Application of an Evolutionary Algorithm-based Ensemble Model to Job-Shop Scheduling

Choo Jun TAN, Siew Chin NEOH, Chee Peng LIM, Samer HANOUN, Wai Peng WONG, Chu Kong LOO, Li ZHANG, Saeid NAHAVANDI

Abstract In this paper, a novel evolutionary algorithm is applied to tackle job-shop scheduling tasks in manufacturing environments. Specifically, a modified micro genetic algorithm (MmGA) is used as the building block to formulate an ensemble model to undertake multi-objective optimisation problems in job-shop scheduling. The MmGA ensemble is able to approximate the optimal solution under the Pareto optimality principle. To evaluate the effectiveness of the MmGA ensemble, a case study based on real requirements is conducted. The results positively indicate the effectiveness of the MmGA ensemble in undertaking job-shop scheduling problems.

Keywords Multi-objective optimisation, evolutionary algorithm, ensemble models, job-shop scheduling

1 Introduction

Computational intelligence models have been widely studied to tackle a variety of industrial optimization and related problems. Examples include the use of an artificial neural network-based method for shape optimization in structural design [71, 72], a hybrid differential evolutionary-based algorithm for welded beam design [73], and a particle swarm optimization algorithm for structural damage detection pertaining to the finite element model of a Timoshenko beam [79]. More recent evolutionary methods, which include nature-inspired algorithms using gravitational search [77, 78] and charged system search [78], have also been investigated for structural design and optimization of vehicle components. Different meta-heuristics models for vehicle crashworthiness as well as noise, vibration, and harness optimization have been studied and compared [74]. Other methods, which include the topology optimization approach, have been adopted for design and optimization tasks in vehicular technology [75, 76]. Motivated by the success of computational intelligence models, particularly evolutionary-based algorithms, in solving industrial optimization problems, we focus on the use of single and ensemble evolutionary algorithms for real-world industrial job-shop scheduling tasks from the multi-objective optimization perspective in this research.

Job-shop scheduling is a Non-deterministic Polynomial-time hard (NP-hard) problem in computational complexity theory [8], [44], [23]. Job-shop scheduling comprises a finite set of jobs that needs to be processed either on single or multiple machines [27], [24], subject to a number of performance measures. As an example, in a multi-stage manufacturing process, each scheduled job needs to go through several operations to become a finished product. In a single machine scenario, any machine breaks down could cause delay of the entire production, as reported in [29], [27].

Most of the investigations pertaining to job-shop scheduling aim to optimise only a single objective [41], [45]. Indeed, the survey in [8] reveals that multi-objective problems rank second after the makespan minimisation problems in Artificial Intelligence (AI) research trends between 1997 and 2012. It is worth noting that while most AI-based solutions involving the Genetic Algorithm (GA) and agent-based systems have shown an increasing trend in the literature, only few of them focus on solving real-world industrial problems [8]. Job-shop scheduling problem is identified as one of the decision making parts in scheduling optimisation in another survey [23]. It has been used to decide the operation sequencing with different effective chromosome representations in the Multi-Objective Evolutionary Algorithm (MOEA).

Based on the aforementioned motivation, we investigate a real-world job-shop scheduling problem of a manufacturing company in this study. Specifically, the problem deals with multi-objective functions, and optimisation is performed using a MOEA-based model. The enumeration method is used to produce theoretically optimal solutions as the baseline results for comparison purposes. Before presenting the details, the related literature and the background of the proposed model are presented in the next section. The case study is then elaborated in details, which is followed by the results and discussion. A summary covering the conclusions and suggestions for further work is presented in the last section.

2 Heuristic-based Scheduling Problems

Methods for solving scheduling problems can be broadly classified into two categories [38]. The first is enumeration-based methods that utilise dynamic programming or branch-and-bound techniques to find the optimal solution. The second is heuristic-based search methods that find near optimal solutions. The enumeration-based methods eliminate candidate solutions by employing restrictive criteria. However, it is well-known that the enumeration-based methods are time-consuming, and are not efficient for large-scale problems [62], [49]. State space explosion as a result of increasing the number of state variables is the key limitation of dynamic programming techniques, owing to increasing number of state variables [62]. On the other hand, the long execution time owing to the number of variables involved in branching as well as the strategies used for bounding is the key limitation of branch-and-bound techniques [49]. In regards to heuristic-based search methods, the candidate solutions are produced by meta-heuristics searches, e.g. single-objective optimisation using constructive heuristics [42]. Another example is a hybrid genetic programming and hyper-heuristic method proposed in [41]. The method is used in a single-objective job-shop scheduling problem for evolving dispatch rules, whereby the results show the effectiveness of the genetic programming-based method over hybrid genetic algorithm (GA) methods.

In addition to the GA [52], examples of recent and classical meta-heuristics methods used to solve optimisation problems include Cuckoo search [28] and simulated annealing [29], [27], respectively. In cellular manufacturing systems, the problem of time-tabling in a flowline was investigated in [52]. A number of objectives were examined, i.e., minimising machine idle-time, makespan, and total flow-time. In [29], [27], a job-shop scheduling problem on single machines was studied using a greedy- heuristic model with SA and a Pareto-based SA model. Two objectives were considered, i.e., minimising material costs and tardiness. The same job-shop scheduling problem was evaluated using a Pareto-based Cuckoo search model in [28]. In a recent study [64], the GA was evaluated against a branch-and-bound method in a single objective optimisation problem. The experimental results indicated that the GA was able to perform well in minimising makespan under a job-shop scheduling problem with two machines.

A hybrid Particle Swarm Optimisation (PSO) algorithm was used to solve multi-objective job-shop scheduling problem in [66], i.e. minimising makespan, working time spent on a machine, and total working time over all machines. The multi-objective functions were combined using the weighted-sum method. The results showed that PSO was able to handle large scale job-shop scheduling problems effectively. A similar multi-objective method was described in [63]. Specifically, a multi-objective combinatorial model with the variable neighborhood descent meta-heuristic method was employed to evaluate the effectiveness of handling job-shop scheduling problems. In [50], a hybrid SA and GA model was proposed. The hybrid model was used to tackle the problem pertaining to minimisation of makespan, workload of the most loaded machine, and the total workload of all machines. The proposed hybrid model was able to reduce the computational time in achieving the optimal solutions.

In [61], multi-objective optimisation in the context of scheduling was presented. A single-machine job-shop scheduling problem was undertaken with a polynomial algorithm. Two objective functions, i.e. minimising the mean flow-time and maximising tardiness, were optimised simultaneously using two rule-based methods. The efficient points or ideal solutions were enumerated, and the results were compared with those from the rule-based methods. The optimum values obtained for both objectives were close to the enumerated efficient points, with negligible computational time. A Pareto-based MOEA, i.e. Non-dominated Sorting Genetic Algorithm II (NSGA-II), was employed to tackle problems related to makespan and delay of schedules in [65]. Both objective functions were minimised simultaneously for various job-shop scheduling scenarios. Another Pareto-based evolutionary algorithm [13] was used to tackle a Multi-objective Optimisation Problem (MOP). The objectives comprised maximising the workload while minimising both total workload and makespan. Table 1 shows a variety of recent models used to tackle MOPs in the manufacturing domain.

