



A simulated annealing approach based simulation-optimisation to the dynamic job-shop scheduling problem

Dinamik atölye tipi çizelgeleme problemine bir tavlama benzetimi yaklaşımı tabanlı simülasyon optimizasyonu

Çağrı SEL^{1*}, Alper HAMZADAYI²

¹Industrial Engineering Department, Faculty of Engineering, Karabuk University, Karabuk, Turkey.
cagrisel@karabuk.edu.tr

²Industrial Engineering Department, Faculty of Engineering, Van Yuzuncu Yil University, Van, Turkey.
alperhamzadayi@yyu.edu.tr

Received/Geliş Tarihi: 06.07.2017, Accepted/Kabul Tarihi: 02.10.2017

* Corresponding author/Yazışılan Yazar

doi: 10.5505/pajes.2017.47108

Research Article/Araştırma Makalesi

Abstract

In this study, we address a production scheduling problem. The scheduling problem is encountered in a job-shop production type. The production system is discrete and dynamic system in which jobs arrive continually. We introduce a simulation model (SM) to identify several situations such as machine failures, changing due dates in which scheduling rules (SRs) should be selected independently. Three SRs, i.e. the earliest due date rule (EDD), the shortest processing time first rule (SPT) and the first in first out rule (FIFO), are incorporated in a SM. A simulated annealing heuristic (SA) based simulation-optimisation approach is proposed to identify the unknown schedules in the dynamical system. In the numerical analysis, the performance of SRs and SA are compared using the simulation experiments. The objective functions minimising the mean flowtime and the mean tardiness are examined with varying levels of shop utilization and due date tightness. As an overall result, we observe that the proposed SA heuristic outperforms EDD and FIFO, the well-known SPT rule provides the best results. However, SA heuristic achieves very close results to the SPT and offers a reasonable computational burden in time-critical applications.

Keywords: Job-shop scheduling, Discrete and dynamic system, Simulated annealing algorithm, Simulation-optimisation, Scheduling rules

Öz

Bu çalışmada, bir üretim çizelgeleme problem ele alınmaktadır. Bu çizelgeleme problemine atölye tipi bir üretim tipinde karşılaşılmaktadır. Üretim sistemi sürekli iş gelişlerinin söz konusu olduğu kesikli dinamik sistemdir. Çizelgeleme kurallarının birbirinden bağımsız şekilde kullanılmasının gerektiği makine bozulmaları ve değişen teslim süreleri gibi bazı durumların değerlendirilmesi için bir simülasyon modeli sunulmaktadır. En erken teslim süresi, en kısa işlem süresi ve ilk giren ilk çıkar kuralı olmak üzere üç çizelgeleme kuralı bu simülasyon modeline dahil edilmiştir. Dinamik sistemdeki belirsiz çizelgeleri ortaya koymak için tavlama benzetimi sezgisel tabanlı bir simülasyon optimizasyonu yöntemi önerilmektedir. Sayısal analizlerde çizelgeleme kurallarının ve önerilen tavlama benzetimi sezgiselinin performansları simülasyon deneyleri kullanılarak kıyaslanmıştır. Ortalama akış süresini ve ortalama gecikme süresini en küçükleyen amaç fonksiyonları farklı seviyelerdeki atölye kullanım oranı ve teslim süresi durumlarında incelenmiştir. Genel bir sonuç olarak, önerilen tavlama benzetimi sezgiselinin en erken teslim zamanı ve ilk giren ilk çıkar kurallarından daha iyi sonuç verdiği, en kısa işlem süresi kuralının en iyi sonuçları sağladığı gözlenmiştir. Fakat tavlama benzetimi sezgiseli en kısa işlem süresi kuralına çok yakın sonuçlara erişmektedir ve çözüm zamanının kritik olduğu uygulamalarda kabul edilebilir bir hesaplama yükü getirmektedir.

Anahtar kelimeler: Atölye tipi çizelgeleme, Kesikli ve dinamik sistem, Tavlama benzetimi algoritması, Benzetim optimizasyonu, Çizelgeleme kuralları

1 Introduction

Scheduling is to allocate the production resources (e.g., man-power, machine and time) to the activities that must be done [1]. The activities are subject to the precedence relationships because of technological constraints of the allocated resources [2] and to be specified in a sequential order [3]. Scheduling effects the performance of production systems [4]. A good scheduling reduces the effort, time or cost of the production [5],[6].

The shop scheduling problems are categorized in several production types such as job-shop, flow-shop, open-shop, mixed-shop and group-shop [7]. In a job-shop scheduling problem (JSP), the jobs are sequenced in an order that the operations are performed in multiple machine [8]. JSP is a common problem in most of production environment [9].

The classical problem assumes that all jobs are available at the beginning of the production period [10]. JSP is to arrange all the jobs to be processed on the multiple machines aiming at optimization of the defined objectives [11]. The objectives are minimisation of the makespan, the mean flowtime and the mean tardiness [12].

The problem refers to a static scheduling problem where a known collection of jobs operated on the machines. Each of the jobs has a pre-determined processing time in a certain operation sequences originates from the technological constraints of the machines [8]. The system information such as the processing time and precedence order of operations for each job, is not changed with any disturbances interrupting the processing operations [11].

The production systems are inherently dynamic with job arriving over time [13]. The system information is uncertain and changes over time. The new jobs should be accounted for the schedule as soon as possible [14]. They are affected by

random perturbations [11] and subject to accidental disturbances such as cancellation of jobs in queues, changing urgency of some jobs, machine failures or temporary unavailability of some resources [5].

The classical solution approaches may help in solving the static scheduling problems [5]. The job-shop scheduling problems are NP-Hard problems [15]. Increasing problem size for solving real-life instances increases the computational time exponentially [13]. Various mathematical formulations SRs, heuristics and artificial intelligence techniques such as neural networks, linear/non-linear searches and metaheuristics are usually designed to optimise the static scheduling problems [9],[16].

The real production systems are complex and hence, the approaches are impractical in the dynamic conditions of real-life [17]. SMs mimic the characteristics of the production systems. They are proven to be powerful tools making analysis on real-life settings [18],[19], on dynamic behaviour of the systems [20],[21]. As it is not an optimisation tool, an extra step is needed to bring the complementary strengths of simulation and optimisation techniques together [22].

