

Efficient Heuristics for Flowshop Scheduling for Minimizing the Makespan and Total Flowtime of Jobs

Yasuhiro Hirakawa[†]

Department of Industrial Administration, Tokyo University of Science
2641 Yamazaki, Noda City, Chiba, 278-8510, Japan
E-mail: hirakawa@rs.noda.tus.ac.jp

Aya Ishigaki

Department of Industrial Administration, Tokyo University of Science
2641 Yamazaki, Noda City, Chiba, 278-8510, Japan
E-mail: ishigaki@rs.noda.tus.ac.jp

Received, January 15, 2011; Revised, April 18, 2011; Accepted, April 22, 2011

Abstract. The problem of scheduling in permutation flowshops has been extensively investigated by many researchers. Recently, attempts are being made to consider more than one objective simultaneously and develop algorithms to obtain a set of Pareto-optimal solutions. Varadharajan *et al.* (2005) presented a multi-objective simulated-annealing algorithm (MOSA) for the problem of permutation-flowshop scheduling with the objectives of minimizing the makespan and the total flowtime of jobs. The MOSA uses two initial sequences obtained using heuristics, and seeks to obtain non-dominated solutions through the implementation of a probability function, which probabilistically selects the objective of minimizing either the makespan or the total flowtime of jobs. In this paper, the same problem of heuristically developing non-dominated sequences is considered. We propose an effective heuristics based on simulated annealing (SA), in which the weighted sum of the makespan and the total flowtime is used. The essences of the heuristics are in selecting the initial sequence, setting the weight and generating a solution in the search process. Using a benchmark problem provided by Taillard (1993), which was used in the MOSA, these conditions are extracted in a large-scale experiment. The non-dominated sets obtained from the existing algorithms and the proposed heuristics are compared. It was found that the proposed heuristics drastically improved the performance of finding the non-dominated frontier.

Keywords: Flowshop scheduling, Makespan, Flowtime, Multi-objectives, Simulated annealing

1. INTRODUCTION

In production scheduling, it is often required to simultaneously optimize two or more objectives such as the makespan and the total flowtime; however, in general, a solution that optimizes two or more objectives simultaneously does not exist. Consequently, attempts are being made to consider more than one objective simultaneously and develop algorithms to obtain a set of Pareto-optimal solutions. Ishibuchi and Murata (1998) transformed a multi-objective production scheduling problem into a single-objective problem by using a weighted sum of multi-objective functions, and proposed a multi-objective genetic algorithm for setting up

the weight at random in the search process. It is shown that the proposed method can calculate the Pareto-optimal solutions. On the other hand, Ulungu *et al.* (1999) proposed a two-dimensional simulated-annealing (SA) algorithm that controls the search direction by selecting temperatures independently for two objective functions. Moreover, Varadharajan *et al.* (2005) proposed the multi-objective SA method (MOSA), which chooses one objective function randomly in a search process, and then, controls the search direction by this probability. It was shown that the performance of the proposed heuristic algorithm is higher than the conventional methods.

The term ‘simulated annealing’ appears in literature

[†] : Corresponding Author

by Kirkpatrick *et al.* (1983); however, the algorithm was developed by Metropolis *et al.* (1953). In the SA algorithm, a current solution may be replaced with one of the neighbourhoods, even if it is not a better solution. The probability of accepting an inferior solution depends on the two solutions and a temperature parameter. The temperature is effective for escaping from a local optimum and is gradually decreased in the search process; however, if the temperature is decreased too much, it loses its function. Therefore, the temperature setting has to be controlled very carefully.

As a result of the experiments, we propose a modified SA method - a new algorithm in which the temperature is not decreased in the search process.

The purpose of this research is to develop a highly efficient search algorithm in order to obtain a set of Pareto-optimal solutions for the permutation-flowshop scheduling problem with the objectives of minimizing the makespan and the total flowtime of jobs. The research is based on the SA method using 1) the neighbourhood structure, 2) the initial solution, 3) weight and temperature, and 4) the number of iterations and replications analysed using a benchmark problem of size (20 jobs \times 20 machines) by Taillard (1993).