From the perspective of machine learning, the use of ensemble methods is useful for performance enhancement. DEMETER [25], as an example, is an effective ensemble-based weather prediction system. It provides forecasts of atmospheric states with probabilistic estimation. In another study, an ensemble method comprising tractable models is used to tackle the problem of articulated human pose estimation in video streams [47]. Using the human VideoPose2.0 data set, the proposed ensemble model shows better performance as compared with the results from a simple max-marginal combination algorithm.

In [58], a hybrid machine learning algorithm, i.e. the Radial Basis Function (RBF) neural network, with an evolutionary algorithm for undertaking feature selection and classification problems, was proposed. The proposed model utilises the divide-and-cooperative mechanism from the evolutionary algorithm to process the hidden layer structure and dominant features of the RBF network. The model is able to produce better accuracy and reduce the number of features involved in tackling complicated multi-objective classification tasks.

Two NP-hard scheduling problems were tackled using an ensemble-based GA model in [11]. The ensemble model inherits the crossover and mutation procedures of its predecessor, i.e. a Self-Guided GA [10], while the evolutionary process is guided by the incorporated probabilistic model. The ensemble model outperforms individual Estimation of Distribution algorithms as well as other algorithms in the literature [11]. However, the scheduling problem combines both the earliness and tardiness as a single-objective function. In this study, the Pareto-based MOEA is adopted. Specifically, an ensemble of modified micro genetic algorithms [55] is adopted for tackling multi-objective job-shop scheduling problems. The background of the Pareto principles and the ensemble model is explained in the next section.

Table 1 Heuristic-based models for Multi-objective Job-Shop Scheduling Problems

Year	Model	Objectives
2014	Hybrid discrete firefly algorithm	Maximising the completion time, workload of the critical machine and total workload of all machines [31].
2014	Improved sheep flock heredity algorithm and artificial bee colony algorithm	Minimising the makespan and total flow-time [9].
2015	Heuristic-based ϵ-constraint method	Minimising the makespan and sum of flow-time, and maximising tardiness [2].
2015	Bi-Variant block-based estimation and distribution algorithm	Minimising the makespan and total flow-time [59].
2015	ELECTRE-based multi-objective GA	Minimising the makespan and overtime costs [46].
2015	Imperialist competitive algorithm	Minimising the makespan and total tardiness [32].
2015	Modified micro GA	Minimising tardiness and maximising cost saving [57].
2016	Goal-guided multi-objective based models	Minimising tardiness conflicts with overtime; tardiness conflicts with robustness, and overtime with robustness promotes each other. [67].
2016	Discrete harmony search algorithm	Minimising the makespan of machine and the mean of earliness and tardiness [22].

3 Background of Work

3.1 Pareto Optimality Principle

Pareto optimality advocates the 80-20 principle [43], e.g. 80% of wealth is held by 20% of the population, or 80% of effects originate from 20% of causes. This principle is useful to provide explanation pertaining to many real-world problems, e.g. sport ranking [19], biological research [48], [51], and pharmacological studies [6], [34]. A solution is said to be Pareto-optimal (i.e. a non-dominated solution) when one objective cannot be further improved without causing a simultaneous degradation in at least another objective [15]. The Pareto front (**pf**) refers to a Pareto optimal set of non-dominated solutions.

MOPs entail problems that require multiple objectives to be satisfied simultaneously [7], [33]. Some examples of MOPs are maximising profit subject to production cost and time, minimising loss in emission and transmission as well as cost in economic dispatch problems. Mathematically, an optimisation (either maximisation or minimisation) problem requires finding $\mathbf{x} \in \mathbb{R}^b$ of free parameters, $\mathbf{x} = \{x_0,\ldots,x_b\}$, $x_i \in \mathbb{R}$. The optimisation problem is subject to $f : \mathbb{R}^b \rightarrow \mathbb{R}$, which is also known as the objective function, with a subset of b-tuples of real numbers [4]. For a minimisation problem, $f(\mathbf{x}) \rightarrow$ *minimisation* derived under n number of objective functions in $\mathbf{f}(\mathbf{x})$ for undertaking an MOP can be derived as follows.

$$\mathbf{pf} = \{\mathbf{f}(\mathbf{x}^*) \mid \mathbf{x}^* \in \eta\} \quad (1)$$

$$\mathbf{pf}_{true} \equiv \{\mathbf{f}(\mathbf{x}) \mid \forall \mathbf{x} \in \mathbf{pf}\} \quad (2)$$

Subject to

$$\mathbf{f}(\mathbf{x}) = f_1(\mathbf{x}),\ldots,f_i(\mathbf{x}),\ldots,f_n(\mathbf{x})\}, \quad (3)$$

$$\mathbf{p} \equiv \mathbf{f}(\mathbf{x}^*) = \{f_1(\mathbf{x}^*),\ldots,f_i(\mathbf{x}^*),\ldots,f_n(\mathbf{x}^*)\} \quad (4)$$

The Pareto front [17] or the objective vector [68], i.e. **pf**, is derived based on Equation 1. Note that **pf** is also known as the Pareto optimal set [69] [17] or Pareto set [68]. The solutions in the Pareto optimum set i.e. $\eta \equiv \{\mathbf{x} \in \mathrm{F} \mid \neg\exists \mathbf{x}^* \in \mathrm{F}, f_i(\mathbf{x}^*) \preccurlyeq f_i(\mathbf{x})\}$ indicate the non-dominated solutions [53] in accordance with strictly Pareto dominance, which is denoted using symbol $\preccurlyeq$ in a feasible region, F. Note that x^* dominates x, i.e. $x^* \preccurlyeq x$, as in the Pareto optimum solutions, $\mathbf{p} = \{p_1,p_2,\ldots\}$, corresponding to decision variables $\mathbf{x} = \{x_1,x_2,\ldots\}$. $\mathbf{x}^*$ is said to be decision variables of the Pareto optimal set in a minimisation MOP if and only if $\forall_{i \in n, x \in \mathrm{F}} \left(f_i(\mathbf{x}^*) \preccurlyeq f_i(\mathbf{x})\right)$. A point $\mathbf{x}^* \in \mathbf{x}$ is said to belong to a strictly Pareto optimal set if there are no other $x \in \mathbf{x}$ and $x \neq x^*$ such that $f_i(\mathbf{x}^*) \preccurlyeq f_i(\mathbf{x})$ [16].

MOP-based **pf** is derived using **p**. Specifically, the search space contains **pf**. The values pertaining to an objective function corresponds to **x**, and is subject to the Pareto dominance concept. Therefore, $\mathbf{f} : \mathbb{R}^b \rightarrow \mathbb{R}^n$, consists of $n \geq 2$, and $f : \mathbb{R}^b \rightarrow \mathbb{R}$. Sets $\mathbb{R}^b$ and $\mathbb{R}^n$ represent the decision variable space and objective function space, respectively. The optimal solutions are **pf**, which are said to be the non-dominated solutions for solving MOPs.

Notice that $\mathbf{pf}_{true}$ consists of **pf** (Equation 1), which is a solution set generated from an EA using S. In other words, $\mathbf{pf}_{true}$ contains the solution points that satisfy $\mathbf{f}(\mathbf{x})$ (i.e. solutions from Equation 3). They are known as the exact solutions in solving an MOP based on the enumeration method [40], [39]. For multi-objective job-shop scheduling problems, the solutions from the enumeration method represent the ideal (optimal) solutions, which are produced based on a brute-force search strategy.

