International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

An interactive memetic algorithm for production and manufacturing problems modelled as a multi-objective travelling salesman problem

F. Samanlioglu , W.G. Ferrell & M.E. Kurz

To cite this article: F. Samanlioglu , W.G. Ferrell & M.E. Kurz (2012) An interactive memetic algorithm for production and manufacturing problems modelled as a multi-objective travelling salesman problem, International Journal of Production Research, 50:20, 5671-5682, DOI: 10.1080/00207543.2011.593578

To link to this article: https://doi.org/10.1080/00207543.2011.593578

International Journal of Production Research
Vol. 50, No. 20, 15 October 2012, 5671–5682

Taylor & Francis
Taylor & Francis Group

An interactive memetic algorithm for production and manufacturing problems modelled as a multi-objective travelling salesman problem

F. Samanlioglu[a]*, W.G. Ferrell[b] and M.E. Kurz[b]

[a]*Department of Industrial Engineering, Kadir Has University, Istanbul, Turkey;*
[b]*Department of Industrial Engineering, Clemson University, Clemson, SC, United States of America*

(*Received 3 December 2010; final version received 31 May 2011*)

In this paper, a preference-based, interactive memetic random-key genetic algorithm (PIMRKGA) is developed and used to find (weakly) Pareto optimal solutions to manufacturing and production problems that can be modelled as a symmetric multi-objective travelling salesman problem. Since there are a large number of solutions to these kinds of problems, to reduce the computational effort and to provide more desirable and meaningful solutions to the decision maker, this research focuses on using interactive input from the user to explore the most desirable parts of the efficient frontier instead of trying to reproduce the entire frontier. Here, users define their preferences by selecting among five classes of objective functions and by specifying weighting coefficients, bounds, and optional upper bounds on indifference tradeoffs. This structure is married with the memetic algorithm – a random-key genetic algorithm hybridised by local search. The resulting methodology is an iterative process that continues until the decision maker is satisfied with the solution. The paper concludes with case studies utilising different scenarios to illustrate possible manufacturing and production related implementations of the methodology.

Keywords: multi-criterion decision making; genetic algorithms; Pareto optimisation; metaheuristics; interactive computing; travelling salesman problems

1. Introduction

In modelling production and manufacturing problems, the travelling salesman problem (TSP) or variants of TSP are widely used. Onwubolu and Clerc (2004) modelled the path of an automated CNC drilling machine as a TSP, where the drill-bit is the salesman visiting the points to be drilled, and the drilling locations are the cities. Their objective was to minimise the total distance moved by the machine in order to cut the total drilling time and production costs. Emmons and Mathur (1995) studied a no-wait flowshop, which manufactures large numbers of copies of a few basic job types, and modelled the related problem as a TSP. Leon and Peters (1996) compared different set-up strategies for printed circuit card assembly, and tried to minimise the makespan on a single machine using methods based on the assignment problem and TSP. Bagchi *et al*. (2006) presented a review of literature in which flowshop scheduling problems are modelled as a TSP. In their review, they mentioned no-wait flowshops, blocking flowshops, group scheduling of parts in a flowshop using a generalised extension of the TSP, lot streaming and scheduling problems, and scheduling of parts and robot movements in automated production cells. Lee and Kuo (2008) studied the multi-order picking problem in an automated carousel conveyor, where they determined the picking sequence for a given number of orders and the picking sequence for the items within each order while minimising the total picking cost or time. They modelled the problem as an open loop version of the multi-travelling salesman problem since the picker does not need to return to the starting point and goes to process the next order without returning to the first order. Lee and Kwon (2006) worked on generating an optimal 2D cutting path for a stock plate nested with a set of regular or irregular parts while minimising the total non-productive travelling distance of a cutter. They formulated the problem as a generalised case of TSP where the locations of visiting nodes are not known in advance. Cattrysse *et al*. (2006) considered the single press brake problem, where they determined how to sequence a selected subset of production layouts and how to allocate all the jobs to the selected stations while minimising the makespan. They used two mathematical models: the Travelling Purchaser Problem and the Generalised Travelling Salesperson Problem to capture the structure of the problem. Dolgui and Pashkevich (2006) worked on the cluster-level welding operations planning for a robotic cell with a positioning table. They presented a generalised TSP model to determine

*Corresponding author. Email: fsamanlioglu@khas.edu.tr

ISSN 0020–7543 print/ISSN 1366–588X online

http://dx.doi.org/10.1080/00207543.2011.593578
http://www.tandfonline.com

the optimal welding cluster sequence and the corresponding optimal motions of the positioned, while minimising the overall manufacturing time by finding a reasonable trade-off between the positioner motion times and the cluster processing times. Kim *et al*. (2003) modelled the order picking sequence problem of a gantry robot in an automated warehouse as a special type of TSP. They determined the optimal sequence of order picking with given vertical paths, while minimising the travelling time of the robot to fulfill the orders. Ganesh and Narendran (2008) worked on the reverse logistics problem of simultaneous distribution of commodities and the collection of reusable empty packages, when there is a single depot and a single vehicle with limited capacity. They modelled the problem as the travelling salesman problem with simultaneous delivery and pick-up, where they minimised the total distance travelled by the vehicle. Yadollahpour *et al*. (2009) studied the hot strip mill scheduling problem with consideration of the selection and sequencing of slabs to be hot rolled. They presented a mathematical model based on the prize collecting vehicle routing problem, where each customer represents a slab and its prize measures the utilisation of processing that slab. In their model, they combined two objectives: transition costs, which include quality, temperature, thickness, and width costs, and utilities, which include length, temperature, earliness, and tardiness utilities into a single objective using the weighted sums method.

As seen in these applications, most of the production and manufacturing problems that are modelled as TSP or its variants in fact include multiple objectives. Some of these objectives could be minimising total cost, total time, total distance, total risk, total waste, and maximising quality, and customer satisfaction. The multi-objective travelling salesman problem (mo-TSP) combines the familiar general requirement of the TSP of finding a tour that begins in a specific city, visits each of the remaining cities exactly once, and returns to the initial city, and the added dimension of classic multi-objective optimisation where there are several objectives such as total distance travelled, total time, total risk, total cost, etc. that must simultaneously be optimised but that potentially conflict. This research is restricted to the symmetric problem, meaning the measures are known and are symmetric; for example, the time to travel from City A to City B is the same as from City B to City A. There are a number of strategies for finding a ''solution'' to this problem but the underlying fact is that there is not a single solution; rather, there are a number of solutions that taken together form an efficient frontier. This research focuses on using interactive input from the user to explore the most desirable parts of the efficient frontier instead of trying to reproduce a large number of solutions along the entire frontier. This will reduce the computational effort and provide more desirable and meaningful solutions to the decision maker (DM) while solving production and manufacturing problems that can be modelled as mo-TSP. To be precise, this research centres on finding (weakly) efficient solutions to mo-TSP that are preferred by the DM by using a preference-based interactive memetic random-key genetic algorithm (PIMRKGA). In PIMRKGA, the DM may set goals for changes in objective function values with bounds on those changes (classifying), may mandate relative weightings of the objective functions (weighting), or may set bounds on the increase of one objective function value relative to the decrease of another objective function value (tradeoff-bounding). None of the interactive multi-objective optimisation methods or interactive, preference-based multi-objective evolutionary algorithms in the literature, incorporate all of these features.

