighted sums ($p=1$)) for kroAB100, kroABC100, and kroABCDE100 problems.

As seen in this figure, the random selection strategy of implementing 2-opt produced better or at least as good results than other strategies for the tested problems. ($p=\infty$) strategy produced the worst results for the tested problems. Average relative excess was equal with random selection and ($p=1$) strategies for

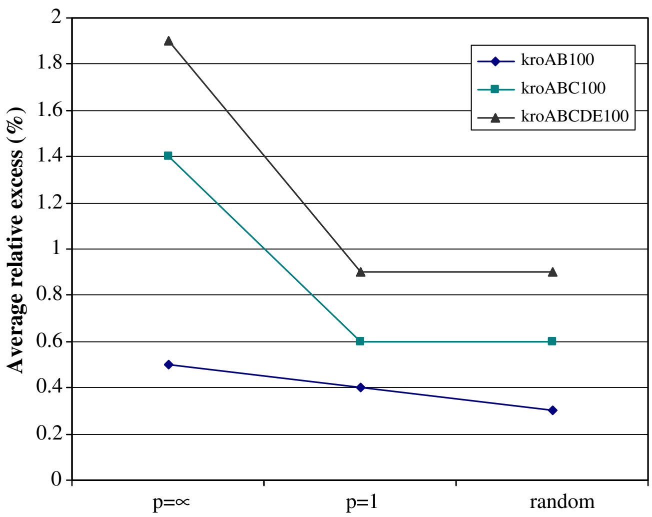

Fig. 3. Average relative excess of 30 replications of the proposed methodology in percent for kroAB100, kroABC100, and kroABCDE100 problems and three different strategies of implementing 2-opt.

kroAB100 and kroABC100. However, for kroABCDE100, guiding with random selection was better than both ($p = 1$) and ($p = \infty$) strategies. Based on these results, in our research, for all computational experiments we have used the random selection strategy.

6. Conclusions

In this paper, we have presented a memetic RKGA in order to find (weakly) Pareto optimal solutions to a mo-TSP. The key contribution in this research is the application of a novel memetic algorithm that includes a RKGA and local search to the mo-TSP. RKGA is preferred to other GA approaches since the random keys encoding eliminates the infeasibilities that can occur in other GA techniques during the application of some genetic operators, so a repair algorithm which might consume time and inhibit convergence is not required to move back to the search space. Another contribution is the way local search is implemented. Here, the metric to guide the 2-opt is selected randomly: a weighted sums or a weighted Tchebycheff metric is applied with equal chances of implementation, even though the general evaluation function of RKGA remains a weighted Tchebycheff function. We speculate that this implementation helps us to obtain better results of weighted Tchebycheff function, helps the search to stay on track in terms of the general evaluation function (weighted Tchebycheff metric), and increases the chance to find non-supported efficient solutions. More research needs to be done in this area where this approach is applied to other multi-criteria combinatorial optimization problems, preferably to problems with different shapes and irregularities in non-dominated sets to be able to observe the performance of this approach further.

Memetic RKGA performed well on average for the 2-, 3-, 4-, and 5- objective tested problems as indicated in the previous section. Based on these encouraging results, for future research, the presented algorithm can be applied to several variants of mo-TSPs. In general, additional instances can be created following Hansen's scheme; i.e., selecting subsets of an appropriate size from TSPLIB instances to compute the costs for various objectives. Also, interactive multi-objective decision making methods can be studied for mo-TSPs and its variants. In the presented research, user preferences are incorporated into decision making by means of "weighting coefficients." However, this approach might not be adequate for reflecting user preferences. Other ways of incorporating user preferences could be by classification of objective functions into several classes (e.g., objective functions that should be decreased, objective functions that are allowed to be increased to a specified upper bound, etc.) and implementation of bounds, indifference tradeoffs, etc.

References

Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing, 6*(2), 154–160.

Bowman, V. J. Jr., (1976). On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In H. Thiriez & S. Zionts (Eds.), *Multiple criteria decision making. Lecture notes in economics and mathematical systems* (Vol. 130, pp. 76–85). Berlin, Heidelberg: Springer-Verlag.

Cheng, R., & Gen, M. (1994). Crossover on intensive search and traveling salesman problem. *Computers & Industrial Engineering, 27*(1–4), 485–488.