An EA is capable of searching and identifying a set of possible solutions to form **pf** [16]. A particular objective function, i.e. i, from Equation 1 to be optimised using a b-tuple of decision vector ($\mathbf{x}^*$) can be formulated as follows.

$$f_i : \mathbf{x}^* \rightarrow \mathbb{R} \quad (5)$$

Let I be the population space of the EA. Each individual of the population, $a \in I$, represents a candidate solution of the optimisation problem. A transformation to yield a fitness value (a real number) is derived as follows [5].

$$\Phi : I \rightarrow \mathbb{R} \quad (6)$$

The fitness function of an MOEA consists of multiple single-objective problems, i.e., [16].

$$\Phi' : I \rightarrow \mathbb{R}^n, n \geq 2 \qquad (7)$$

where I indicates the initial parent population and n indicates multiple objective functions. In essence, the MOEA yields **pf** that comprises the optimal solution(s) [20] by taking into consideration multiple objective functions, as in Equation 7.

There are a number of indicators to measure the performance of an MOEA with respect to the Pareto optimality in tackling MOPs. Among them, the Generational Distance (I_{gd}) [18], [20] is widely used. I_{gd} provides an indication pertaining to the distance from the solutions in **pf** with respect to those in $\mathbf{pf}_{true}$ as follows [18], [20].

$$I_{gd}(\mathbf{pf}_{true}, \mathbf{pf}) = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \qquad (8)$$

where $n = |\mathbf{pf}|$, d_i is the Euclidean distance between a solution and the nearest solution in $\mathbf{pf}_{true}$. When all the **pf** solutions reside in $\mathbf{pf}_{true}$, then $I_{gd}(\mathbf{pf}_{true}, \mathbf{pf}) = 0$.

3.2 Modified Micro Genetic Algorithm (MmGA)

A Modified micro GA (MmGA) [54] is employed as the building block to develop an ensemble model for tackling multi-objective job-shop scheduling problems in this study. MmGA is an extension of the traditional micro GA (mGA) [17].

mGA was formulated based on the GA principles [26], but with a small population size. It usually contains only three to six chromosomes in its population. Based on the findings in [26], [17], [36], [12], it has been shown that mGA is capable of achieving convergence in undertaking optimisation problem with arbitrary chromosome lengths, in spite of its small population size. This is owing to two main properties of mGA, which is designed to overcome the limitations associated with its small population size in solving MOPs [17], viz, (i) a re-initialisation procedure for its population size in random generation of chromosomes; (ii) a special population memory structure that comprises both **rm** (replaceable memory) and **irm** (irreplaceable memory) components.

Figure 1 depicts that mGA has two operational cycles: (i) a nominal evolution cycle executes the generic crossover, mutation, and selection operations with re-initialisation of the working population memory within a pre-specified number of rounds; (ii) an outlier evolution cycle that repeats the entire nominal evolution cycle within a pre-specified number of rounds in producing its search results, i.e. **pf**. It should be noted unlike the traditional GA, mGA has a smaller population size, and it uses a special strategy to preserve diversity [1]. On top of the normal mutation operation, a re-start strategy is utilised to introduce diversity in the mGA population overtime [17], [37], [36].

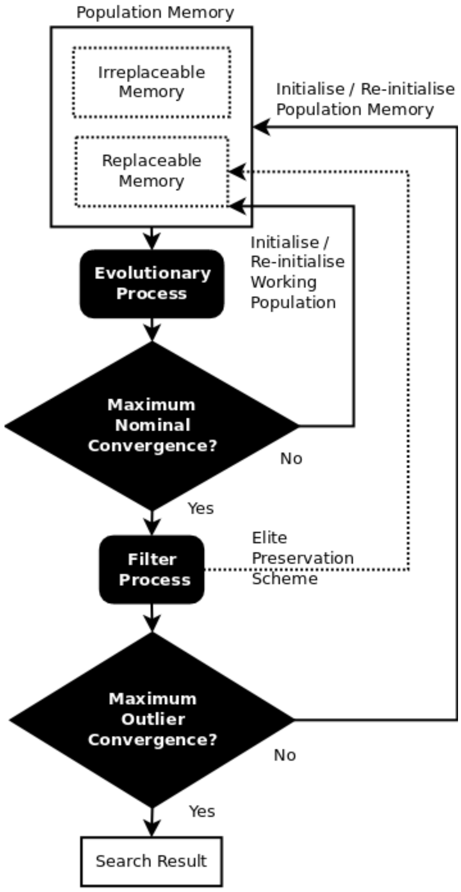

Fig. 1 A schematic diagram of the mGA, adopted from [17].

On the other hand, MmGA is derived to improve its convergence capability towards $\mathbf{pf}_{true}$ without sacrificing the salient properties of mGA. To achieve this aim, two modifications are introduced, viz. an new elitism method based on the principle of NSGA-II and a new rule for population formation [54]. In the new elitism method, a user-defined elite-preservation size (ω) of selected chromosomes ($\mathbf{x}$) and target chromosomes ($\mathbf{y}$) are derived. The elitism method produces a vector $\mathbf{z}$ as its outcome, which consists of ω-elite chromosomes, i.e. $\mathbf{z} = \{\neg(\mathbf{x} \cap \mathbf{y}) \cup \mathbf{y}\}$[54]. Note that $\mathbf{x}$ and $\mathbf{y}$ are the vectors of chromosomes, which exist in the evolutionary and filter processes within both nominal and outlier evolution cycles, respectively. The population of chromosomes is re-initialised ($\mathbf{p}$) using four main components in MmGA, i.e. $\mathbf{p} = \{\mathbf{irm} \cup \preccurlyeq_S(\neg(\mathbf{rm} \cap \mathbf{irm})) \cup \preccurlyeq_S(\neg(\mathbf{rm} \cap \mathbf{irm} \cap \breve{\mathrm{P}})) \cup \preccurlyeq_S(\neg(\mathbf{rm} \cap \mathbf{irm} \cap \breve{\mathrm{P}} \cap \mathbf{r}))\}$ [54]. $\mathbf{p}$ is generated using the Pareto dominance-based merge sort procedure ($\preccurlyeq_S$), a potential solutions for the MOP ($\breve{\mathrm{P}}$), and a vector of $\mathbf{r}$ newly randomised chromosomes, as well as existing replaceable memory ($\mathbf{rm}$) and irreplaceable memory ($\mathbf{irm}$) in mGA.

3.3 Modified Micro Genetic Algorithm Ensemble

To improve the robustness of MmGA, an ensemble MmGA model is formed for tackling MOPs [56], [55]. Figure 2 shows a schematic diagram of the MmGA ensemble, which is used to search for $\mathbf{pf}$ in solving multi-objective job-shop scheduling problems. To construct an ensemble model, multiple individual MmGA entities are grouped together with a decision combination module.