In multi-criterion decision making, one way of categorising existing methods is based on when the DM is engaged in the solution process. In *a priori* methods such as the Value Function Method, Goal Programming and Lexicographic Ordering, a model is constructed and a solution is generated based on preferences provided by the DM. The main disadvantage of these methods is that the DM preferences might be unrealistic. *A posteriori* methods such as the ε-Constraint Method, the Weighting Method, the Method of Weighted Metrics and the Achievement Scalarizing Function Approach generate or approximate the entire Pareto front without incorporating the DM preferences into the process. The DM then selects the most preferred solution among the set of solutions. The main disadvantage of this strategy, regardless of implementation, is that generating the entire Pareto front might be impossible or, if it can be done, will be extraordinarily expensive computationally. Interactive methods seek to overcome these drawbacks by combining parts of *a priori* and *a posteriori* methods. Generally, interactive algorithms start with an initial feasible solution and new solutions are generated iteratively in coordination with consistent preference information provided by the DM. This allows a final solution to be obtained with only a portion of the (weakly) Pareto optimal solutions being produced and evaluated.

For many years, the research related to multi-objective evolutionary algorithms has focused on approximating the shape and extent of the Pareto front but then not really focusing on a specific solution based on preference information. The first preference-based interactive procedure (I-EMO) was suggested by Deb and Chaudhuri (2003). This approach starts by presenting the extent and the shape of the complete or partial Pareto front, knee or robust solutions to the DM and then generates an improvement using any of several techniques. This allows the DM to focus on the desired portion(s) of the Pareto front or individual solution(s). One limitation is that the I-EMO

software can handle, at most, three objectives. There are other interactive evolutionary multi-objective algorithms in the literature that are typified by IIMOM (Huang *et al*. 2005), and i-MOM (Caballero *et al*. 2006). These methods, however, do not offer the DM the level of control in search strategy (i.e., classifying, value bounding, weighting, and tradeoff-bounding) in higher-order Pareto-optimal solution sets (i.e., solution sets for five or more objectives).

Some researches combine interactive, preference information with evolutionary algorithms to solve real life problems. Quan *et al*. (2007) introduced a new method based on incompletely specified multiple attribute utility theory (ISMAUT) so that the DM can target a subset of Pareto optimal solutions based on the DM's preferences, specifically by ranking a small set of initial solutions. They applied their method to schedule preventive maintenance tasks to ensure critical equipment availability in heavy industry maintenance facilities while simultaneously minimising the conflicting objectives, makespan and workforce. Ecemis *et al*. (2008) implemented a GA and interactive evolutionary computation-based software entitled Mobius and developed a drug candidate design environment. Brintrup *et al*. (2007) proposed an interactive genetic algorithm, handling qualitative and quantitative objectives simultaneously in design optimisation that could address, for example, the manufacturing plant layout design problem. In a similar vein, Brintrup *et al*. (2008) implemented an interactive genetic algorithm to design an ergonomic chair, fusing qualitative and quantitative criteria. While all of these methods have interesting features, none offer the level of flexibility and control to the mo-TSP that is proposed in this research.

Successfully designing an interactive procedure requires careful consideration of a number of criteria including: the amount and form of information taken from and provided to the DM; the robustness, convergence and ease of use of the algorithm; and the computational efficiency of the algorithm. Existing methods differ from each other and the methodology proposed in this research, particularly in terms of these criteria and assumptions. The closest interactive system to our research is an interactive method that was specifically designed for nondifferentiable functions called Nondifferentiable Interactive Multiobjective Bundle-based optimization System (NIMBUS) (Miettinen 1999, Miettinen and Makela 2000, 2002, 2006, Miettinen *et al*. 2003, WWW-NIMBUS). NIMBUS is capable of solving nonlinear, nondifferentiable, and nonconvex problems. In NIMBUS, DM incorporates preferences by classifying objective functions into five classes based on demanded changes. NIMBUS has more objective function classes than STEM (Benayoun *et al*. 1971), STOM (Nakayama and Furukawa 1985), Bi-reference procedure (Michalowski and Szapiro 1992), and the Reference Direction Method (Narula *et al*. 1994) so the DM has more flexibility in specifying the desired changes in objective functions. NIMBUS uses local and global optimisers. The local solver is the proximal bundle method and the global optimisers are a GA with the constraint handling technique suggested by Deb (2000), and a GA with the method of adaptive penalties for handling constraints (WWW-NIMBUS). The GA of NIMBUS is real-coded and includes elite reproduction, mutation, and heuristic crossover with tournament selection operators (Miettinen *et al*. 2003, WWW-NIMBUS).

The proposed PIMRKGA is substantially different, unique from all the methodologies in the literature because of the flexibility and control provided to the DM through classifying, value bounding, weighting, and tradeoff-bounding as well as using an RKGA specifically designed for mo-TSP.

Attention is now turned to describing the details of the global optimiser, memetic RKGA, and its application to mo-TSP.

2. Memetic RKGA applied to TSP and mo-TSP

TSP and mo-TSP belong to the class of NPC (Nondeterministic Polynomial-time Complete) problems, so heuristic methods which provide approximate solutions are justified and widely used. As such, several meta-heuristic methods have been developed for mo-TSP including Hansen (2000), Jaszkiewicz (2002) and Samanlioglu *et al*. (2008). It is noted that a TSP with a larger number of cities often benefits from combining approaches to enhance the ability of the algorithm to find good solutions; the caveat is that it is critical for the search space to remain diverse. In particular, methods that combine domain-specific local search and evolutionary algorithms have received special attention and are called memetic or hybrid algorithms (Knowles and Corne 2005). In this research, the 2-opt local search is combined with RKGA (Bean 1994) so that the RKGA is allowed to work in a search space of local minima but with diversity preserved via different genetic operators to avoid trapping. The memetic algorithm utilising the random-keys encoding is used as the global optimiser during the interactive search procedure of PIMRKGA rather than other GA approaches because it progresses using only feasible tours, thereby eliminating the need for a repair algorithm.