In this study, the dynamic job-shop scheduling problem (DJSP) is addressed. The dynamic nature of production environments is represented with new jobs arriving continuously and changing due dates in line with our previous study [23]. The job-shop include several universal machines. The machines are subjected to failures. Each job proceeds the predefined operations in a predefined sequence on the machines. The job-shop production system is a queuing system. A job proceeds on the machines following its route. The rest of the jobs waits, and a queue is formed in front of the occupied machine.

The order and/or priority of the waiting jobs in the queue is decided in DJSP. For this purpose, we introduce a SA heuristic

in addition to the well-known EDD, SPT and FIFO rules. The approaches are incorporated into a discrete-event SM. In general, the performance measurement criteria in SMs are lead time and lead time deviation [24]. The lead time is a flowtime oriented performance criterion and, a deviation from the lead time results in tardiness. Therefore, SA algorithm is compared to SRs according to the mean flowtime and the mean tardiness values.

This research aims to make a twofold contribution to the scheduling literature. The first contribution is to develop a support tool for the scheduling decisions in the job-shop production type. The second contribution is to present the applicability of the simulation-optimisation method into DJSP.

The following section presents a literature summary on DJSP and clarifies the contribution of our study. The subsequent sections first present the formal description of the problem and then describe SRs, SA heuristic and SM. This section is followed by the numerical analysis comparing the performance of the approaches on a hypothetical case. The last section is the conclusion section discussing the overall results and the future directions.

2 Related literature

The research papers are summarized in terms of dynamic factors, solution methods, and objective functions in Table 1. This summarized review shows the need for heuristic/metaheuristic algorithms and, SA heuristic has a potential of solving DJSP. Our study bridges this gap for solving DJSP with continuously arriving jobs, machine failures and changing due dates. In this study, we propose a SA heuristic-based simulation-optimisation approach for minimisation of the mean flowtime and the mean tardiness.

Table 1: An overview of approaches on DJSP.

Authors	Dynamic Factors	Method	Objective Function
Adibi et al. [9]	Randomly arriving jobs and machine failures	Artificial neural networks and variable neighbourhood search	Makespan and tardiness
Alotaibi et al. [5]	Continuously arriving new jobs and machine failures	Dynamic Agent-based approach	Robustness
Chang [25]	Continuously arriving new jobs	Heuristic methods	Queueing time
Dominic et al. [26]	Continuously arriving new jobs	Combined SRs	Flowtime, tardiness and tardy jobs
Fattahi and Fallahi [13]	Continuously arriving new jobs, additional machines and changing processing times,	A genetic algorithm	Efficiency and stability
Gao et al. [27]	Machine failures and order cancellation	A hybrid approach (ant-colony/genetic algorithms)	Makespan, utilization rate and schedule deviation degree
Ghomi and Iranpoor [28]	Order acceptance/rejection	A hybrid approach (genetic/SA algorithms)	The earliness-tardiness-lost sales
Hao and Gen [29]	Randomly arriving jobs and machine failures	An evolutionary algorithm	Weighted fitness in terms of effectiveness and stability
Hao and Lin [30]	Randomly arriving jobs and machine failures	Interactive adaptive-weight evolutionary algorithm	Makespan, transportation cost, setup cost and available resources
Holloway and Nelson [31]	Process times	A heuristic scheduling procedure	Tardiness and tardy jobs
Holthaus [32]	Continuously arriving new jobs	SRs	Flowtime and due date-based objectives
Holthaus and Rajendran [3]	Continuously arriving new jobs	SRs	Flow-time, tardiness and tardy jobs
Hosseinabadi et al. [33]	Input of jobs	A new heuristic approach	Makespan, flowtime and latency time
Kapanoglu and Alikalfa [34]	Continuously arriving new jobs	Genetic algorithm	Tardiness

Table 1: An overview of approaches on DJSP (continued).

Authors	Dynamic Factors	Method	Objective Function
Kundakcı and Kulak [8]	Continuously arriving new jobs, machine failures and changing processing times	Hybrid genetic algorithms	Makespan
Kutanoğlu and Sabuncuoğlu [35]	Unexpected machine failures	Reactive scheduling	Utilization, machine failures
Liu et al. [7]	Machine failures and continuously arriving new jobs	Tabu search	Makespan
Li and Chen [36]	Continuously arriving new jobs and machine failures	A hybrid approach (artificial neural networks / genetic algorithms)	Makespan
Lu and Romanowski [37]	Continuously arriving new jobs	SRs	Makespan, flowtime
Muhlemann et al. [38]	Intermittently arriving jobs, uncertain processing times and machine failures	Heuristic methods	Flowtime, lateness, process time and queueing time
Nahavandi et al. [15]	Continuously arriving new jobs	Genetic Algorithm	Makespan
Nelson, Holloway, and Wong [39]	Intermittently arriving jobs and process times represented statistically	A heuristic scheduling procedure	Tardiness
Nie et al. [10]	Continuously arriving new jobs	A new heuristic approach	Makespan, flowtime, tardiness and lateness
Nguyen et al. [12]	Dynamic due-date rules, dynamic processing and waiting times	Four new multi-objective heuristic approaches	Makespan, tardiness
Rajendran and Holthaus [40]	Dynamic job arrivals	SRs	Flowtime, tardiness and tardy jobs
Rangsaritratamee et al. [14]	Dynamic job arrivals	Genetic local search	Makespan, tardiness and stability
Sabuncuoğlu and Bayız [4]	Machine failures	Reactive scheduling	Makespan and tardiness
Sharma and Jain [41]	Sequence-dependent setup times	SRs	Flowtime and tardiness, makespan, tardy jobs and setup times
Sharma and Jain [42]	Continuously arriving new jobs and sequence-dependent setup times	SRs	Flowtime and tardiness, makespan, tardy jobs and setup times
Suwa and Sandoh [43]	Machine failures	Reactive scheduling	Tardiness and scheduling frequency
Qi et al. [44]	Continuously arriving new jobs	Genetic algorithm	Makespan
Qiu and Lau [11]	Continuously arriving new jobs and machine failures	A hybrid approach (artificial immune systems / priority SRs)	Flowtime, tardiness and lateness
Vinod and Sridharan [6]	Sequence-dependent setup times	New setup-based SRs	Flowtime and tardiness
Xiong et al. [45]	Continuously arriving new jobs	SRs	Tardiness and tardy jobs
Zhang et al. [46]	Randomly arriving jobs and machine failures	A hybrid approach (genetic/tabu search algorithms)	Schedule efficiency and stability
Ze et al. [47]	Emergent jobs arising, machine breakdown and order cancellation.	A hybrid approach (genetic/SA algorithms)	Makespan