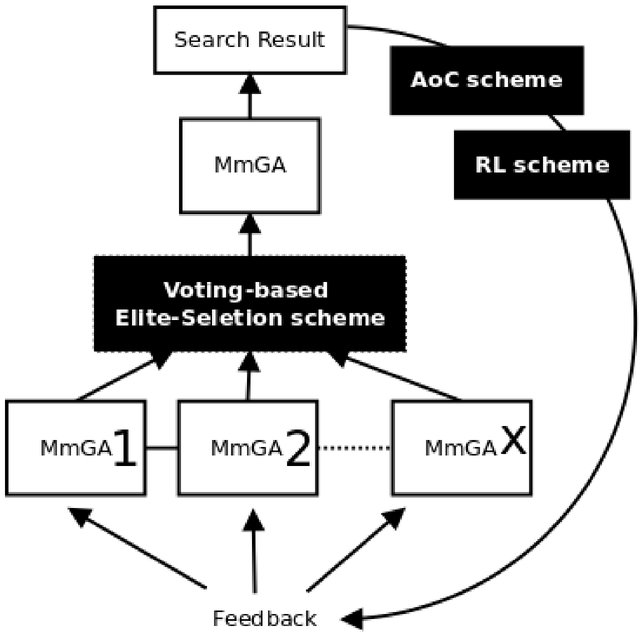

Fig. 2 A schematic diagram of the proposed MmGA ensemble.

In this study, we aim to improve convergence of the MmGA solutions towards $\mathbf{pf}_{true}$ by using the ensemble model. A number of useful techniques are introduced into the MmGA ensemble to realize this objective [55]. Firstly, an elite selection technique based on majority voting is devised. Both a reinforcement learning (RL) technique and an Apportionment of Credit (AoC) technique are used for improving the **rm** structure. The RL technique is also applied to **rm** adoption ratio. Then the Euclidean distance is utilised to facilitate replacement of the final **rm** components. The candidate solution with the lowest value of MOEA performance indicator (i.e. as stated in the Definition 5) is identified for determining the winning offspring (i.e. elite) in the process of forming **pf**. This elitism technique is used to form the accumulated **rm** component. The AoC scheme is used to evaluate the performance of MmGA members in the ensemble. Subsequently, RL is deployed for instilling the reward-penalty scheme, i.e. MmGA members with good quality **pf** are rewarded while those with poor quality **pf** are penalized. It should be noted that the RL-based selection of the **rm** component affects the formation of subsequent **pf** created by MmGA.

The expected output of the MmGA ensemble represents **pf** from MmGA members, which is affected by MmGA winner in each round of feedback. Specifically, the MmGA ensemble produces **pf** = $\left\{ \{pf_1,\ldots,pf_\alpha\} \parallel pf_i \rightarrow \tilde{\mathbf{pf}} \right\}$ where $\tilde{\mathbf{pf}} = \{\tilde{pf}_1,\ldots,\tilde{pf}_\beta\}$. Note that α and β denote the maximum feedback round and the ensemble size of MmGA, respectively. **pf** consists of α-feedback round search results, which is based on β-output from the MmGA members. As such, it is an extended derivation of **pf** (Equation 1) with multiple MmGA models in an ensemble structure. The next section describes the details of the job-shop scheduling case study using both MmGA and the ensemble models.

4 A Case Study

To demonstrate the usefulness of the MmGA ensemble, we examined a real-world multi-objective job-shop scheduling problem with information solicited from a company in Australia. The task required scheduling a set of (n jobs) for a single machine. The machine was available at all times, but had the capability of processing only one job at any specific time. An example is provided to illustrate the job-shop scheduling problem.

In general, we first generate a total of n jobs. Each job i, $i = 1,\ldots,n$, requires a specific processing time, pi, and has a due date, di. Based on the completion day, ci, we need to calculate cost-saving, Si of job i, its tardiness, $Ti = ci - di$, as well as total earliness, $Ei = di - ci$. As such, each job has parameters Ti, Ei, and $Si \in \mathbb{R}^+$. Therefore, this job-shop scheduling problem entails finding the processing order of n jobs subject to MOEA fitness functions, i.e. as stated in the Definition 4.

Maximising total cost-saving (S dollars),

which is formulated as a minimisation function of

$$f_1 = -1 \times \sum_{i=1}^{n} S_i \qquad (9)$$

Minimising total tardiness (T days) of

$$f_2 = \sum_{i=1}^{n} T_i \qquad (10)$$

Maximising total earliness (E days),

which is formulated as a minimisation function of

$$f_3 = -1 \times \sum_{i=1}^{n} E_i \qquad (11)$$

In accordance with the company policy, when a job is completed on or before its due date, tardiness does not arise. Otherwise, tardiness of a job, T_i, occurs based on the difference in days, which is computed from the difference between the completion time of its last operation and its due date. A similar formulation is applied to determine earliness of a job. As such, when a job is completed after its due date, earliness does not arise. Otherwise, earliness of a job, E_i, occurs based on the difference in days, which is computed from the difference between the completion time of its last operation and its due date. Two examples to highlight the main computation of the job-shop scheduling problems are presented as follows.

Example 1 *A 5-job problem is generated, as shown in Table 2. The starting date of all five jobs is the same, i.e., the schedule start date: 18/09/2015. Each job has a given due date.*

Table 2 Requirements of the 5-job problem

Job	**Due date (Day/Month/Year)**	**Duration for Completion (Day.Hours:Minutes)**
1	19/09/2015	07:40
2	24/09/2015	06:40
3	24/09/2015	15:00
4	01/10/2015	1.01:00
5	26/09/2015	09:00

Based on the information, tardiness and earliness of each job, i.e. T_i and E_i, are computed. As an example, the order of 5 jobs is $\mathbf{x}' = \{1,4,5,2,3\}$. Let the daily working hour start at 09:00, and finish at 17:00. Table 3 shows an example of the detailed schedule and the performance calculation based on the requirements in Table 2. The starting time and date of Job 1 is 09:00 on 18/09/2015, and it has been completed after 7 hours and 40 minutes, i.e. at 16:40 on 18/09/2015. Subsequently, the Job 4 is taken up, and it has been completed after 25 hours, i.e. 09:40 on 22/09/2015 (based on 8 working hours per day). The earliness (in day) of each job is computed by comparing the given due date and the completion date (last column of Table 3). Notice that Jobs 1, 4 and 5 result in earliness ranging from 1 to 10 days, while Jobs 2 and 3 incur tardiness of 1 and 2 days, respectively. All tardiness and earliness scores are shown in Table 3.

Table 3 A numerical example of a 5-job problem

Job	Start Date Time	Completion Date Time	*T* Days	*E* Days
1	18/09 09:00	18/09 16:40	0	19/09 - 18/09 = 1
4	18/09 16:40	22/09 09:40	0	01/10 - 22/9 =10
5	22/09 09:40	24/09 10:40	0	26/09 - 23/09 = 3
2	24/09 10:40	25/09 09:20	25/09 - 24/09 = 1	0
3	25/09 09:20	26/09 16:40	26/09 - 24/09 =2	0
		Total	3	14

One of the important factor considered by the company is cost-saving of materials. There is a cost-saving factor for the material used when two jobs are performed in sequence, i.e. $S_{v,w}$ for Jobs *v* and *w*.

Example 2 *Table 4 shows the material cost-saving matrix for each pair of jobs. Equation 9 is used to calculate the cost-saving factor.*

As an example, the order of 5 jobs is $\mathbf{x}' = \{(1,4),(5,2),3\}$, where pair (1,4) with $S_{1,4} = 1.95$ is first conducted, leading to $\mathbf{x}' = \{-,-,5,2,3\}$. Next, pair (5,2) with $S_{5,2} = 1.14$ is conducted, leading to $\mathbf{x}' = \{-,-,-,-,3\}$, i.e. Job 3 is standalone. As such, the total cost-saving is 1.95 + 1.14 = 3.09.