In general, the random keys representation for the TSP encodes a solution with random integers between a lower and upper value, which are then used as ''sort keys'' to decode the solution. Each node is assigned a key in the chromosome. To decode a chromosome, the nodes are sorted in the ascending order of their corresponding keys to indicate the travel order of a salesman. For example, for a five-node problem, the random key encoding of a chromosome $C = \{2, 15, 13, 98, 24\}$ decodes as the tour $1 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 1$. The random key encoding eliminates the possibility of infeasible tours that might occur during the implementation of some genetic operators such as crossover in the algorithm. This avoids the need for repair algorithms to reconstruct feasible solutions that are required when literal encoding such as binary vector is used. Genetic operators are executed on the random key encoding, not on the decoded tour, so that all the resulting chromosomes decode to produce feasible solutions. Details of this procedure and an illustration of application of crossover to literal encoding and random key encoding can be found in Samanlioglu *et al*. (2007).

While the existing interactive methods have never been integrated with RKGA and applied to a mo-TSP which is the focus of this research, there is some important research that must be acknowledged in this general area. Since its introduction by Bean (1994), RKGA has been applied to sequencing and scheduling problems (Kurz and Askin 2004), and a single objective generalised TSP (Snyder and Daskin 2006). Recently, memetic RKGA has been applied to a symmetric TSP (Samanlioglu *et al*. 2007) and a symmetric mo-TSP (Samanlioglu *et al*. 2006, 2008). We now present our memetic RKGA, and the methodology by which this can be applied to the interactive procedure.

The genetic operators used in the research are elitist reproduction, immigration, parameterised uniform crossover with tournament selection, and swap mutation, similar to those in the Bean (1994), Ryan *et al*. (2004), and Samanlioglu *et al*. (2006, 2007, 2008) research. Since memetic RKGA is an iterative procedure, a stopping criterion is required and, in our approach, the algorithm is stopped after a specified number of generations (MAX_GEN) is reached. The basic steps of the memetic RKGA are:

(1) Randomly generate the initial population of POP_SIZE chromosomes representing POP_SIZE feasible tours and apply 2-opt to all chromosomes.
(2) Perform elitist reproduction whereby the best ELITE_NUM chromosomes of the current generation are copied to the next generation.
(3) Perform swap mutation with 2-opt applied after each chromosome creation to each chromosome in the next generation, adding the resulting chromosomes to the next generation. Here, s nodes are selected at random along the chromosome, all the possible permutations of the keys of these s nodes are created (implementing 2-opt after each permutation), and the best found chromosome is kept. We repeat this procedure r times for the same chromosome. The next generation now has 2*ELITE_NUM chromosomes.
(4) Perform parameterised uniform crossover with tournament selection and 2-opt to create CROSS new chromosomes. Basically, in parameterised uniform crossover with tournament selection and 2-opt, two chromosomes are randomly selected from the population. Then, for each gene that corresponds to a city in the TSP, a random number between 0 and 1 is generated. If the number is less than or equal to the crossover probability PAR1_PERCENT, 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 number is greater than PAR1_PERCENT, the genes are swapped.Then 2-opt is applied to both new chromosomes, and as a result of the tournament selection, both chromosomes are evaluated and the better one is included in the next generation. The next generation now has 2*ELITE_NUM + CROSS chromosomes.
(5) Use immigration (randomly create new chromosomes) to generate the remaining members of the next generation, applying 2-opt to these chromosomes. The next generation is now complete with POP_SIZE chromosomes.
(6) If the stopping criterion is satisfied, go to step 7, otherwise, go to step 2.
(7) Present the best chromosome and terminate the algorithm.

In a single objective TSP, the evaluation function consists of total cost, distance, risk, or time of the travelling salesman tour, so the aim is to find the ''best'' tour, which has the lowest evaluation function. Memetic RKGA must be adapted for use in multi-objective optimisation because 2-opt, genetic operators and the stopping criterion are aligned for one objective when the concept of ''best'' solution is unambiguous. This is not the case with multi-objective problems in general and mo-TSP in particular; therefore, a modification is required to adjust memetic RKGA to mo-TSP and an interactive environment. In fact, the only adjustment is the modification of the evaluation function in memetic RKGA based on the phases in the interactive search procedure as explained in the following section.

3. Interactive search procedure of PIMRKGA

In PIMRKGA, the goal is to find a DM-preferred solution when several objective function characteristics are determined by the DM and the DM is allowed to change them as the search progresses. Specifically, the DM is allowed to set goals for changes in objective function values with bounds (classifying), mandate relative weightings of the objective functions (weighting), or set bounds on the increase of one objective function value relative to the decrease of another objective function value (tradeoff-bounding). The overall structure of the PIMRKGA involves: (1) Initialisation phase – an initial weakly Pareto optimal solution is obtained, (2) Classification phase – the DM is queried for preferences and the preferences are incorporated into the memetic RKGA and preferred (weakly) Pareto optimal solutions are obtained, (3) Alternative solutions phase – alternative (weakly) Pareto optimal solutions between the solutions already obtained are computed, and (4) Pareto optimality test phase – final solution is tested for Pareto optimality. It should be noted that the interactive nature of the PIMRKGA methodology allows the DM to continue searching iteratively or stop if he or she is satisfied each time a solution is presented. We now present a few essential definitions related to multi-objective optimisation before describing the phases of PIMRKGA.

A multi-objective program (MOP), $\min f(x) = \{f_1(x), f_2(x), \ldots, f_k(x)\}$ s.t. $x \in X$ is assumed to have k ($k \geq 2$) competing objective functions ($f_i : \Re^n \to \Re$) that are to be minimised 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 $f_i(x) \leq f_i(x^*)$ 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.)

Definition: $z_i'' = (z_1'', z_2'', \ldots, z_k'')$ with $z_i'' = \min f_i(x)$ s.t. $x \in X$ is called the ideal point.

Theorem: *We assume that $w_i > 0$ for all $i = 1, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$, where the w_i are weighting coefficients provided by the DM. Weakly efficient solutions can be generated with any reference point $\ddot{z} \in R^k$ by solving the achievement scalarising problem* (Wierzbicki 1982, Miettinen 1999)

$$\min \max_i \left[w_i \left(f_i(x) - \ddot{z} \right) \right] \quad \text{s.t. } x \in X. \tag{1}$$

Definition: Two feasible solutions are said to be situated on the same indifference curve if the DM finds them equally desirable. For any two solutions on the same indifference curve, there is a tradeoff involving the amount of increment in the corresponding value of an objective function f_i that the DM is willing to tolerate in exchange for one unit decrement in the value of objective function f_j, while the values of all the other objectives remain unaltered and $i \neq j$. This tradeoff is called "*indifference tradeoff*" or "*marginal rate of substitution*". Indifference tradeoffs can be approximated by $m_{ij} \approx \frac{\Delta f_i}{\Delta f_j} \quad i, j = 1, \ldots, k, \ i \neq j$ where Δf_i is the amount of increment in objective function i, and Δf_j is the amount of decrement in objective function j. The approximation becomes arbitrarily exact when $\Delta \to 0$ (Miettinen 1999).