2. TWO-OBJECTIVE PERMUTATION-FLOWSHOP SCHEDULING PROBLEM

The problem of permutation-flowshop scheduling with the objectives of minimizing the makespan and the total flowtime of jobs is considered. The problem that develops efficient sequences $\mathbf{x} = (x_1, x_2, \dots, x_n)$ of n jobs is formulated as a combinatorial optimization problem. For generality, we denote the makespan and the total flowtime of jobs yielded by a sequence \mathbf{x} by $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, respectively.

2.1 Pareto-Optimal Solution

Sequence \mathbf{x} is said to dominate sequence \mathbf{x}^* if

$$f_s(\mathbf{x}) \leq f_s(\mathbf{x}^*) \quad \text{for all } s = 1, 2$$

and

$$f_s(\mathbf{x}) < f_s(\mathbf{x}^*) \quad \text{for at least one } s.$$

This research aims to find all non-dominated solutions (Pareto-optimal solutions) of the two-objective optimization problems.

2.2 Fitness Function

When applying the SA method to a two-objective optimization problem, we employ the following weighted sum of the two objectives: makespan (f_1) and total flowtime (f_2), as a fitness function of the sequence \mathbf{x} :

$$F(\mathbf{x}) = w \frac{f_1(\mathbf{x})}{f_1(\mathbf{x}_0)} + (1-w) \frac{f_2(\mathbf{x})}{f_2(\mathbf{x}_0)} \quad (1)$$

where w is the non-negative constant weight for the makespan and \mathbf{x}_0 is the initial sequence. The feature here is that each objective function is normalized by the initial value.

3. BASICS OF SA IMPLEMENTED IN THE CURRENT STUDY

SA is a sophisticated random- optimization method and tackles the problem of getting trapped in a local optimum by tolerating deteriorating moves with small but positive probability.

The basic SA algorithm employed in this paper is as follows:

Step 1: Let \mathbf{x} be the current sequence and \mathbf{x}^* be an alternative candidate sequence.

If \mathbf{x}^* is at least as good as \mathbf{x} with respect to the fitness function defined by Eq. (1): $F(\mathbf{x}^*) \leq F(\mathbf{x})$, the candidate sequence \mathbf{x}^* is always preferred to \mathbf{x} .

Otherwise, if $F(\mathbf{x}^*) > F(\mathbf{x})$, \mathbf{x}^* is accepted only with a probability determined by the following function

$$\exp\left\{\frac{F(\mathbf{x}) - F(\mathbf{x}^*)}{c}\right\} \quad (2)$$

The temperature parameter (c) is fixed throughout a search process.

Step 2: If \mathbf{x}^* is selected, set \mathbf{x}^* to \mathbf{x} .

Step 3: Select a schedule \mathbf{x}^* from the neighbourhood $N(\mathbf{x})$ of \mathbf{x} , which is defined below.

Step 4: Go back to *step 2* until the terminal condition is satisfied.

The following three operations are well-known methods for generating a neighbourhood $N(\mathbf{x})$ of a sequence \mathbf{x} . In each of the following operations, two positions of the current sequence \mathbf{x} are randomly chosen.

(1) Exchange: Two elements x_i and x_j are exchanged.

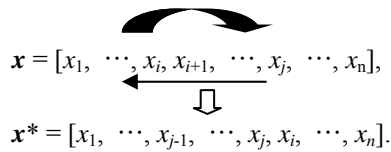
$$\begin{array}{c} \mathbf{x} = [x_1, \dots, x_i, \dots, x_j, \dots, x_n], \\ \quad \quad \quad \swarrow \quad \searrow \\ \mathbf{x}^* = [x_1, \dots, x_j, \dots, x_i, \dots, x_n]. \end{array}$$

(2) Inversion: The order of the elements between positions i and j is reversed.

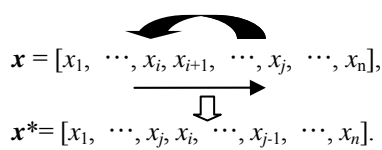
$$\begin{array}{c} \mathbf{x} = [x_1, \dots, x_i, x_{i+1}, \dots, x_{j-1}, x_j, \dots, x_n], \\ \quad \quad \quad \longleftarrow \quad \longrightarrow \\ \mathbf{x}^* = [x_1, \dots, x_j, x_{j-1}, \dots, x_{i+1}, x_i, \dots, x_n]. \end{array}$$

(3) Shift: The order of the elements between positions i and j is shifted by one to the left or the right in a rotational manner as follows.