Table 4 Cost-saving matrix of the 5-job problem

$S_{v,w}$	Job 1	Job 2	Job 3	Job 4	Job 5
Job 1	0.00	1.70	2.16	1.95	1.80
Job 2	1.70	0.00	2.60	1.90	1.14
Job 3	2.16	2.60	0.00	3.36	2.61
Job 4	1.95	1.90	3.36	0.00	2.46
Job 5	1.80	1.14	2.61	2.46	0.00

Table 5 shows the optimal solutions obtained using the enumeration method. They serve the Definition 3 as well as the baseline results for performance evaluation and comparison with those from the MmGA models.

Table 5 Non-dominated solutions covering *S*, *T* and *E* from the enumeration method for the 5-job problem

Job Order	*S* (dollars)	*T* (days)	*E* (days)
(1 - 5) - 2 - (3 - 4)	5.16	0	19
(1 - 5) - (4 - 3) - 2	5.16	2	15
(1 - 5) - (3 - 4) - 2	5.16	1	16
(4 - 3) - (1 - 5) - 2	5.16	5	13
1 - (4 - 5) - (2 - 3)	5.06	1	13
1 - (3 - 2) - (4 - 5)	5.06	0	16
(4 - 1) - (3 - 5) - 2	4.56	4	12
(1 - 4) - 2 - (3 - 5)	4.56	0	13
(4 - 1) - (3 - 2) - 5	4.55	3	12
1 - (4 - 3) - (2 - 5)	4.5	0	12

To understand the MmGA ensemble used in this study, its algorithm is shown in Figure 3. Lines 1 to 10 show the MmGA feedback (*fb*) cycles, which contain the objective functions (i.e., Equations 9 to 11) and the associated fitness indicators (i.e., *S*, *T*, and *E*).

```
1:  repeat
2:   for each i-th individual ensemble MmGA do
3:   Let x_i' = {a_1,...a_n}
4:   ∴ x_i ← a processing order of job i based on x_i'
5:   pf_i ← Pareto front of MmGA based on x_i
6:   end for
7:   elite_fb ← elite-selector based on pf_fb where pf_fb = {..., pf_i,...}
8:   pf_fb ← preserve good x* of elite_fb based on pf_fb
9:   Update rm in MmGA with x*
10:  until fb = Maximum feedback cycles or pf_fb = pf_true is met
11:  return pf
```

Fig. 3 The pseudo-code of the MmGA ensemble for tackling the job-shop scheduling.

Let a be the sequence of n jobs, and a set of n jobs is represented by $\mathbf{x}_i{}' = \{a_1,\ldots a_n\}$ (line 3, Figure 3). As such, the length of the MmGA chromosome is n. In essence, each chromosome represents the decision vector of a sequence of n jobs. As an example, let consider $n = 5$, each chromosome allele is given a probability value from 0 to 1, e.g. $\mathbf{x}_i{}' = \{0.1,0.5,0.3,0.9,0.8\}$. Based on the probabilities values ranging from the smaller to the larger, the sequence of n jobs is {1,3,2,5,4}. Table 6 shows all possible combinations of paired jobs for n = 5 to 8. For n = 5, there are three possible combinations of paired jobs. Note that the search space of $n = 5$ is $(n! \times 3) = 360$, while that of $n = 8$ is $(n! \times 6) = 241{,}920$. It is obvious that the search space grows rapidly with respect to increasing number of jobs. Therefore, the challenge of obtaining the optimal solutions, as fulfilled the Definition 2, increases tremendously even for a moderate job size of 8.

For the 5-job problem, the job processing schedule, cost-saving matrix, and the associated optimal results from the enumeration method are shown in Tables 2, 4, and 5, respectively. To comprehensively evaluate the effectiveness of the MmGA ensemble model for this job-shop scheduling problem, four experimental configurations with 5 to 8 jobs have been used for evaluation. The software program has been developed using a combination of Java and C programming languages. The programs have been executed using the Intel Xeon-based server with 2.5GHz and 8GB RAM in terms of speed and memory, respectively.

5 Results and Discussion

We conducted two experimental studies to evaluate the usefulness of the single and ensemble MmGA models. The first experiment was concerned with a two-objective job-shop scheduling problem, i.e., minimising tardiness and maximising cost saving. The second experiment included another objective, i.e., maximising earliness, making it a more complex three-objective job-shop scheduling problem. The optimal results from the enumeration method served as the baseline performance for comparison with the results from the MmGA models. To quantify the performance of the MmGA models statistically, the bootstrap method [21] was employed to compute the average results from 30 runs with 100,000 re-samplings as well as the associated 95% confidence intervals. Note that bootstrap is a useful method to determine the population parameters based on a small sample size, and its effectiveness has been shown in different domains including medicine [30], [35], signal processing [70], and biometrics [60]. The unique non-dominated solutions and the search space associated with n jobs are shown in Table 7.

Table 6 Possible pair variations of *n* jobs

Size	Pair variation	Total
5	(job1-job2),(job3-job4),job5; job1,(job2-job3),(job4-job5); (job1-job2),job3,(job4-job5);	3
6	(job1-job2),(job3-job4),(job5-job6); (job1-job2),job3,(job4-job5),job6; job1,(job2-job3),(job4-job5),job6; job1,(job2-job3),job4,(job5-job6);	4
7	job1,(job2-job3),(job4-job5),(job6-job7); job1,(job2-job3),job4,(job5-job6),job7; (job1-job2),(job3-job4),(job5-job6),job7; (job1-job2),job3,(job4-job5),(job6-job7); (job1-job2),(job3-job4),job5,(job6-job7);	5
8	(job1-job2),(job3-job4),(job5-job6),(job7-job8); (job1-job2),job3,(job4-job5),(job6-job7),job8; (job1-job2),job3,(job4-job5),job6,(job7-job8); (job1-job2),(job3-job4),job5,(job6-job7),job8; (job1-job2),(job3-job4),job5,job6,(job7-job8); job1,(job2-job3),(job4-job5),(job6-job7),job8; job1,(job2-job3),(job4-job5),job6,(job7-job8); job1,(job2-job3),job4,(job5-job6),(job7-job8);	8

Table 7 The job pair pattern for the job-shop scheduling

Jobs	Total search space size
5	5! x 3 = 360
6	6! x 4 = 2,880
7	7! x 4 = 20,160
8	8! x 6 = 241,920

5.1 Two-objective Experiment

For this two-objective experiment, the average results in minimising tardiness and maximising cost saving are shown in Table 8. It can be observed that the MmGA ensemble model yielded a comparable performance as those of the enumeration method and the MmGA model. This could be justified by the 95% confidence intervals of the average results. Specifically, the lower and upper bounds of the 95% confidence intervals of the results of the MmGA ensemble contained the best and worst results from the enumeration method as well as the MmGA model for tardiness and cost-saving. In other words, there was no statistical difference in performance between the enumeration method and the proposed MmGA models for job size *n* from 5 to 8. The computational cost of the MmGA ensemble was also lower than those from the published model in [57] for all problems.