3.1 *Initialisation phase of PIMRKGA*

At the initialisation phase of PIMRKGA, the achievement scalarising problem (1) is used as the evaluation function to locate an initial weakly Pareto optimal solution. Here, the ideal point is used as the reference point ($\ddot{z} = z''$) in problem (1).

3.2 *Classification phase*

The proposed methodology allows the DM to classify the objective functions into five classes. Without loss of generality, simultaneous minimisation of all objectives is assumed. The objective function values of the h^{th} iteration are defined as $z_i^h = f_i(x^h)$. The classes for the h^{th} iteration are defined in terms of goals that the DM has for each objective function, namely:

(1) objective functions that should be decreased ($i \in I^{<,h}$),

(2) objective functions that must be decreased to a value less than or equal to a certain bound $\bar{z}_i^h$ ($i \in I^{\leq,h}$), such that $\bar{z}_i^h < f_i(x^h)$, $\bar{z}_i^h > z_i''$, $f_i(x) \leq \bar{z}_i^h \quad \forall i \in I^{\leq,h}$,
(3) objective functions that are satisfactory at the moment ($i \in I^{=,h}$),
(4) objective functions that are allowed to increase to a certain upper bound β_i^h ($i \in I^{>,h}$), such that $f_i(x^h) < \beta_i^h$, $f_i(x) \leq \beta_i^h \ \forall i \in I^{>,h}$,
(5) objective functions that are allowed to change freely ($i \in I^{\diamond,h}$). Note that, in practice, this class means that not all objective functions need to be classified by the DM. If the DM does not want to classify an objective function, he or she can assume that the objective function belongs to this class.

Clearly, objectives can belong to only one of the five classes and, furthermore, we require $I^{<,h} \cup I^{\leq,h} \neq \varnothing$ and $I^{=,h} \cup I^{>,h} \cup I^{\diamond,h} \neq \varnothing$.

We can introduce the following sets of constraints (Equations (2)–(5)) to enforce goals 1–4 above as follows:

$$f_i(x) < f_i\left(x^h\right) \quad \forall i \in I^{<,h} \tag{2}$$

$$f_i(x) \leq f_i\left(x^h\right) \quad \forall i \in I^{=,h} \tag{3}$$

$$f_i(x) \leq \bar{z}_i^h \quad \forall i \in I^{\leq,h} \tag{4}$$

$$f_i(x) \leq \beta_i^h \quad \forall i \in I^{>,h} \tag{5}$$

3.2.1 *Weighting*

For objective functions f_i that belong to classes $I^{<,h}$ and $I^{\leq,h}$, the DM may specify optional weighting coefficients (for the h^{th} iteration) to adjust the order of importance, $w_i^h > 0$ and $\sum_i w_i^h = 1$. If weighting coefficients are not assigned by the DM, they are assumed to be equal. These are included by adapting the objective function to be $\min \max_{i \in I^{<,h} \cup I^{\leq,h}} \left[w_i^h\left(f_i(x) - z_i''\right)\right]$. Note that this is an achievement scalarising function.

3.2.2 *Tradeoff-bounding*

Here, the DM may define optional upper bounds T_{ij}^h on the approximate "indifference tradeoffs" between objective functions f_i ($i \in I^{>,h}$) and f_j ($j \in I^{<,h}$). For the h^{th} iteration, this is computed as $\frac{f_i(x) - f_i(x^h)}{f_j(x^h) - f_j(x)} \leq T_{ij}^h$ which relates the $f_i(x^h)$ ($i \in I^{>,h}$) and $f_j(x^h)$ ($j \in I^{<,h}$). So, we can introduce the following set of constraints to enforce DM tradeoff-bounding as follows:

$$\frac{f_i(x) - f_i(x^h)}{f_j(x^h) - f_j(x)} \leq T_{ij}^h \quad i \in I^{>,h}, j \in I^{<,h} \tag{6}$$

3.2.3 *DM-preferred mo-TSP formulation*

Objective classification, weighting and tradeoff-bounding resulted in additional constraints and a transformed objective function, as shown in problem (7) below.

$$\begin{aligned}
\min \ & \max_{i \in I^{<,h} \cup I^{\leq,h}} \left[w_i^h\left(f_i(x) - z_i''\right)\right] \\
\text{s.t. } & f_i(x) < f_i\left(x^h\right) \quad \forall i \in I^{<,h} \\
& f_i(x) \leq f_i\left(x^h\right) \quad \forall i \in I^{=,h} \\
& f_i(x) \leq \bar{z}_i^h \quad \forall i \in I^{\leq,h} \\
& f_i(x) \leq \beta_i^h \quad \forall i \in I^{>,h} \\
& \frac{f_i(x) - f_i\left(x^h\right)}{f_j(x^h) - f_j(x)} \leq T_{ij}^h \quad i \in I^{>,h}, j \in I^{<,h} \\
& x \in X
\end{aligned} \tag{7}$$

Problem (7) consists of an achievement scalarising function (ideal point is used as the reference point) that includes objective functions f_i, $i \in I^{<,h}, I^{\leq,h}$ and several constraints (2)–(6), so the result of this problem is a weakly Pareto optimal solution. We can linearise the objective function in problem (7), creating problem (8) below.

$$
\begin{aligned}
&\min \alpha \\
&\text{s.t. } \alpha \geq w_i^h\left(f_i(x) - z_i''\right) \quad \forall i \in \left(I^{<,h} \cup I^{\leq,h}\right) \\
&\quad f_i(x) < f_i\left(x^h\right) \quad \forall i \in I^{<,h} \\
&\quad f_i(x) \leq f_i\left(x^h\right) \quad \forall i \in I^{=,h} \\
&\quad f_i(x) \leq \bar{z}_i^h \quad \forall i \in I^{\leq,h} \\
&\quad f_i(x) \leq \beta_i^h \quad \forall i \in I^{>,h} \\
&\quad \frac{f_i(x) - f_i\left(x^h\right)}{f_j(x^h) - f_j(x)} \leq T_{ij}^h \quad i \in I^{>,h}, j \in I^{<,h} \\
&\quad x \in X
\end{aligned}
\tag{8}
$$

This problem can be solved using a standard solver such as ILOG CPLEX and exact solutions can be found easily for mo-TSP containing a small number of nodes. Recall, that both TSP and mo-TSP are NPC so heuristic methodologies like the one presented here are justified for larger problems.