3 Problem statement

The practical assumptions made on the job-shop system and the considered objectives are provided as follows:

- (i) A job-shop type of production is processed in non-identical machines,
- (ii) The machines are universal,
- (iii) There is one machine from each type,
- (iv) Only one job is operated on one machine at the same time,
- (v) The jobs can be operated by only one machine at the same time,
- (vi) The operations cannot be interrupted,
- (vii) The jobs can only be processed after its preceding jobs are completed,
- (viii) There is no alternative route,
- (ix) The setup times can be neglected,

- (x) The queue length is unlimited for any machine. Each of the parameters and variables in the following equations are listed in the nomenclature section.

$$F_i = C_i - A_i \quad (1)$$

$$F = \left[\sum_{i=1}^n F_i \right] / n \quad (2)$$

Equation (1) defines the flowtime of each job i , F_i which is the deviation between the completion time of job i , C_i and the arrival time of job i , A_i . Equation (2) calculates the mean flowtime F .

$$L_i = C_i - d_i \quad (3)$$

$$T_i = \max\{0, L_i\} \quad (4)$$

$$\bar{T} = \left[\sum_{i=1}^n T_i \right] / n \quad (5)$$

Equation (3) defines the lateness of each job i , L_i which is the deviation between the completion time of job i , C_i and the due date of job i , d_i . Equation (4) defines the tardiness of job i , T_i with the lateness of job i , L_i which is equal to and bigger than zero. Equation (5) calculates the mean tardiness \bar{T} .

4 Scheduling heuristics

Many practical scheduling problems become complex with the increasing number of jobs and machines. An optimal solution might not be reached in a reasonable time due to the NP-Hard structure of the problems. In this case, knowledge-based rules and metaheuristic algorithms are needed. The classic SRs decides for the time horizon and the next job to be run after the current job. The metaheuristic algorithms approximate the optimum values of a fitness function. Therefore, the SRs and metaheuristic algorithms might be useful in the increasing problem sizes of scheduling problems. In this section, SRs and SA heuristic inspired by annealing process in metallurgy are described. The interested readers are directed to the following papers for the details of the aforementioned discussion [5],[9],[13],[15].

4.1 Scheduling rules

EDD, SPT and FIFO are of the simple SRs. EDD rule processes the job with the earliest due date at first. It is a well-known SR minimizing the tardiness in the scheduling problems [48]. SPT rule sequence the jobs in an increasing order according to their processing times. It is the most common SR decreasing the makespan of a JSP. The rule is used to minimise the mean flowtime and the mean tardiness under overloaded shop-floor conditions and minimising the percentage of tardy jobs when the production has slack due-dates [11]. FIFO rule is a queue processing technique. In this technique, the earliest job in the queue is selected. The rule is used to minimise the maximum flowtime and the variance of flowtime.

4.2 Simulated annealing heuristic

SA heuristic is used to determine priority of jobs waiting in front of the machine queue. The idea behind SA first presented by Metropolis et al. [49]. Kirkpatrick et al. [50] used SA in combinatorial optimisation problems. There are many local optimum solutions in optimisation problems. The basic algorithms generate a neighbourhood solution from a random initial solution. The cost decreasing neighbourhood solution is accepted for the further iterations. Therefore, the algorithm often converges to a local optimum. SA avoids from getting stuck in a local optimum by saving the neighbours which have non-decreasing cost value with a pre-defined probability. The algorithm parameters are initial temperature level $Temp_o$, final temperature level $Temp_{fin}$, number of decrease in the temperature level q , and number of iterations for each temperature level t . In SA, an initial solution is randomly generated to reach a neighbourhood solution (intensification), and the search is extended to the solution space. The algorithm sometimes allows the search to move away from a local optimum (diversification). If the new neighbourhood solution is not improving the cost, SA can accept the neighbourhood solution with the criteria presented in Equation (6). The algorithm uses the slow temperature reduction to reach the global optimum.

$$u < e^{-\frac{\Delta}{Temp}} \quad (6)$$

SA algorithm is preferred because of its neighbourhood structure that is easy to implement. The neighbourhood generation logic is shown in Figure 1. A neighbour solution can be reached either completely changing the priorities of the randomly selected two different queues (see Figure 2) or swapping the priorities of randomly selected two different tasks in the randomly selected a queue (see Figure 3). The random number r determines the type of the neighbour solution.

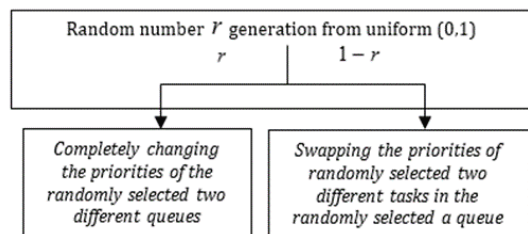


Figure 1: Neighbourhood generation of the SA algorithm.

Queue	Before changing the priority										After changing the priority									
1	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1
3	3	4	5	6	7	8	9	10	1	2	7	8	9	10	1	2	3	4	5	6
4	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3
5	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4
6	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5
7	7	8	9	10	1	2	3	4	5	6	3	4	5	6	7	8	9	10	1	2
8	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7
9	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8
10	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9

Figure 2: Completely changing the priorities of the randomly selected two different queues.