(3-1) Left shift



(3-2) Right shift



4. SET OF PARETO-OPTIMAL SOLUTIONS OBTAINED FOR PROBLEM 1 OF SIZE (20 × 20) GIVEN BY TAILLARD (1993)

The Pareto-optimal solutions for problem 1 of size (20 × 20) given by Taillard (1993), which were obtained by an exact enumeration method, are shown in Figure 1. The number of the Pareto-optimal solutions is 32.

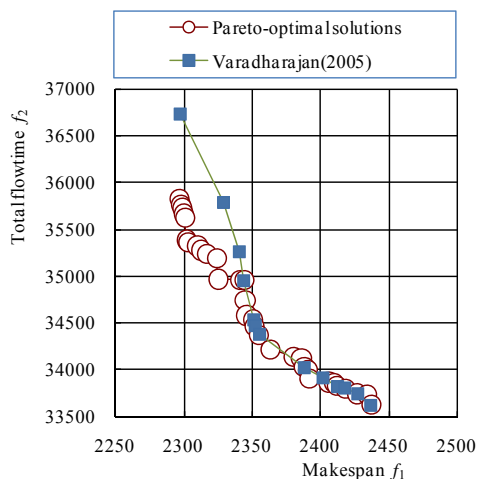


Figure 1. Pareto-optimal solutions for problem 1 of size (20 × 20) given by Taillard (1993).

Figure 1 also shows the result of Varadharajan *et al.* (2005). They show that the net set of non-dominated solutions from various multi-objective flowshop-scheduling algorithms included their proposed algorithm (MOSA); however, only 9 of the 14 solutions in the net set are the Pareto-optimal solutions.

5. EFFECT OF THE NEIGHBOURHOOD STRUCTURES

The search efficiency of the neighbourhood structures in the SA method is analysed with problem 1 of

size (20 × 20) given by Taillard (1993).

5.1 Combined Shift Operations

As the result of the experiments, the shift operation was the most effective in the permutation-flowshop scheduling problems. In this paper, we propose two combined shift operations, mixed shift and best shift, which combine the right and left shifts as follows:

- (1) Mixed-shift operation: For two positions i and j selected in the shift operation, if $i < j$, left shift is taken; otherwise, right shift is taken.
- (2) Best-shift operation: Right-shift and left-shift operations are performed, two generated sequences are evaluated and the best one is selected.

5.2 Numerical Experiments

Numerical experiments are performed for different temperatures and several iterations. The results of replications with different random seeds are shown in Figure.

In the experiments, we used the following parameter specifications, and the influence of these parameters is shown later.

Initial sequence: Job number
 Weight: $w = 0.5$
 Replication: $R = 100$

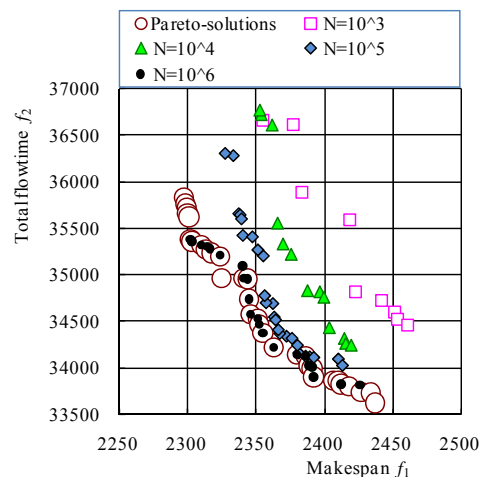


Figure 2. Pareto-optimal solutions for problem 1 of size (20 × 20) given by Taillard (1993) using the neighbourhood structure of the inversion operation.

Figure 2 shows the effect of the number of iterations N as an obtained Pareto-optimal frontier. A numerical experiment is performed at $c = 0.0012$ and the neighbourhood structure of the inversion operation. Using the neighbourhood structure of the inversion operation, only one Pareto-optimal solution is found at $N = 100,000$, but 16 Pareto-optimal solutions are found at N

= 1,000,000.

The effect of the neighbourhood structures are summarized in Figures 3 and 4.