3.2.4 *A preference-based evaluation function*

Once the DM has conveyed his or her preferences during the classification phase memetic RKGA is modified by taking the original problem's (7) objective function and adding constraints (2)–(6). Now, there are several ways of handling constraints in GAs (Coello Coello 2002); this research utilises penalty functions to incorporate constraints (2)–(6). The approach used here assigns penalty coefficients P_1, P_2, P_3 and P_4 to the deviations between the current objective function value and the goals set by the DM in the DM classification phase. The implementation yields the revised, preference-based evaluation function in problem (9). P_1 is used to penalise solutions that do not reduce objective functions in $I^{<,h}$ and $I^{=,h}$; P_2 is used to penalise solutions that do not reduce objective functions in $I^{\leq,h}$ below the desired bound; P_3 is used to penalise solutions that allow objective functions in $I^{>,h}$ to increase beyond the desired upper bound; and P_4 is used to penalise solutions that violate the tradeoff-boundings. These coefficients are assumed constant for all iterations. The revised, preference-based evaluation function (9) therefore consists of the sum of the maximum weighted deviation from the ideal point for the objective functions in $I^{<,h}$ and $I^{\leq,h}$ and the penalty terms for violating constraints (2)–(6):

$$
\begin{aligned}
\min \quad & \left[\begin{array}{l}
\left[\max_{i \in I^{<,h} \cup I^{\leq,h}}\left[w_i^h\left(f_i(x) - z_i''\right)\right]\right] + P_1\left[\max_{\forall i \in I^{<,h} \cup I^{=,h}}\left[0, \left(f_i(x) - f_i\left(x^h\right)\right)\right]\right] \\
\quad + P_2\left[\max_{\forall i \in I^{\leq,h}}\left[0, \left(f_i(x) - \bar{z}_i^h\right)\right]\right] + P_3\left[\max_{\forall i \in I^{>,h}}\left[0, \left(f_i(x) - \beta_i^h\right)\right]\right] \\
\quad + P_4\left[\max_{\forall i \in I^{>,h},\ \forall j \in I^{<,h}}\left[0, \left(\left(f_i(x) + T_{ij}^h f_j(x)\right) - \left(f_i\left(x^h\right) + T_{ij}^h f_j\left(x^h\right)\right)\right)\right]\right]
\end{array}\right] \\
\text{s.t.} \quad & x \in X.
\end{aligned}
\tag{9}
$$

Preliminary tests were performed to determine the values of the penalty functions that yielded high quality solutions with a reasonable amount of computing time. These values were used in the experimentation that we present in a later section.

3.3 *Alternative solutions phase*

The achievement scalarising problem (1) is used during the alternative solutions phase to find weakly Pareto optimal alternatives located between solutions already obtained. To find P alternatives between solutions z^h and $\hat{z}^h$, problem (1) is implemented as the evaluation function of PIMRKGA with reference point $\ddot{z} = f_i(x^h) + t_j d^h$ where $t_j = \frac{j}{P+1}$ and $d^h = \hat{z}^h - z^h$ for each alternative j ($j = 1, 2, \ldots, P$).

3.4 *Pareto optimality test phase*

A solution, x^h, can be tested for Pareto optimality by solving problem (10), where $x^h \in X$ is given and $x \in X,\ \varepsilon \in R_+^k$ are decision variables.

$$\begin{aligned} \max\ & \sum_{i=1}^{k} \varepsilon_i \\ \text{s.t. } & f_i(x) + \varepsilon_i = f_i(x^h) \quad \forall i = 1, \ldots, k \\ & \varepsilon_i \geq 0 \quad \forall i = 1, \ldots, k \\ & x \in X,\ \varepsilon \in R_+^k \end{aligned} \tag{10}$$

Let the solution of problem (10) be $(\tilde{x}, \tilde{\varepsilon})$. The vector x^h is Pareto optimal if and only if problem (10) has an optimal objective function value of zero ($\tilde{\varepsilon} = 0$). In that case, $\tilde{x} = x^h$. If problem (10) has a finite nonzero optimal objective function value ($\tilde{\varepsilon} > 0$) obtained at point $\breve{x}$ ($\tilde{x} = \breve{x}$), then $\breve{x}$ is Pareto optimal (Miettinen 1999).

The Pareto optimality test phase utilises the memetic RKGA with an adapted evaluation function and sorting method to solve problem (10). First, the chromosome corresponding to the solution that must be tested for Pareto optimality is inserted into the initial population. Then, the objective function in problem (10) is maximised, while testing each new solution if $\varepsilon_i \geq 0 \quad \forall i = 1, \ldots, k$ in order to find a preferred Pareto optimal solution. The population is sorted in such a way that when two chromosomes are compared: (1) the one satisfying the constraints ($\varepsilon_i \geq 0\ \forall i = 1, \ldots, k$) is better, (2) if both satisfy the constraints ($\varepsilon_i \geq 0\ \forall i = 1, \ldots, k$), the one with the larger evaluation function is better since we try to maximise the objective function in problem (10), and (3) if neither satisfies the constraints ($\varepsilon_i \geq 0\ \forall i = 1, \ldots, k$), the one with the larger evaluation function is better since we assume that would be the least violating chromosome.

Once memetic RKGA terminates, the value of the objective function for the best chromosome is checked. If this value is zero, the tested chromosome is found to be Pareto optimal. If this value is positive, then the corresponding new chromosome is Pareto optimal, not the one tested for Pareto optimality. This phase of PIMRKGA terminates when the Pareto optimal solution is presented to the DM.

4. Case studies

PIMRKGA is implemented in C++ and applied to two case studies, a 50-node five-objective mo-TSP and a 100-node five-objective mo-TSP based on TSPLIB (Reinelt 1995). These case studies utilising different scenarios are created to illustrate possible manufacturing and production related implementations of the methodology. The parameters used for the algorithm are presented in Table 1 and were selected based on previous experience and preliminary testing (Samanlioglu *et al.* 2008). Now, as in all heuristics akin to genetic algorithms, choice of parameter can sometimes dramatically influence performance. Since the focus of this research is to present a new methodology by addressing mo-TSP problems and not improved computing efficiency, the choice of parameter is not considered critical; hence, selection was based on finding solutions in a reasonable time. With this in mind, it might be interesting to note that keeping values for the penalty coefficients reasonably small was important in finding Pareto optimal solutions in a modest amount of computing time. It seemed important to keep the value small enough that violating a constraint did not create a large impact on the objective function but large enough that the infeasible solutions were sufficiently poor that they were discounted in the next generation.

Table 1. Parameters used in PIMRKGA.