5.2 Three-Objective Experiment

For this three-objective problem, the same experiment with 30 runs was conducted. An additional objective of maximising earliness was included, on top of the two objectives in the previous experiment. There are a few key observations. In addition to producing the optimal (non-dominated) results for different job sizes, the MmGA ensemble model was capable of yielding the unique solutions existed in the optimal results, as shown in Table 9. As an example, out of a total of 1019 solutions produced by the MmGA ensemble for the 5-job problem, 219 non-dominated solutions were found. Among these non-dominated solutions, there were five unique solutions. The MmGA ensemble was able to identify all these five unique solutions. This constituted a useful feature in practical environments, because the decision makers could have the option to choose one of the unique solutions to meet all the required objectives under different conditions.

Table 8 A comparison of *S*, *T*, and *E* for the enumeration method, and the single and ensemble MmGA models for job-shop scheduling

	Enumeration Method	**MmGA [57]**			**Proposed MmGA ensemble**		
	(worst to best) [57]	Lower Bound	Mean	Upper Bound	Lower Bound	Mean	Upper Bound
5 Jobs							
S (dollars)	4.50 to 5.16	4.97	5.03	5.08	5.16	5.16	5.16
T (days)	5.00 to 0.00	0.87	0.85	0.83	0.17	0.07	0.00
E (days)	-	-	-	-	20.33	20.03	19.67
Time (ms)	> 1000	2.8	3.23	4.23	1.47	1.73	2.37
6 Jobs							
S (dollars)	11.09 to 14.87	11.92	12.26	12.67	14.57	14.72	14.80
T (days)	21.00 to 12.00	15.08	14.30	13.77	14.40	13.67	13.07
E (days)	-	-	-	-	14.00	13.7	13.27
Time (ms)	> 1000	10.37	12.93	17.37	1.87	2.13	2.83
7 Jobs							
S (dollars)	13.68 to 14.49	13.93	14.00	14.08	14.27	14.36	14.43
T (days)	30.00 to 17.00	23.67	22.49	21.46	22.23	21.33	20.60
E (days)	-	-	-	-	16.67	15.67	14.60
Time (ms)	> 1000	4.23	4.53	5.17	1.87	2.17	2.87
8 Jobs							
S (dollars)	17.50 to 19.06	17.74	17.86	18.01	18.35	18.5	18.67
T (days)	31.00 to 20.00	26.62	25.90	25.26	26.93	26.33	25.73
E (days)	-	-	-	-	16.40	15.93	15.37
Time (ms)	> 1000	4.87	5.17	5.43	2.10	2.50	3.80

Table 9 The non-dominated solutions of the MmGA Ensemble

Jobs	**No. of non-dominated solutions found**	**No. of unique non-dominated Solutions**
5	219/1019	5
6	39/1111	8
7	10/1039	4
8	3/913	3

It is worth noting that the enumeration method consumed more than a second to produce the optimised results, as compared with those of the MmGA-based models, i.e., from 2.8ms to 5.43ms for single MmGA and from 1.43ms to 3.8ms for the MmGA ensemble, using the same hardware and server configurations. Comparing the single and the ensemble MmGA models, the latter is statistically more efficient from the computational time perspective in tackling MOP-based job-shop scheduling. This is supported by the 95% confidence levels of the mean computational time, in which the upper bounds of the MmGA ensemble are better than the lower bounds of the single MmGA for all computational durations summarised in Table 8.

6 Conclusions

The application of an MOEA model, i.e. the MmGA ensemble, to real-world multi-objective job-shop scheduling problems has been demonstrated. The details of the scenarios are based on a manufacturing company in Australia. All jobs have to be scheduled on a single machine with one job at a time. The MmGA ensemble has been deployed to optimize the processing orders of 5 to 8 jobs. The MmGA ensemble model is able to produce statistically equivalent performance as compared with those from the enumeration method, as ascertained by the 95% confidence intervals of the average results. In addition, more than one unique solutions could be solicited from the MmGA ensemble model; therefore allowing the decision makers to have options in selecting the most appropriate solution that is able to meet all the objectives under different conditions.

Although positive results have been obtained, it is necessary to further evaluate the effectiveness of the MmGA ensemble model with multiple MOP indicators, e.g. the inverted generational distance and spread [20]. In addition, an agent-based modelling method [14], [3] can be used to formulate the elite-selection scheme and for tracking the behaviour of individual MmGA models. Besides that, the applicability of the proposed model to undertaking other MOPs can be evaluated. All these constitute the direction for further work.

Acknowledgements This research is supported by Collaborative Research in Engineering, Science & Technology (CREST) Grant P05C2-14.

References

1. Abu-Lebdeh, G., Benekohal, R.F.: Convergence variability and population sizing in micro-genetic algorithms. Computer-Aided Civil and Infrastructure Engineering **14**(5), 321–334 (1999)

2. Amirian, H., Sahraeian, R.: Augmented ϵ-constraint method in multi-objective flowshop problem with past sequence set-up times and a modified learning effect. International Journal of Production Research **53**(19), 5962-5976 (2015)

3. Archimede, B., Letouzey, A., Memon, M., Xu, J.: Towards a distributed multi-agent framework for shared resources scheduling. Journal of Intelligent Manufacturing **25**(5), 1077–1087 (2014)

4. Bäck, T., Hammel, U., Schwefel, H.P.: Evolutionary computation: comments on the history and current state. IEEE Transactions on Evolutionary Computation **1**(1), 3–17 (1997)

5. Bäck, T., Schwefel, H.P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation **1**(1), 1–23 (1993)

6. Besnard, J., Ruda, G.F., Setola, V., Abecassis, K., Rodriguiz, R.M., Huang, X.P., Norval, S., Sassano, M.F., Shin, A.I., Webster, L.A., Simeons, F.R.C., Stojanovski, L., Prat, A., Seidah, N.G., Constam, D.B., Bickerton, G.R., Read, K.D., Wetsel, W.C., Gilbert, I.H., Roth, B.L., Hopkins, A.L.: Automated design of ligands to polypharmacological profiles. Nature **492**(7428), 215–220 (2012)

7. Capón-García, E., Bojarski, A.D., Espuña, A., Puigjaner, L.: Multiobjective evolutionary optimization of batch process scheduling under environmental and economic concerns. AIChE Journal **59**(2), 429–444 (2013)

8. Çaliş, B., Bulkan, S.: A research survey: review of ai solution strategies of job shop scheduling problem. Journal of Intelligent Manufacturing pp. 1–13 (2013)

9. Chakaravarthy, G., Marimuthu, S., Ponnambalam, S., Kanagaraj, G.: Improved sheep flock heredity algorithm and artificial bee colony algorithm for scheduling m-machine flow shops lot streaming with equal size sub-lot problems. International Journal of Production Research **52**(5), 1509–1527 (2014)

10. Chen, S.H., Chang, P.C., Cheng, T., Zhang, Q.: A self-guided genetic algorithm for permutation flowshop scheduling problems. Computers & Operations Research **39**(7), 1450–1457 (2012)

11. Chen, S.H., Chen, M.C.: Addressing the advantages of using ensemble probabilistic models in estimation of distribution algorithms for scheduling problems. International Journal of Production Economics **141**(1), 24–33 (2013)