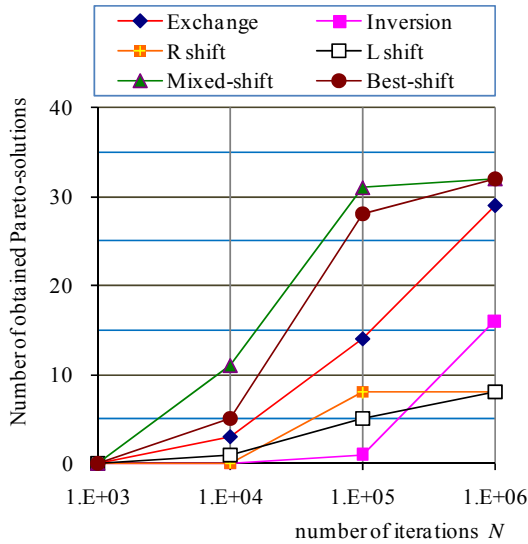


Figure 3. Number of obtained Pareto-optimal solutions and number of iterations ($c = 0.0012$).

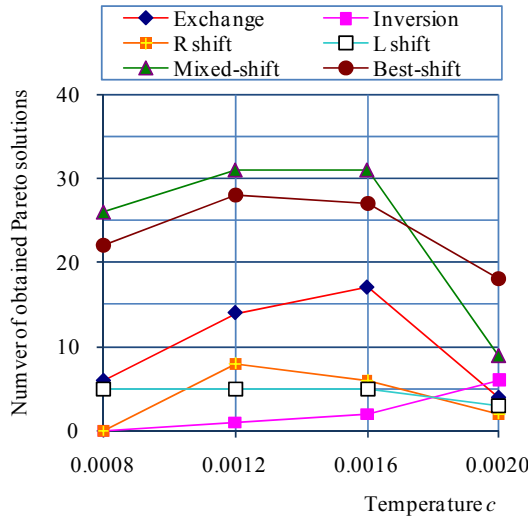


Figure 4. Number of obtained Pareto-optimal solutions and temperature ($N = 100,000$).

Figure 3 shows the relationship between the number of the obtained Pareto-optimal solutions and the number of iterations N for each neighbourhood structure: exchange, inversion, right shift, left shift, mixed shift and best shift. Figure 4 shows the relationship between the number of the obtained Pareto-optimal solutions and the temperature c for each neighbourhood structure.

Figures 3 and 4 show that the mixed-shift and best-shift operations outperformed the other neighbourhood structures.

6. EFFECT OF THE INITIAL SEQUENCE AND WEIGHT SETTING

In the MOSA, Varadharajan *et al.* (2005) use two initial sequences, with one being generated with the objective of minimizing the makespan and the other with the objective of minimizing the total flowtime of jobs, which are obtained using the existing heuristics. The MOSA uses a probability function to select probabilistically either the objective of minimizing the make-span or the objective of minimizing the total flowtime of jobs.

6.1 Setting the Initial Sequence

To analyse the effect of the initial sequences, the following two settings were used:

- (a) The sequence of job number.
- (b) Pareto-optimal sequence x_0 with $f_1(x_0) = 2297$, $f_2(x_0) = 35831$ for problem 1 of size (20×20) minimizing $f_1(x)$.

6.2 Setting the Weight in the SA Search

As seen in Figures 3 and 4, for the fixed weight $w = 0.5$ and replication $R = 100$, almost all Pareto-optimal solutions are found. We employ a similar procedure with phase 2 of the MOSA. After carrying out the SA search for the pre-determined number of iterations, we repeat the SA search process with a different w setting. If the number of replications is R , we set $w = (r - 0.5)/R$ for the r th SA search.

6.3 Numerical Experiments

Anumerical experiment is performed at $c = 0.0012$ and the neighborhood structure of the mixed-shift operation. Figure 5 and 6 show the rate of finding the Pareto-optimal solutions for each of the w setting. Figure 5(a) and 5(b) show the effect of two types of the initial sequences used in the SA algorithm. Figure 5 show that the sequence of job number taken as the initial solution in the SA search gives a performance better than the sequence nearer to the Pareto-optimal solutions. And Figure 5 and 6 show that the weight setting is changed after carrying out the SA search gives a performance better than the fixed weight. The weight setting is changed after carrying out the SA search in such a way that the weight covers the entire objective-function space uniformly during the replications.

The proposed SA search method is summarized as follows;

- 1) Neighbourhood structure: mixed-shift operation
- 2) Initial solution: job number
- 3) $w = (r - 0.5)/R$, $c = 0.0012$
- 4) $N = 100,000$, $R = 100$

The proposed SA search method could find all 32 Pareto-optimal solutions of problem 1 of size (20×20) .