	Parameter	Value
Genetic algorithm parameters used in PIMRKGA	ELITE_NUM	0.20*POP_SIZE
	CROSS	0.59*POP_SIZE
	PAR1_PERCENT	0.70
	MAX_GEN	6
	POP_SIZE	500
	S,*R*	4,10
Penalisation weights	P_1, P_2, P_3, P_4	20

4.1 *Case study with exact solutions*

A 50-node, five-objective mo-TSP is created by modifying Euclidean TSP instances (kroA100, kroB100, kroC100, kroD100, and kroE100) taken from TSPLIB, and using the first 50 nodes of kroA100, kroB100, ..., kroE100 instances to calculate the values of five different objectives. The results obtained from PIMRKGA are then compared with exact solutions obtained from ILOG OPL Development Studio 4.2 (ILOG CPLEX 10.0). To get exact solutions from CPLEX, ideal point calculations ($z_i'' = (z_1'', z_2'', \ldots, z_k'')$ with $z_i'' = \min f_i(x)$ s.t. $x \in X$), and problems (1), (7), and (10) are used during the interactive search procedure. The problem size is limited to 50 nodes in order to allow comparison of the results of the mathematical formulations shown in problems (1), (7), and (10) with PIMRKGA at each phase of the interactive search procedure.

An interactive scenario is created to demonstrate the implementation of PIMRKGA. The results obtained from PIMRKGA matched completely with exact solutions obtained from ILOG OPL Development Studio 4.2 (ILOG CPLEX 10.0) so only one solution is presented in Table 2.

As shown in Table 2, the procedure starts with finding the ideal point (z_i'') and an initial weakly Pareto optimal solution (z^0). Based on this initial solution, the DM classifies five objective functions, specifies bounds, weights and tradeoff-bounding. According to this classification, a new weakly Pareto optimal solution is found ($\hat{z}^0$). The DM is not satisfied and decides to make another classification at the point $z^1 = \hat{z}^0$. According to this classification a new weakly Pareto optimal solution is found ($\hat{z}^1$). The DM wants to see two weakly Pareto optimal alternatives between z^1 and $\hat{z}^1$, so alternatives $z^{1(A1)}$ and $z^{1(A2)}$ are presented. The DM prefers the second alternative and wants to stop, so $z^{1(A2)}$ is tested for Pareto optimality and found to be Pareto optimal. Note, that α values presented here are as a result of linearisation of achievement scalarising functions (1) and (7) as shown in problem (8).

4.2 *A more realistic case study*

A more realistic, five-objective, 100-node mo-TSP (kroABCDE100) is created by combining five Euclidean TSPLIB instances (kroA100, kroB100, kroC100, kroD100, and kroE100). Optimal solutions for each problem (kroA100, kroB100, ..., kroE100) are listed in TSPLIB, so the ideal point was readily available for use in the created mo-TSP. Also, the best result found so far for the initialisation phase of PIMRKGA was available from our previous research (Samanlioglu *et al*. 2008).

Since we were not able to obtain exact solutions of the kroABCDE100 experiments from ILOG OPL Development Studio 4.2 (ILOG CPLEX 10.0) in a few days time due to computer limitations (the software stopped without finding solutions), in Table 3 only PIMRKGA solutions are presented. Comparison with other interactive multi-objective optimisation methods or interactive, preference-based multi-objective evolutionary algorithms was not performed because they do not address the type of problem presented here.

Table 2. Interactive results for a 50-node five-objective mo-TSP.

Ideal point: $z_i'' = (16{,}461,\ 16{,}520,\ 15{,}772,\ 16{,}319,\ 15{,}911)$
Initialisation: $z^0 = f(x^0) = (48{,}211,\ 48{,}214,\ 47{,}434,\ 47{,}782,\ 47{,}368) \quad \alpha = 6350$

Classification at $z^0 = f(x^0) : f_1(x^0) \in I^{<,0},\ w_1^0 = 0.25,\ f_2(x^0) \in I^{\leq,0},\ \bar{z}_2^0 = 47{,}500,\ w_2^0 = 0.75,$
$f_3(x^0) \in I^{=,0},\ f_4(x^0) \in I^{\diamond,0},\ f_5(x^0) \in I^{>,0},\ \beta_5^0 = 55{,}000,\ T_{51}^0 = 2$
$\hat{z}^0 = f(\hat{x}^0) = (44{,}544,\ 35{,}273,\ 47{,}323,\ 89{,}063,\ 54{,}571) \quad \alpha = 14{,}065$

$x^1 = \hat{x}^0, z^1 = \hat{z}^0$. Classification at
$z^1 = f(x^1) : f_1(x^1) \in I^{\diamond,1},\ f_2(x^1) \in I^{>,1},\ \beta_2^1 = 60{,}000,\ f_3(x^1) \in I^{=,1},\ f_4(x^1) \in I^{<,1},$
$w_4^1 = 0.50,\ f_5(x^1) \in I^{\leq,1},\ \bar{z}_5^1 = 50{,}000,\ w_5^1 = 0.50,\ T_{24}^1 = 1.5$
$\hat{z}^1 = f(\hat{x}^1) = (81{,}244,\ 59{,}917,\ 47{,}295,\ 37{,}648,\ 37{,}266) \quad \alpha = 10677$

Two alternatives between z^1 and $\hat{z}^1$:

Alternative 1: $z^{1(A1)} = f(x^{1(A1)}) = (53{,}142,\ 39{,}996,\ 43{,}874,\ 68{,}184,\ 45{,}495) \quad \alpha = -3308$
Alternative 2: $z^{1(A2)} = f(x^{1(A2)}) = (64{,}507,\ 47{,}848,\ 43{,}426,\ 50{,}060,\ 39{,}191) \quad \alpha = -3843$

DM prefers alternative two, and wants to stop. Pareto test and $z^{1(A2)} = f(x^{1(A2)})$ is Pareto optimal.

Table 3. Interactive results for a 100-node five-objective mo-TSP (kroABCDE100).

Ideal point: $z_i'' = (21{,}282,\ 22{,}141,\ 20{,}749,\ 21,\ 294,\ 22{,}068)$
Initialisation: $z^0 = f(x^0) = (85{,}604,\ 86{,}536,\ 84{,}795,\ 85{,}734,\ 86{,}499) \quad \alpha = 12{,}888$

Classification at $z^0 = f(x^0) : f_1(x^0) \in I^{>,0},\ \beta_1^0 = 94{,}000,\ f_2(x^0) \in I^{=,0},\ f_3(x^0) \in I^{\diamond,0},$
$f_4(x^0) \in I^{<,0},\ w_4^0 = 0.7,\ f_5(x^0) \in I^{\leq,0},\ \bar{z}_5^0 = 83{,}000,\ w_5^0 = 0.3,\ T_{14}^0 = 2$
$\hat{z}^0 = f(\hat{x}^0) = (93{,}353,\ 85{,}921,\ 168{,}602,\ 63{,}570,\ 82{,}993) \quad \alpha = 29{,}593$