12. Chen, Y.: Fuzzy skyhook surface control using micro-genetic algorithm for vehicle suspension ride comfort. In: M. Kppen, G. Schaefer, A. Abraham (eds.) Intelligent Computational Optimization in Engineering, *Studies in Computational Intelligence*, vol. 366, pp. 357–394. Springer Berlin / Heidelberg (2011)

13. Chiang, T.C., Lin, H.J.: A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling. International Journal of Production Economics **141**(1), 87–98 (2013)

14. Chou, Y.C., Cao, H., Cheng, H.H.: A bio-inspired mobile agent-based integrated system for flexible autonomic job shop scheduling. Journal of Manufacturing Systems **32**(4), 752 – 763 (2013)

15. Coello, C.A.C.: Evolutionary multi-objective optimization: a historical view of the field. IEEE Computational Intelligence Magazine **1**(1), 28–36 (2006)

16. Coello, C.A.C., Lamont, G.B., Van Veldhuisen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Genetic and evolutionary computation series. Springer London, Limited (2007)

17. Coello, C.A.C., Pulido, G.: Multiobjective structural optimization using a microgenetic algorithm. Structural and Multidisciplinary Optimization **30**, 388–403 (2005)

18. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)

19. Deng, W., Li, W., Cai, X., Bulou, A., Wang, Q.A.: Universal scaling in sports ranking. New Journal of Physics **14**(9), 093,038 (2012)

20. Durillo, J.J., Nebro, A.J.: jMetal: A java framework for multi-objective optimization. Advances in Engineering Software **42**(10), 760–771 (2011)

21. Efron, B.: The Jackknife, the Bootstrap and Other Resampling Plans. Cambridge University Press (1982)

22. Gao, K.Z., Suganthan, P.N., Pan, Q.K., Chua, T.J., Cai, T.X., Chong, C.S.: Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives. Journal of Intelligent Manufacturing **27**(2), 363–374 (2014)

23. Gen, M., Lin, L.: Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. Journal of Intelligent Manufacturing **25**(5), 849–866 (2014)

24. Geyik, F., Cedimoglu, I.: The strategies and parameters of tabu search for job-shop scheduling. Journal of Intelligent Manufacturing **15**(4), 439–448 (2004)

25. Gneiting, T., Raftery, A.E.: Weather forecasting with ensemble methods. Science **310**(5746), 248–249 (2005)

26. Goldberg, D.E.: Sizing populations for serial and parallel genetic algorithms. In: Proceedings of the third international conference on Genetic algorithms, pp. 70–79. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)

27. Hanoun, S., Nahavandi, S.: A greedy heuristic and simulated annealing approach for a bicriteria flowshop scheduling problem with precedence constraints - a practical manufacturing case. The International Journal of Advanced Manufacturing Technology **60**(9-12), 1087–1098 (2012)

28. Hanoun, S., Nahavandi, S., Creighton, D., Kull, H.: Solving a multiobjective job shop scheduling problem using pareto archived cuckoo search. In: IEEE 17th Conference on Emerging Technologies Factory Automation, 2012 (ETFA 2012), pp. 1–8 (2012)

29. Hanoun, S., Nahavandi, S., Kull, H.: Pareto archived simulated annealing for single machine job shop scheduling with multiple objectives. In: The Sixth International Multi-Conference on Computing in the Global Information Technology (ICCGI 2011), pp. 99–104. Luxembourg (2011)

30. Inker, L.A., Schmid, C.H., Tighiouart, H., Eckfeldt, J.H., Feldman, H.I., Greene, T., Kusek, J.W., Manzi, J., Van Lente, F., Zhang, Y.L., Coresh, J., Levey, A.S.: Estimating glomerular filtration rate from serum creatinine and cystatin c. New England Journal of Medicine **367**(1), 20–29 (2012)

31. Karthikeyan, S., Asokan, P., Nickolas, S.: A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints. The International Journal of Advanced Manufacturing Technology **72**(9-12), 1567–1579 (2014)

32. Khalili, M., Naderi, B.: A bi-objective imperialist competitive algorithm for no-wait flexible flow lines with sequence dependent setup times. The International Journal of Advanced Manufacturing Technology **76**(1-4), 461–469 (2015)

33. Li, D., Das, S., Pahwa, A., Deb, K.: A multi-objective evolutionary approach for generator scheduling. Expert Systems with Applications **40**(18), 7647–7655 (2013)

34. Lounkine, E., Keiser, M.J., Whitebread, S., Mikhailov, D., Hamon, J., Jenkins, J.L., Lavan, P., Weber, E., Doak, A.K., Cote, S., Shoichet, B.K., Urban, L.: Large-scale prediction and testing of drug activity on side-effect targets. Nature **486**(7403), 361–367 (2012)

35. McWilliams, A., Tammemagi, M.C., Mayo, J.R., Roberts, H., Liu, G., Soghrati, K., Yasufuku, K., Martel, S., Laberge, F., Gingras, M., Atkar-Khattra, S., Berg, C.D., Evans, K., Finley, R., Yee, J., English, J., Nasute, P., Goffin, J., Puksa, S., Stewart, L., Tsai, S., Johnston, M.R., Manos, D., Nicholas, G., Goss, G.D., Seely, J.M., Amjadi, K., Tremblay, A., Burrowes, P., MacEachern, P., Bhatia, R., Tsao, M.S., Lam, S.: Probability of cancer in pulmonary nodules detected on first screening CT. The New England journal of medicine **369**(10), 910–919 (2013)

36. Mendoza, J., Lopez, M., Coello, C.A.C., Lopez, E.: Microgenetic multiobjective reconfiguration algorithm considering power losses and reliability indices for medium voltage distribution network. IET Generation, Transmission & Distribution **3**(9), 825–840 (2009)

37. Mendoza, J., Morales, D., Lopez, R., Lopez, E., Vannier, J.C., Coello, C.A.C.: Multiobjective location of automatic voltage regulators in a radial distribution network using a micro genetic algorithm. IEEE Transactions on Power Systems **22**(1), 404–412 (2007)

38. Nagar, A., Haddock, J., Heragu, S.: Multiple and bicriteria scheduling: A literature survey. European Journal of Operational Research **81**(1), 88–104 (1995)

39. Nebro, A., Alba, E., Luna, F.: Multi-objective optimization using grid computing. Soft Computing **11**(6), 531–540 (2007)

40. Nebro, A.J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J.J., Beham, A.: AbYSS: Adapting scatter search to multiobjective optimization. IEEE Transactions on Evolutionary Computation **12**(4), 439–457 (2008)

41. Nguyen, S., Zhang, M., Johnston, M., Tan, K.C.: A computational study of representations in genetic programming to evolve dispatching rules for the job shop scheduling problem. IEEE Transactions on Evolutionary Computation **17**(5), 621–639 (2013)

42. Panneerselvam, R.: Simple heuristic to minimize total tardiness in a single machine scheduling problem. The International Journal of Advanced Manufacturing Technology **30**(7-8) (2006)

43. Pareto, V., Schwier, A.S.: Manual of political economy. Augustus M. Kelley, Publishers (1971)

44. Prot, D., Bellenguez-Morineau, O., Lahlou, C.: New complexity results for parallel identical machine scheduling problems with preemption, release dates and regular criteria. European Journal of Operational Research **231**(2), 282–287 (2013)