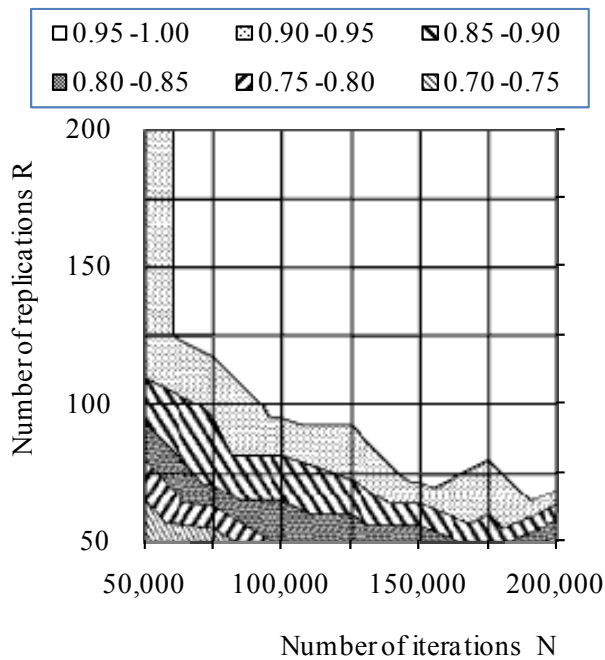


Figure 5(a). Contour line of the number of obtained Pareto solutions for the number of iterations and the number of replications ($c = 0.0012$, $w = 0.5$).

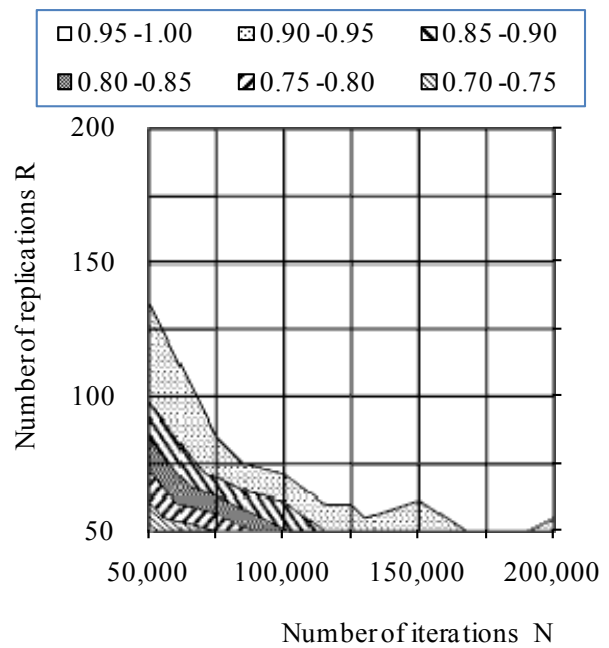


Figure 6(a). Contour line of the number of obtained Pareto solutions for the number of iterations and the number of replications ($c = 0.0012$, $w = (r - 0.5)/R$).

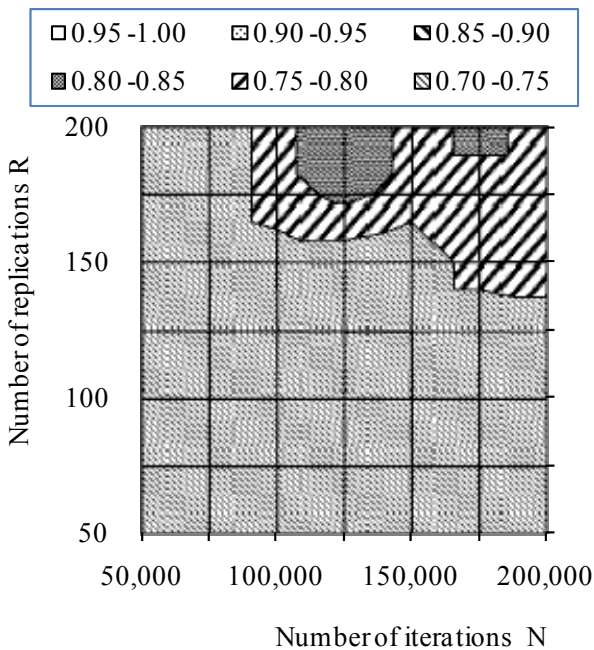


Figure 5(b). Contour line of the number of obtained Pareto solutions for the number of iterations and the number of replications ($c = 0.0012$, $w = 0.5$).