$x^1 = \hat{x}^0, z^1 = \hat{z}^0$. Classification at
$z^1 = f(x^1) : f_1(x^1) \in I^{>,1},\ \beta_1^1 = 100{,}000,\ f_2(x^1) \in I^{=,1},\ f_3(x^1) \in I^{<,1},\ w_3^1 = 0.50,$
$f_4(x^1) \in I^{\diamond,1},\ f_5(x^1) \in I^{\leq,1},\ \bar{z}_5^1 = 80{,}000,\ w_5^1 = 0.50,\ T_{13}^1 = 1.5$
$\hat{z}^1 = f(\hat{x}^1) = (98{,}198,\ 85{,}618,\ 70{,}917,\ 151{,}669,\ 72{,}231) \quad \alpha = 25{,}084$

Two alternatives between z^1 and $\hat{z}^1$:

Alternative 1: $z^{1(A1)} = f(x^{1(A1)}) = (87{,}575,\ 79{,}349,\ 129{,}511,\ 86{,}335,\ 72{,}811) \quad \alpha = -6471$
Alternative 2: $z^{1(A2)} = f(x^{1(A2)}) = (89{,}133,\ 78{,}006,\ 95{,}368,\ 115{,}431,\ 68{,}520) \quad \alpha = -6872$

DM prefers alternative one, and wants to continue.

$x^2 = x^{1(A1)},\ z^2 = z^{1(A1)}$. Classification at $z^2 = f(x^2) : f_1(x^2) \in I^{>,2},\ \beta_1^2 = 95{,}000,\ f_2(x^2) \in I^{=,2},$
$f_3(x^2) \in I^{<,2},\ w_3^2 = 0.45,$
$f_4(x^2) \in I^{\leq,2},\ \bar{z}_4^2 = 86{,}000,\ w_4^2 = 0.55,\ f_5(x^2) \in I^{\diamond,2},\ T_{13}^2 = 0.5$
$\hat{z}^2 = f(\hat{x}^2) = (94{,}915,\ 79{,}089,\ 80{,}511,\ 70{,}292,\ 162{,}586) \quad \alpha = 26{,}949$
DM prefers $\hat{z}^2$, and wants to stop. Pareto test and $\hat{z}^2 = f(\hat{x}^2)$ is Pareto optimal.

Consistent with Samanlioglu *et al*.'s (2008) research, at each phase of the interactive search procedure of PIMRKGA, we let memetic RKGA stop after about 10,000 function evaluations, which corresponds to setting the population size to 500 (POP_SIZE = 500) and the stopping criterion to six (MAX_GEN = 6) as seen in Table 1. In Table 3, results of PIMRKGA for the kroABCDE100 problem are presented including an interactive scenario.

Here, the procedure starts with finding the ideal point (z_i'') and an initial weakly Pareto optimal solution (z^0). Note, that these were readily available to us based on TSPLIB and previous research (Samanlioglu *et al*. 2008). Based on this initial solution, the DM classifies five objective functions, specifies bounds, weights and tradeoff-bounding. According to this classification, a new weakly Pareto optimal solution is found ($\hat{z}^0$). The DM is not satisfied and decides to make another classification at the point $z^1 = \hat{z}^0$. According to this classification a new weakly Pareto optimal solution is found ($\hat{z}^1$). The DM wants to see two weakly Pareto optimal alternatives between z^1 and $\hat{z}^1$, so alternatives $z^{1(A1)}$ and $z^{1(A2)}$ are presented. The DM prefers the first alternative ($z^{1(A1)}$), but wants to continue from this alternative, so the DM makes another classification at point $z^2 = z^{1(A1)}$. Based on this classification, a new weakly Pareto optimal solution is found ($\hat{z}^2$). The DM is satisfied with this answer and wants to stop, so $\hat{z}^2$ is tested for Pareto optimality and found to be Pareto optimal.

5. Conclusions

In this paper, we have presented a PIMRKGA in order to find preferred (weakly) efficient solutions to a mo-TSP. Different and also superior from existing interactive multi-objective optimisation methods or interactive, preference-based multi-objective evolutionary algorithms in the literature, PIMRKGA integrates classification of objective functions into five classes, specification of bounds and weighting coefficients and especially indifference trade-off information in one method. Thus, the DM has more flexibility and control over the interactive search process. This integration has never been done before and RKGA has never been combined with any other interactive method to solve NPC problems, especially mo-TSP.

Specifically, the methodology proposed here differs from other existing methodologies in several ways. First, this methodology allows the DM more flexibility and control during the search since the classification phase of PIMRKGA has five classes. NIMBUS is the closest methodology in that sense since NIMBUS also classifies

objective functions into five classes; however, our classification is different from NIMBUS because NIMBUS restricts objective functions in $I <$ to be minimised as far as possible and functions in $I^{\leq}$ to be minimised to the aspiration level ($\bar{z}_i^h$). In the proposed methodology, objectives in $I^{<,h}$ and $I^{\leq,h}$ are to be minimised as much as possible with the restriction that objectives in $I^{\leq,h}$ must reach a value that is less than or equal to a specified bound ($\bar{z}_i^h$). As a result, problem (7) is different from the ''vector'' or ''scalar'' sub-problems NIMBUS uses. Furthermore, NIMBUS does not utilise the idea of indifference tradeoffs. There are other interactive methods based on indifference tradeoffs in the literature, for instance, GDF (Geoffrion *et al*. 1972) and SPOT (Sakawa 1982); however, these methods do not allow the DM to classify objective functions into five classes. The initialisation phase of NIMBUS and PIMRKGA are also different. NIMBUS takes the DM-identified starting point and projects it onto the feasible region by solving an auxiliary problem. The weakly Pareto optimal counterpart of this point is calculated by assuming all objective functions belong to the class $I <$ and solving the selected ''vector'' or ''scalar'' sub-problem. Finally, NIMBUS uses different local and global optimisers whereas the memetic RKGA in PIMRKGA is specifically designed for mo-TSP.

In conclusion, we claim that integration of objective classification with tradeoff-bounding found in PIMRKGA provides the DM with more flexibility and control over the interactive search process than NIMBUS and other existing methods. We do not claim that PIMRKGA would work the best for all multi-objective problems since the memetic RKGA, the global optimiser of the interactive method, is designed specifically for mo-TSPs.

Since PIMRKGA is designed specifically for mo-TSP, future work includes adaptation of PIMRKGA to different versions of mo-TSP such as generalised TSP, and other combinatorial optimisation problems where RKGA can be used, such as scheduling problems. Designing interactive methods that take into account fuzziness of objective functions, goals and bounds specified by the DM, and integrating with PIMRKGA can also be an interesting area of research.

References

Bagchi, T.P., Gupta, J.N.D., and Sriskandarajah, C., 2006. A review of TSP based approaches for flowshop scheduling. *European Journal of Operational Research*, 169 (3), 816–854.