Queue	Before swapping the priorities										After swapping the priorities									
1	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1
3	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2
4	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3
5	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4
6	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5
7	7	8	9	10	1	2	3	4	5	6	1	8	9	10	7	2	3	4	5	6
8	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7
9	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8
10	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9

Figure 3: Swapping the priorities of randomly selected two different tasks in the randomly selected queue.

5 Modelling approach

As a modelling approach, we introduce a SM by using Arena 10.0. The Arena logic model of this system is presented in Figure 4. The simulation analysis is made at the steady state which is determined by Welch's procedure [51].

The Welch's procedure is a graphical method which tracks the moving averages of the defined performance criteria. Machine failures are modelled with the busy time approach in the next machine failure depends on the total busy time [51].

The problems include nonlinearities, combinatorial relationships and uncertainties in real-life which increase the complexity of analytical models. However, simulation-optimisation is an appropriate tool to optimise the system settings in such cases [22]. Simulation-optimisation is used to find the appropriate parameter values for a system in which the system performance is evaluated by the outputs achieved by a SM.

Therefore, to optimise a simulated system, two components are required: the simulation of the system and an optimisation procedure.

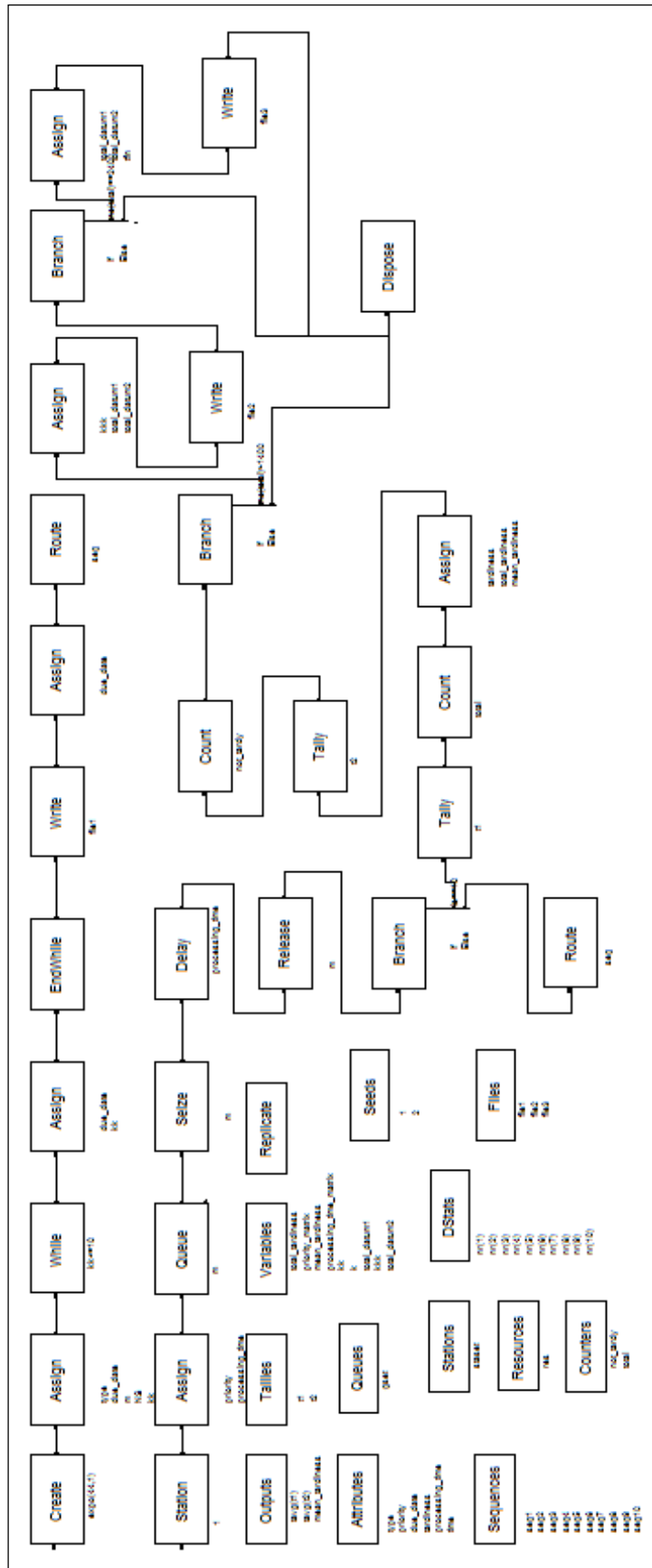


Figure 4: Arena model logic of DJSP.

The optimisation procedure can be automated in the form of a computer program that selects system inputs based on a specified optimisation technique and runs the simulation using these inputs to produce an estimate of the system output. The output is then fed back into the optimisation program where it is used by the optimisation technique to select further system inputs. The algorithm proceeds checking the stopping criteria.

Further, in this study, we introduce an embedded SA heuristic in SM instead of using simulation in scheduling with the aid of optimization. SM mimics the production process in job-shop production type. The proposed simulation-optimisation approach decides and revises continuously the sequencing of jobs in front of the machine queues in SM.

Initial solutions are obtained from SM by the randomly generated priorities of the tasks in the queues. The SA algorithm changes the priorities of the jobs waiting on queues in front of the machines and suggests a new priority configuration for jobs at each iteration. Next, this priority configuration is used in SM. The SM evaluates the objective function value of the given priority configurations of jobs. In Figure 5, the approach is described as the integration of a SA algorithm module with a simulation module in a closed loop configuration.

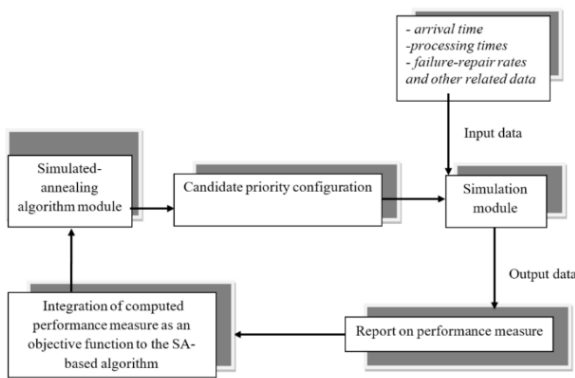


Figure 5: The control logic of the simulation-optimisation approach.