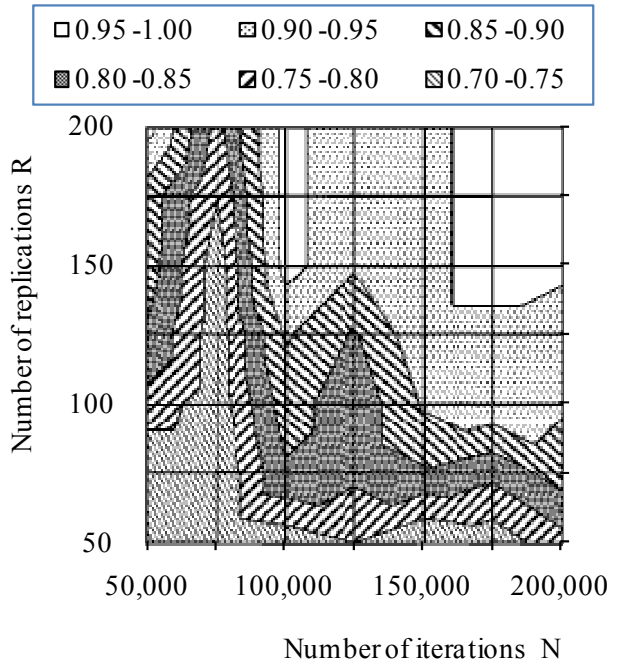


Figure 6(b). Contour line of the number of obtained Pareto solutions for the number of iterations and the number of replications ($c = 0.0012$, $w = (r - 0.5)/R$).

7. PERFORMANCE OF THE PROPOSED SA SEARCH METHOD

In the MOSA, Varadharajan *et al.* (2005) use 10 problems of size(20×20). A numerical experiment is performed at the above conditions and using 10 problems. Table 1 shows the rate of finding the Pareto-optimal solutions for 10 problems of size(20×20). Table 1 shows that the proposed SA search method is good performance in all instances.

Table 1. Rate of finding the Pareto-optimal solutions.

Taillard (20×20) Problems	Varad- Harajan (2005)	Hirakawa and Ishigaki	
		$W = 0.5$	$W = (r-0.5)/R$
1	0.28	0.97	1.00
2	0.03	0.59	1.00
3	0.20	0.64	0.93
4	0.47	0.89	1.00
5	0.03	0.39	0.97
6	0.14	0.67	0.95
7	0.22	0.61	1.00
8	0.02	0.54	0.94
9	0.21	0.33	0.97
10	0.21	0.75	0.96

8. CONCLUSION

We have considered the problem of permutation-flowshop scheduling with minimizing the makespan and the total flowtime of jobs, and proposed an effective heuristics based on SA in which the weighted sum of the makespan and the total flowtime of jobs is used. The essences of the heuristics are in the selection of the initial sequence, weight setting and neighbourhood structure to generate a solution in the search process.

The initial sequence of job number is better than the sequence near the Pareto-optimal solutions. The weight setting is changed after carrying out the SA search in such a way that the weight covers the entire objective-function space uniformly during the replications. The mixed-shift and best-shift operations outperform other neighbourhood structures.

It is necessary to investigate the performance of the proposed method for solving various problems; however, these results will support the fact that the SA search algorithm with the proposed settings is efficient.

REFERENCES

- Varadharajan, T. K. and Rajendran, C. (2005), A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs, *European Journal of Operational Research*, **167**, 772-795.
- Taillard, E. (1993), Benchmark for basic scheduling problems, *European Journal of Operational Research*, **64**, 278-285.
- Ishibuchi, H. and Murata, T. (1998), A Multi-objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling, *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, **28**, 392-403.
- Ulungu, E. L., Teghem, J., Fortemps, P. H., and Tuytens, D. (1999), MOSA Method: A tool for solving multiobjective combinatorial optimization problems, *Journal of Multicriteria Decision Analysis*, **8**, 221-236.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983), Optimization by simulated annealing, *Science*, **220**, 671-680.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953), Equation of State Calculation by Fast Computing Machines, *Journal of Chemical Physics*, **21**, 1087-1092.