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

Benayoun, R., *et al*., 1971. Linear programming with multiple objective functions: STEP method (STEM). *Mathematical Programming*, 1 (1), 366–375.

Brintrup, A.M., Ramsden, J., and Takagi, H., 2008. Ergonomic chair design by fusing qualitative and quantitative criteria using interactive genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 12 (3), 343–354.

Brintrup, A.M., Ramsden, J., and Tiwari, A., 2007. An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. *Computers in Industry*, 58 (3), 279–291.

Caballero, R., *et al*., 2006. i-MOM: an interactive multi-objective metaheuristic method. Application to a real problem. *In*: *Proc. MOPGP06: 7th International Conference on Multi-Objective Programming and Goal Programming*, 12–14 June 2006, Loire Valley, France.

Cattrysse, D., *et al*., 2006. Automatic production planning of press brakes for sheet metal bending. *International Journal of Production Research*, 44 (20), 4311–4327.

Coello Coello, C.A., 2002. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191 (11–12), 1245–1287.

Deb, K., 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186 (2–4), 311–338.

Deb, K. and Chaudhuri, S., 2003. I-EMO: an interactive evolutionary multi-objective optimization tool. Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, PIN 208016, India, KanGAL Report Number 2005003, 1–9.

Dolgui, A. and Pashkevich, A., 2006. Cluster-level operations planning for the out-of-position robotic arc-welding. *International Journal of Production Research*, 44 (4), 675–702.

Ecemis, M.İ., *et al*., 2008. A drug candidate design environment using evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 12 (5), 591–603.

Emmons, H. and Mathur, K., 1995. Lot sizing in a no-wait flow shop. *Operations Research Letters*, 17 (4), 159–164.

Ganesh, K. and Narendran, T.T., 2008. TASTE: a two-phase heuristic to solve a routing problem with simultaneous delivery and pick-up. *International Journal of Advanced Manufacturing Technology*, 37 (11–12), 1221–1231.

Geoffrion, A.M., Dyer, J.S., and Feinberg, A., 1972. An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management Science*, 19 (4), 357–368.

Hansen, M.P., 2000. Use of substitute scalarizing functions to guide a local search based heuristic: the case of moTSP. *Journal of Heuristics*, 6 (3), 419–431.

Huang, H., Tian, Z., and Zuo, M.J., 2005. Intelligent interactive multiobjective optimization method and its application to reliability optimization. *IIE Transactions*, 37 (11), 983–993.

Jaszkiewicz, A., 2002. Discrete optimization, genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137 (1), 50–71.

Kim, B., *et al*., 2003. Clustering-based order-picking sequence algorithm for an automated warehouse. *International Journal of Production Research*, 41 (15), 3445–3460.

Knowles, J. and Corne, D., 2005. Memetic algorithms for multiobjective optimization: issues, methods and prospects. *In*: W.E. Hart, J.E. Smith and N. Krasnogor, eds. *Recent advances in memetic algorithms*. *Studies in fuzziness and soft computing, 166*. Berlin: Springer-Verlag, 313–352.

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

Leon, V.J. and Peters, B.A., 1996. Replanning and analysis of partial setup strategies in printed circuit board assembly. *International Journal of Flexible Manufacturing Systems*, 8 [Special Issue on Electronics Manufacturing], 389–412.

Lee, M. and Kwon, K., 2006. Cutting path optimization in CNC cutting processes using a two-step genetic algorithm. *International Journal of Production Research*, 44 (24), 5307–5326.

Lee, S.D. and Kuo, Y.C., 2008. Exact and inexact solution procedures for the order picking in an automated carousel conveyor. *International Journal of Production Research*, 46 (16), 4619–4636.

Michalowski, W. and Szapiro, T., 1992. A bi-reference procedure for interactive multiple criteria programming. *Operations Research*, 40 (2), 247–258.

Miettinen, K., 1999. *Nonlinear multiobjective optimization*. International series in operations research & management science. Massachusetts, USA: Kluwer Academic Publishers.

Miettinen, K. and Makela, M.M., 2000. Interactive multiobjective optimization system WWW-NIMBUS on the Internet. *Computers & Operations Research*, 27 (7–8), 709–723.

Miettinen, K. and Makela, M.M., 2002. On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24 (2), 193–213.

Miettinen, K., Makela, M.M., and Toivanen, J., 2003. Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms. *Journal of Global Optimization*, 27 (4), 427–446.

Miettinen, K. and Makela, M.M., 2006. Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170 (3), 909–922.

Nakayama, H. and Furukawa, K., 1985. Satisficing trade-off method with an application to multiobjective structural design. *Large Scale Systems*, 8 (1), 47–57.

Narula, S.C., Kirilov, L., and Vassilev, V., 1994. Reference direction approach for solving multiple objective nonlinear programming problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 24 (5), 804–806.

Onwubolu, G.C. and Clerc, M., 2004. Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization. *International Journal of Production Research*, 42 (3), 473–491.

Quan, G., *et al*., 2007. Searching for multiobjective preventive maintenance schedules: combining preferences with evolutionary algorithms. *European Journal of Operational Research*, 177 (3), 1969–1984.

Reinelt, G., 1995. *TSPLIB* [online]. Available from: http://www.comopt.ifi.uni-heidelberg.de/software/TSPLIB95/ [Accessed 21 July 2011].

Ryan, E., Azad, M.A., and Ryan, C., 2004. On the performance of genetic operators and the random key representation. *In*: *Genetic programming*, Lecture Notes in Computer Science, 3003/2004, 162–173.

Sakawa, M., 1982. Interactive multiobjective decision making by sequential proxy optimization technique: SPOT. *European Journal of Operational Research*, 9 (4), 386–396.

Samanlioglu, F., *et al*., 2007. A hybrid random-key genetic algorithm for a symmetric traveling salesman problem. *International Journal of Operations Research*, 2 (1), 47–63.

Samanlioglu, F., Kurz, M.E., and Ferrell, 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*, 20–24 May 2006, Orlando, Florida, Institute of Industrial Engineers.

Samanlioglu, F., Ferrell, W.G., and Kurz, M.E., 2008. A memetic random-key genetic algorithm for a symmetric multi-objective traveling salesman problem. *Computers & Industrial Engineering*, 55 (2), 439–449.

Snyder, L.V. and 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., 1982. A mathematical basis for satisficing decision making. *Mathematical Modeling*, 3 (25), 391–405.

WWW-NIMBUS [online]. Available from: http://nimbus.mit.jyu.fi/ [Accessed 21 July 2011].

Yadollahpour, M.R., *et al*., 2009. Guided local search algorithm for hot strip mill scheduling problem with considering hot charge rolling. *International Journal of Advanced Manufacturing Technology*, 45 (11–12), 1215–1231.