Cheng, R., Gen, M., & Sasaki, M. (1995). Film-copy deliverer problem using genetic algorithms. *Computers & Industrial Engineering, 29*(1–4), 549–553.

Coello Coello, C. A. (2001). A short tutorial on evolutionary multiobjective optimization. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, & D. Corne (Eds.). *Lecture notes in computer science, evolutionary multi-criterion optimization, first international conference, EMO 2001 proceedings* (pp. 21–40). Springer.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation, 6*(2), 181–197.

Ehrgott, M. (2000). Approximation algorithms for combinatorial multicriteria optimization problems. *International Transactions in Operational Research, 7*, 5–31.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic algorithms for multi-objective optimization: formulation, discussion and generalization. In S. Forrest (Ed.), *Genetic algorithms: proceedings of 5th international conference* (pp. 416–423). CA: Morgan Kaufmann.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. MA: Addison-Wesley.

Goncalves, J. F., & Resende, M. G. C. (2004). An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering, 47*, 247–273.

Hansen, M. P. (2000). Use of substitute scalarizing functions to guide a local search based heuristic: The case of moTSP. *Journal of Heuristics, 6*, 419–431.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.

Jaszkiewicz, A. (2002). Discrete optimization, genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research, 137*, 50–71.

Knowles, J., & Corne, D. (2004). Memetic algorithms for multiobjective optimization: Issues, methods and prospects. In N. Krasnogor, J. E. Smith, & W. E. Hart (Eds.), *Recent advances in memetic algorithms* (pp. 313–352). Springer.

Kubota, N., Fukuda, T., & Shimojima, K. (1996). Virus-evolutionary genetic algorithm for a self-organizing manufacturing system. *Computers and Engineering, 30*(4), 1015–1026.

Kurz, M. E., & Askin, R. G. (2004). Scheduling flexible flow lines with sequence-dependent setup times. *European Journal of Operational Research, 159*(1), 66–82.

Merz, P., & Freisleben, B. (2001). Memetic algorithms for the traveling salesman problem. *Complex Systems, 13*, 297–345.

Michalewicz, Z. (2000). Repair algorithms. In T. Back, D. B. Fogel, & Z. Michalewicz (Eds.), *Evolutionary computation 1, basic algorithms and operators* (pp. 56–61). Bristol: Philadelphia Institute of Physics Publishing.

Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Kluwer International Series.

Norman, B. A., & Bean, J. C. (1999). A genetic algorithm methodology for complex scheduling problems. *Naval Research Logistics, 46*, 199–211.

Norman, B. A., & Bean, J. C. (2000). Scheduling operations on parallel machine tools. *IIE Transactions, 32*, 449–459.

Reinelt, G. (1995). TSPLIB. URL: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

Ryan, E., Azad, M. A., & Ryan, C. (2004). On the performance of genetic operators and the random key representation. In *Genetic programming, proceedings of 7th european conference* (pp. 162–173). Coimbra, Portugal: EuroGP.

Samanlioglu, F., Kurz, M.E., & Ferrell Jr., W.G. (2006). A genetic algorithm with random-keys representation for a symmetric multi-objective traveling salesman problem. In: Proceedings of the Institute of Industrial Engineers Annual Conference. Orlando: Florida.

Samanlioglu, F., Kurz, M. E., Ferrell Jr., W. G., & Tangudu, S. (2007). A hybrid random-key genetic algorithm for a symmetric traveling salesman problem. *International Journal of Operations Research, 2*(1), 47–63.

Schaffer, J.D. (1984) Multiobjective optimization with vector evaluated genetic algorithms. Ph.D. Thesis. Nashville, TN: Vanderbilt University.

Snyder, L. V., & Daskin, M. S. (2006). A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research, 174*(1), 38–53.

Wierzbicki, A. P. (1986). On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spektrum, 8*, 73–87.

Yu, P. L. (1973). A class of solutions for group decision problems. *Management Science, 19*(8), 936–946.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the performance of the strength pareto evolutionary algorithm. In: Technical Report 103, Computer Engineering and Communications Networks Lab (TIK). Zurich: Swiss Federal Institute of Technology (ETH).