6 Numerical analysis

In the numerical analysis, a JSP consisting of 10 machines is examined. The simulation experiments are conducted to provide the comparative analysis on SRs and SA algorithm.

6.1 Hypothetical case

The case includes ten types of jobs which have equal probability to arrive at the shop. All jobs have different deterministic processing times, which is known and deterministic. Ten operations are performed for each job. The work station routes for each type of jobs are known in advance and given in Table 2. Processing times are produced by an exponential distribution with the mean of 30 min. The processing times are given in Table 3. The mean time between machine failures and the mean time to repair times for the machines are represented by the exponential distribution with the mean of 400 min. and 100 min. respectively.

From the literature, we know that the job arrivals closely follow Poisson distribution [14]. Therefore, the time between the arrivals is presented by an exponential distribution. To define the parameter of the exponential distribution (i.e. the mean of

the times spent between the job arrivals) the mean processing time of all jobs is divided to shop utilization in Equation (7). Using this utilization rate, pilot experiments have been carried out and the bottleneck has been detected on the seventh workstation with the utilization rate as 0.9999. Therefore, mean processing time of the seventh workstation as 39.6002 is used to calculate shop utilization rate of the system. The simulation experiments are performed by two shop-utilization levels ($U=0.90$ and $U=0.87$). It should be noted that the values of U are used in Equation (7) to calculate MIAT, \bar{A} values, i.e. 44 min. and 46 min. respectively.

$$\text{Mean inter - arrival time } (\bar{A}) = \frac{\text{Mean processing time of the bottleneck workstation}}{\text{Shop utilization } (U)} \quad (7)$$

In JSP, the total work content (TWK) method is commonly used for assigning due dates [48]. In the TWK method, the due dates are assigned with the sum of the arrival times and the total processing times. In Equation (8), due date tightness factor k varies in three different values. The increasing factor increase the slackness of the due dates, i.e. $k = 3$ provides tight due dates, $k = 5$ provides moderate due dates and $k = 7$ provides slack due dates.

$$d_i = A_i + k * p_i \quad (8)$$

6.2 Steady state conditions

In this study, first a pilot simulation experiment is conducted to reach the steady state conditions and ten replications are made. The optimal number of replication is determined using truncated replications method [51]. Each replication simulates the production operations to complete 5.000 jobs. The experimental settings are that the mean inter-arrival time (MIAT) of jobs is equal to 44 min, the due date tightness factor is equal to 5, SR is FIFO. The window lengths are set 30 and the graphical solution obtained from Matlab has been shown for the mean flowtime and the mean tardiness in Figure 6 and Figure 7, respectively. From the graphics, the warm-up period is determined as 1.400 min.

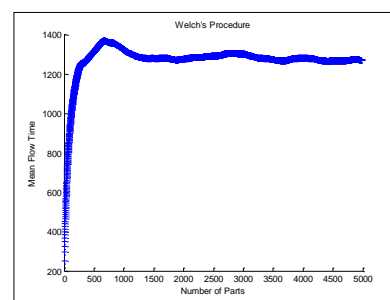


Figure 6: Moving average plot for mean flowtime (in min.)

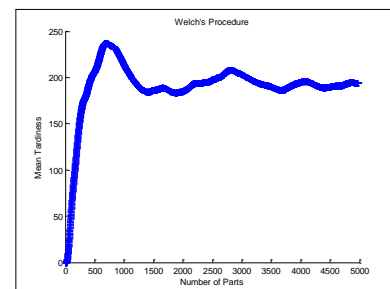


Figure 7: Moving average plot for mean tardiness (in min.)

Table 2: The work station routes for each type of job (default route).

JobType/Operation	1	2	3	4	5	6	7	8	9	10
1	1*	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	1
3	3	4	5	6	7	8	9	10	1	2
4	4	5	6	7	8	9	10	1	2	3
5	5	6	7	8	9	10	1	2	3	4
6	6	7	8	9	10	1	2	3	4	5
7	7	8	9	10	1	2	3	4	5	6
8	8	9	10	1	2	3	4	5	6	7
9	9	10	1	2	3	4	5	6	7	8
10	10	1	2	3	4	5	6	7	8	9

*: The numbers given under the Job/Operation match represent the workstations.

Table 3: Processing times of the operations in the machines (in minutes).

Operation/Work st.	1	2	3	4	5	6	7	8	9	10
1	8.816	8.237	70.991	10.200	15.836	71.448	65.774	21.116	1.096	161.255
2	26.164	7.135	4.883	3.605	9.588	0.089	31.699	2.009	43.127	5.982
3	28.961	18.349	13.159	71.628	42.335	141.989	24.564	58.216	8.858	9.901
4	6.028	48.831	41.752	8.950	16.974	2.963	40.038	1.554	67.111	13.384
5	26.173	14.066	1.218	16.555	5.118	34.761	15.381	12.079	16.364	63.710
6	54.444	47.658	29.648	3.179	54.091	11.209	22.743	40.985	6.511	0.303
7	27.234	55.971	2.427	67.845	8.058	55.556	0.348	45.808	2.003	1.472
8	59.757	18.513	8.768	50.648	87.802	8.522	157.400	19.757	9.829	23.866
9	71.763	16.769	23.236	15.392	28.930	2.684	21.393	60.414	13.487	8.133
10	7.973	37.364	8.075	10.429	31.076	18.508	16.662	27.861	35.818	59.813
Mean	31.731	27.289	20.416	25.843	29.981	34.773	39.600	28.980	20.420	34.782

6.3 Experimental conditions

Full-factorial experimental design is used with ten replications (Table 4). The two levels of MIAT, three levels of due date tightness factor, two performance criteria and four scheduling heuristics (i.e. three SRs plus SA algorithm) result in $2 \times 3 \times 4 \times 2 = 48$ different runs. 2.400 jobs are completed in each replication. The first 1400 jobs are discarded to reach the warm-up level. The performance criteria are computed by using the following 1.000 jobs.

Table 4: Experimental conditions.