45. Qiu, X., Lau, H.Y.: An ais-based hybrid algorithm for static job shop scheduling problem. Journal of Intelligent Manufacturing **25**(3), 489–503 (2014)

46. Rohaninejad, M., Kheirkhah, A., Fattahi, P., Vahedi-Nouri, B.: A hybrid multi-objective genetic algorithm based on the electre method for a capacitated flexible job shop scheduling problem. The International Journal of Advanced Manufacturing Technology **77**(1-4), 51–66 (2015)

47. Sapp, B., Weiss, D., Taskar, B.: Parsing human motion with stretchable models. In: IEEE Conference on Computer Vision and Pattern Recognition, 2011 (CVPR 2011), pp. 1281–1288. Providence, RI (2011)

48. Schuetz, R., Zamboni, N., Zampieri, M., Heinemann, M., Sauer, U.: Multidimensional optimality of microbial metabolism. Science **336**(6081), 601–604 (2012)

49. Sen, T., Raiszadeh, F.M.E., Dileepan, P.: A branch-and-bound approach to the bicriterion scheduling problem involving total flowtime and range of lateness. Management Science **34**(2), 254–260 (1988)

50. Shahsavari-Pour, N., Ghasemishabankareh, B.: A novel hybrid meta-heuristic algorithm for solving multi objective flexible job shop scheduling. Journal of Manufacturing Systems **32**(4), 771 – 780 (2013)

51. Shoval, O., Sheftel, H., Shinar, G., Hart, Y., Ramote, O., Mayo, A., Dekel, E., Kavanagh, K., , Alon, U.: Evolutionary trade-offs, pareto optimality, and the geometry of phenotype space. Science **336**(6085), 1157–1160 (2012)

52. Sridhar, J., Rajendran, C.: Scheduling in flowshop and cellular manufacturing systems with multiple objectives - a genetic algorithmic approach. Production Planning & Control **7**(4), 374–382 (1996)

53. Srinivas, N., Deb, K.: Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation **2**, 221–248 (1994)

54. Tan, C.J., Lim, C.P., Cheah, Y.N.: A modified micro genetic algorithm for undertaking multi-objective optimization problems. Journal of Intelligent and Fuzzy Systems **24**(3), 483–495 (2013)

55. Tan, C.J., Lim, C.P., Cheah, Y.N.: A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models. Neurocomputing **125**, 217–228 (2014)

56. Tan, C.J., Lim, C.P., Cheah, Y.N., Tan, S.C.: Classification and optimization of product review information using soft computing models. In: International Symposium on Affective Engineering, 2013 (ISAE 2013), pp. 115–120. Japan Society of Kansei Engineering, Kitakyushu, Japan (2013)

57. Tan, C.J., Samer, H., Lim, C.P., Creighton, D., Nahavandi, S.: A multi-objective evolutionary algorithm-based decision support system: A case study on job-shop scheduling in manufacturing. In: 2015 9th Annual IEEE International Systems Conference (SysCon), pp. 170–174 (2015)

58. Tian, J., Li, M., Chen, F.: Dual-population based coevolutionary algorithm for designing RBFNN with feature selection. Expert Systems with Applications **37**(10), 6904–6918 (2010)

59. Tiwari, A., Chang, P.C., Tiwari, M., Kollanoor, N.J.: A pareto block-based estimation and distribution algorithm for multi-objective permutation flow shop scheduling problem. International Journal of Production Research **53**(3), 793–834 (2015)

60. Tsai, M.Y.: Variable selection in bayesian generalized linear-mixed models: An illustration using candidate gene case-control association studies. Biometrical Journal **57**(2), 234–253 (2015)

61. Wassenhove, L.N.V., Gelders, L.F.: Solving a bicriterion scheduling problem. European Journal of Operational Research **4**(1), 42–48 (1980)

62. Wassenhove, L.V., Gelders, L.: Four solution techniques for a general one machine scheduling problem: A comparative study. European Journal of Operational Research **2**(4), 281–290 (1978)

63. Wilson, D.T., Hawe, G.I., Coates, G., Crouch, R.S.: A multi-objective combinatorial model of casualty processing in major incident response. European Journal of Operational Research **230**(3), 643–655 (2013)

64. Wu, W.H., Wu, W.H., Chen, J.C., Lin, W.C., Wu, J., Wu, C.C.: A heuristic-based genetic algorithm for the two-machine flowshop scheduling with learning consideration. Journal of Manufacturing Systems **35**, 223 – 233 (2015)

65. Xiong, J., ning Xing, L., wu Chen, Y.: Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. International Journal of Production Economics **141**(1), 112–126 (2013)

66. Zhang, G., Shao, X., Li, P., Gao, L.: An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. Computers and Industrial Engineering **56**(4), 1309–1318 (2009)

67. Zhang, J., Yang, J., Zhou, Y.: Robust scheduling for multi-objective flexible job-shop problems with flexible workdays. Engineering Optimization **0**(0), 1–17 (2016). DOI 10.1080/0305215X.2016.1145216

68. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by MOEA/D with gaussian process model. IEEE Transactions on Evolutionary Computation **14**(3), 456–474 (2010)

69. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation **7**(2), 117–132 (2003)

70. Zoubir, A., Boashash, B.: The bootstrap and its application in signal processing. IEEE Signal Processing Magazine 15(1), 56–76 (1998)

71. A.R. Yildiz, N. Öztürk, N. Kaya, F. Öztürk: Integrated optimal topology design and shape optimization using neural networks. Structural and Multidisciplinary Optimization 4(25), 251-260 (2003)
72. Nursel Öztürk, Ali R. Yıldız, Necmettin Kaya, Ferruh Öztürk: Neuro-Genetic Design Optimization Framework to Support the Integrated Robust Design Optimization Process in CE. Concurrent Engineering 14(1), 5-16 (2006)
73. Yildiz, Ali R.: Comparison of Evolutionary-based Optimization Algorithms for Structural Design

Optimization. Engineering Applications of Artificial Intelligence 26(1), 327-333 (2013)
74. Morteza Kiani, Ali R. Yildiz: A Comparative Study of Non-traditional Methods for Vehicle Crashworthiness and NVH Optimization. Archives of Computational Methods in Engineering, 1-12 (2015)
75. Ali Rıza Yildiz, Necmettin Kaya, Ferruh Ozturk, Orhan Alankus: Optimal design of vehicle components using topology design and optimisation. International Journal of Vehicle Design, 34(4), 387–398 (2004)
76. Ali Riza Yildiz: Optimal Structural Design of Vehicle Components Using Topology Design and Optimization. Materials Testing, 50(4), 224-228 (2008)
77. Ali Rıza Yıldız, Enes Kurtuluş, Emre Demirci, Betul Sultan Yıldız, and Selçuk Karagöz: Optimization of thin-wall structures using hybrid gravitational search and Nelder-Mead algorithm. Materials Testing: 58(1), 75-78 (2016)
78. Betul Sultan Yildiz, Huseyin Lekesiz, and Ali Riza Yildiz: Structural design of vehicle components using gravitational search and charged system search algorithms. Materials Testing, 58(1), 79-81 (2016)
79. Hakan Gökdağ and Ali R. Yildiz: Structural Damage Detection Using Modal Parameters and Particle Swarm Optimization. Materials Testing 54(6), 416-420 (2012)