Conditions	Values
MIAT, \bar{A}	44 min, 46 min
Due date tightness factor, d	3, 5, 7
Performance criteria	Mean Flowtime, Mean tardiness

Table 5: The control parameters for the numerical illustration.

Parameters	Values
Initial temperature	100
Cooling rate	0.99
Length of each temperature level	10
Crystallization temperature	1
Probability of random number, r	0.02

6.4 Numerical illustration

A numerical example has been used to illustrate the effect of the changing priorities of jobs on mean tardiness. MIAT and due date tightness factor are set to 44 min and 3, respectively. The parameters that are used in the SA algorithm for the numerical illustration are given in Table 5. Table 6 shows the procedure of the initial solution applied to the numerical example.

Table 5: The job priorities in the initial solution.

Queue/JobType	1	2	3	4	5	6	7	8	9	10
1	10*	6	7	4	2	5	1	9	8	3
2	10	1	2	8	7	9	4	3	6	5
3	2	4	10	7	6	5	3	9	8	1
4	7	2	4	10	3	5	9	1	6	8
5	10	9	8	4	7	3	2	1	6	5
6	8	6	2	4	7	3	1	9	10	5
7	9	3	5	6	7	8	1	10	4	2
8	9	1	3	2	10	6	4	5	7	8
9	2	6	7	3	8	9	1	5	4	10
10	6	5	8	7	10	3	2	9	1	4

Mean Tardiness=673.38, *Job priorities.

Table 6: The job priorities in an intermediate solution.

Queue/JobType	1	2	3	4	5	6	7	8	9	10
1	8*	6	2	4	10	9	7	5	3	1
2	6	2	3	1	4	8	7	5	10	9
3	7	5	3	6	9	10	2	4	8	1
4	3	8	6	7	9	2	1	5	10	4
5	4	3	9	1	2	7	10	5	6	8
6	10	8	7	2	5	3	9	1	4	6
7	7	3	8	2	1	6	5	4	10	9
8	7	4	6	2	10	1	3	9	8	5
9	2	8	9	4	5	6	3	7	1	10
10	9	3	10	1	6	8	2	4	7	5

Mean Tardiness=455.03, *Job priorities.

The initial solution is obtained as 673.38 min. After several iterations of SA heuristic, an intermediate solution and so the best solution is generated starting from the initial solution. The

intermediate solution is 455.03 min (Table 7), the best solution is 418.89 min (Table 8).

Job priorities sequence the waiting jobs in front of the queues of the limited resources. The lower the job priority number corresponds to the higher priority. If there is more than one job waiting on a queue, the job with the highest priority gets the resource first.

Table 7: The job priorities in the best solution.

Queue/JobType	1	2	3	4	5	6	7	8	9	10
1	3*	4	7	1	5	8	6	9	10	2
2	2	1	5	9	3	8	10	6	4	7
3	10	3	6	9	1	8	2	5	7	4
4	4	2	10	3	7	1	9	8	6	5
5	4	3	8	5	1	9	2	10	6	7
6	9	1	10	3	7	5	8	4	2	6
7	9	7	6	8	2	5	1	10	4	3
8	5	2	9	1	3	7	8	4	10	6
9	1	9	4	10	7	3	2	5	6	8
10	10	3	5	6	9	1	2	7	4	8

Mean Tardiness=418.89, *: Job priorities.

Table 8: The control parameters of the SA heuristic.

Parameters	Values
Initial temperature	100
Cooling rate	0.6
Length of each temperature level	5
Crystallization temperature	1
Probability of p1	0.45

6.5 Results and discussion

The outputs of ten replications are averaged for each of the experimental settings. Numerical illustration has taken considerable amount of computer time (appx. 3.5 hr.) and different parameter settings are used to run the different experimental settings.

The parameters used in SA heuristic are given in Table 8. The average results are illustrated in Figure 8 and Figure 9. The figures show how the mean flowtime tardiness values for each scheduling heuristics differs in varying MIAT (i.e. 44 min., 46 min.) and due dates of jobs (i.e., tight, moderate, slack).

7 Conclusion

The schedule provided for a DJSP should possess sufficient qualifications to tackle with machine failures and changing due dates. First, we introduce a SM to mimic DJSP in which order arrivals are continuous. Second, three SRs, i.e. EDD, SPT and FIFO, are incorporated into SM accounting for the machine failures. Then, SA heuristic is proposed to reveal the potential benefits of using a simulation-optimisation approach. Finally, several simulation experiments are conducted to provide the comparative analysis on SRs and SA heuristic. In the comparative analysis, we observe that SPT rule provides the best results. The SA heuristic offers a reasonable computational performance comparing to SRs, i.e. outperforming EDD and FIFO, achieving very close results to SPT.

A possible limitation in this study is that the system is assumed to have continuously arriving jobs, uncertain machine failures and changing due dates. However, other types of accidental disturbances such as cancellation of jobs in queues, changing urgency of some jobs or temporary unavailability of some resources might also be confronted in practice.



Figure 8: Comparative results for the performance criterion of the mean flowtime (in min.)

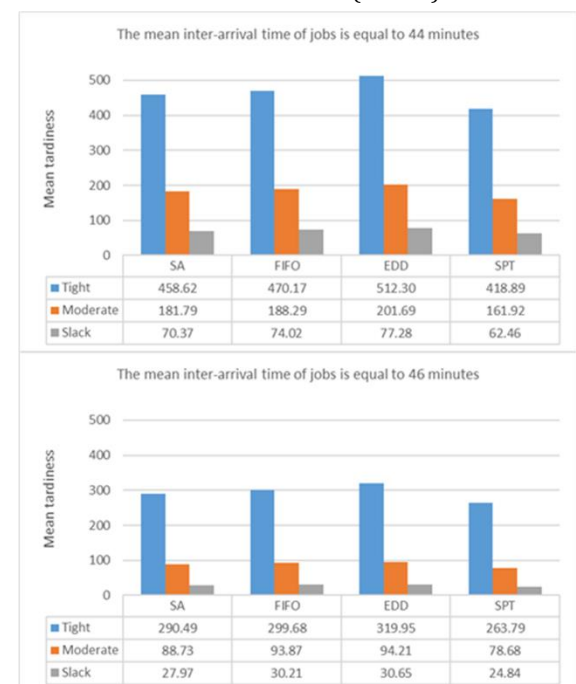


Figure 9: Comparative results for the performance criterion of the mean tardiness (in min.)

Note: Comparative results are presented in changing MIATs (i.e. 44 min., 46 min.) and due dates (i.e. tight, moderate, slack).

Generalization of the proposed SM and the simulation-optimisation approach for representing the accidental disturbances which are more error prone, would be a possible direction for a future research.

Several other extensions are possible for the current study. One extension would be to incorporate different combinations of SRs in a SM to manage the accidental disturbances. Another possible extension is to make machine selection decisions,

parameter optimizations in the proposed SM. In addition, new heuristics such as bee algorithms will enable to handle applications that are practical and larger-in-size.

8 Nomenclature

Abbreviations used in this paper:

SR	Scheduling rule
DJSP	Dynamic job-shop scheduling problem
EDD	Earliest due date
FIFO	First in first out
JSP	Job-shop scheduling problem
JP	Job-type
MIAT	The mean inter-arrival time
SA	Simulated annealing
SPT	Shortest processing time
SM	Simulation model
OP	Order of the operation for the job
TWK	Total work content
WN	Work station
Parameters describing the scheduling rules:	
d_i	Due date of job i
k	Due date tightness factor
n	Number of jobs completed after reaching the steady-state conditions
p_i	Processing time of job i
Parameters describing SA heuristic:	
r	Random number
t	Number of iterations at each temperature
q	Number of temperature decline
Variables describing the scheduling rules:	
A_i	Arrival time of job i
\bar{A}	The mean inter-arrival time
C_i	Completion time of job i
F_i	Flowtime of job i
F	Mean flowtime
L_i	Lateness of job i
T_i	Tardiness of job i
\bar{T}	The mean tardiness
Variables describing SA heuristic:	
u	The number generated from a uniform(0,1)
U	Shop utilisation
Δ	Δ is the cost deviation between the current solution and the new neighbourhood solution.
d_i	
Temp	Current temperature
Temp ₀	Initial temperature
Temp _{fin}	Final temperature

9 References

- [1] Pinedo M. *Scheduling Theory, Algorithms, and Systems*. New York, USA, Springer-Verlag, 2015.
- [2] Baker KR. *Elements of sequencing and scheduling*, Hanover, USA, Dartmouth College, 1995.
- [3] Holthaus O, Rajendran C. "New scheduling rules for scheduling in a job shop-An experimental study". *The International Journal of Advanced Manufacturing Technology*, 13(2), 148-153, 1997.
- [4] Sabuncuoğlu I, Bayız M. "Analysis of reactive scheduling problems in a job shop environment". *European Journal of Operational Research*, 126(3), 567-586, 2000.
- [5] Alotaibia A, Lohsea N, Vub TM. "Dynamic agent-based bi-objective robustness for tardiness and energy in a dynamic flexible job shop". *CIRP Conference on Manufacturing Systems*, Stuttgart, Germany, 25-27 May 2016.
- [6] Vinod V, Sridharan R. "Scheduling a dynamic job shop production system with sequence dependent setups: An experimental study". *Robotics and Computer Integrated Manufacturing*, 24, 435-449, 2008.
- [7] Liu SQ, Ong HL, Ng KM. "Metaheuristics for minimizing the makespan of the dynamic shop scheduling problem". *Advances in Engineering Software*, 36, 199-205, 2005.
- [8] Kundakçı N, Kulak O. "Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem." *Computers & Industrial Engineering*, 96, 31-51, 2016.
- [9] Adibi MA, Zandieh M, Amiri M. "Multi-objective scheduling of dynamic job shop using variable neighborhood search". *Expert Systems with Applications*, 37, 282-287, 2010.
- [10] Nie L, Gao L, Li P, Shao X. "Reactive scheduling in a job shop where jobs arrive over time". *Computers & Industrial Engineering*, 66, 389-405, 2013.
- [11] Qiu X, Lau HYK. "An AIS-based hybrid algorithm with PDRs for multiobjective dynamic online job shop scheduling problem". *Applied Soft Computing*, 13, 1340-1351, 2013.
- [12] Nguyen S, Zhang M, Johnston M, Tan KC. "Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming". *IEEE Transactions on Evolutionary Computation*, 18(2), 193-208, 2014.
- [13] Fattahi P, Fallahi A. "Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability". *CIRP Journal of Manufacturing Science and Technology*, 2, 114-123, 2010.
- [14] Rangsaritratsamee R, Ferrell W Jr, Kurz M. "Dynamic rescheduling that simultaneously considers efficiency and stability". *Computers & Industrial Engineering*, 46, 1-15, 2004.
- [15] Nahavandi N, Zegordi S H, Abbasian M. "Solving the Dynamic Job Shop Scheduling Problem using Bottleneck and Intelligent Agents based on Genetic Algorithm". *International Journal of Engineering*, 29(3), 347-358, 2016.
- [16] Aydemir E, Koruca HI. A new production scheduling module using priority-rule based genetic algorithm. *International Journal of Simulation Modelling*, 14(3), 450-462, 2015.
- [17] Ouelhadj D, Petrovic S. "A survey of dynamic scheduling in manufacturing systems". *Journal of scheduling*, 12(4), 417-431, 2009.
- [18] Koruca, Hİ, Özdemir G, Aydemir, E, Çayırılı M. "Bir Simülasyon Yazılımı için Esnek İş Akış Planı Editörü Geliştirilmesi ve İşlemlerin Gantt Şemasında Çizelgelenmesi." *Gazi Üniv. Müh. Mim. Fak. Der.*, 25(1), 77-81, 2010a.
- [19] Zülch G, Halil IK, Mikko B. "Simulation-supported change process for product customization-A case study in a garment company." *Computers in Industry*, 62(6), 568-577, 2011.
- [20] Koruca Hİ. "Simülasyon Destekli Vardiya Planlama Modülü Geliştirilmesi." *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 25(3), 369-382, 2010b.
- [21] Zülch G, Bogus T, Koruca HI, Kurbanoglu C, Brinkmeier B. "Simulation aided design of organizational structures in manufacturing systems using structuring strategies." *Journal of Intelligent Manufacturing*, 15(4), 431-437, 2004.

- [22] April J, Better M, Glover F, Kelly, J. "New advances and applications for marrying simulation and optimization". *IEEE Simulation Conference*, Washington, USA, 5-8 December 2004.
- [23] Hamzadayı A, Yıldız G. "Hybrid strategy based complete rescheduling approaches for dynamic m identical parallel machines scheduling problem with a common server". *Simulation Modelling Practice and Theory*, 63, 104-132, 2016.
- [24] Korusa, Hİ, Özdemir G, Aydemir, E, Çayırılı M. "The simulation-based performance measurement in an evaluation module for Faborg-Sim simulation software." *Expert Systems with Applications*, 37(12), 8211-8220, 2010c.
- [25] Chang FCR. "Heuristics for dynamic job shop scheduling with real-time updated queueing time estimates". *International Journal of Production Research*, 35(3), 651-665, 1997.
- [26] Dominic PDD, Kaliyamoorthy S, Kumar S. "Efficient scheduling rules for dynamic job shop scheduling". *International Journal of Advanced Manufacturing Technology*, 24, 70-75, 2004.
- [27] Gao Y, Ding YS, Zhang HY. "Job-shop scheduling considering rescheduling in uncertain dynamic environment". *International Conference On Management Science & Engineering*, Moscow, Russia, 14-16 September 2009.
- [28] Ghomi SMTF, Iranpoor M. "Earliness-tardiness-lost sales dynamic job-shop scheduling". *Production Engineering*, 4(2-3), 221-230, 2010.
- [29] Hao XC, Gen M. "Multi-objective job shop rescheduling by using evolutionary algorithm". *IEEJ Transactions on Electronics, Information and Systems*, 131(3), 674-681, 2011.
- [30] Hao XC, Lin L. "Job shop rescheduling by using multi-objective genetic algorithm." *CIE40 International Conference on Computers and Industrial Engineering*, Awaji Island, Japan, 26-28 July 2010.
- [31] Holloway CA, Nelson RT. "Job shop scheduling with due dates and variable processing times". *Management Science*, 20(9), 1264-1275, 1974.
- [32] Holthaus O. "Scheduling in job shops with machine failures: an experimental study". *Computers & industrial engineering*, 36(1), 137-162, 1999.
- [33] Hosseinabadi AAR, Siar H, Shamsirband S, Shojafar M, Nasir M HNM. "Using the gravitational emulation local search algorithm to solve the multi-objective flexible dynamic job shop scheduling problem in Small and Medium Enterprises". *Annals of Operations Research*, 229, 451-474, 2015.
- [34] Kapanoglu M, Alikalfa M. "Learning IF-THEN priority rules for dynamic job shops using genetic algorithms". *Robotics and Computer-Integrated Manufacturing*, 27, 47-55, 2011.
- [35] Kutanoğlu E, Sabuncuoğlu I. "Routing-based reactive scheduling policies for machine failures in dynamic job shops". *International Journal of Production Research*, 39(14), 3141-3158, 2001.
- [36] Li Y, Chen Y. "Neural network and genetic algorithm-based hybrid approach to dynamic job shop scheduling problem." *IEEE international Conference on Systems, Man, and Cybernetic*, San Antonio, TX, USA, 11-14 October 2009.
- [37] Lu MS, Romanowski R. "Multicontextual scheduling rules for job shops with dynamic job arrival". *International Journal of Advanced Manufacturing Technology*, 67, 19-33, 2013.
- [38] Muhlemann AP, Lockett AG, Farn CK. "Job shop scheduling heuristics and frequency of scheduling". *International Journal of Production Research*, 20(2), 227-241, 1982.
- [39] Nelson RT, Holloway CA, Wong RM. "Centralized scheduling and priority implementation heuristics for a dynamic job shop model". *AIIE Transactions*, 9(1), 95-102, 1977.
- [40] Rajendran C, Holthaus O. "A comparative study of scheduling rules in dynamic flowshops and jobshops". *European Journal of Operational Research*, 116, 156-170, 1999.
- [41] Sharma P, Jain A. "Analysis of scheduling rules in a stochastic dynamic job shop manufacturing system with sequence-dependent setup times". *Frontiers of Mechanical Engineering*, 9(4), 380-389, 2014.
- [42] Sharma P, Jain A. "Performance analysis of scheduling rules in a stochastic dynamic job shop manufacturing system with sequence-dependent setup times: Simulation approach". *CIRP Journal of Manufacturing Science and Technology*, 10, 110-119, 2015.
- [43] Suwa H, Sandoh H. "Capability of cumulative delay based reactive scheduling for job shops with machine failures". *Computers & Industrial Engineering*, 53, 63-78, 2007.
- [44] Qi JG, Burns GR, Harrison DK. "The application of parallel multipopulation genetic algorithms to dynamic job-shop scheduling". *International Journal of Advanced Manufacturing Technology*, 16, 609-615, 2000.
- [45] Xiong H, Fan H, Jiang G, Li G. "A simulation-based study of scheduling rules in a dynamic job shop scheduling problem with batch release and extended technical precedence constraints". *European Journal of Operational Research*, 257, 13-24, 2017.
- [46] Zhang L, Gao L, Li X. "A hybrid genetic algorithm and tabu search for multi-objective dynamic job shop scheduling problem". *International Journal of Production Research*, 51(12), 3516-3531, 2013.
- [47] Ze T, Lie-Yang X, Chang-Zhong H, Xiao-Xia L. "Petri Net and GASA Based Approach for Dynamic JSSP". *International Journal of Performability Engineering*, 3(2), 213-224, 2007.
- [48] Baker KR. "Sequencing rules and due-date assignments in a job shop". *Management Science*, 30(9), 1093-1104, 1984.
- [49] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. "Equation of state calculations by fast computing machines". *The journal of chemical physics*, 21(6), 1087-1092, 1953.
- [50] Kirkpatrick S, Gelatt CD, Vecchi MP. "Optimisation by simulated annealing". *Science*, 220(4598), 671-680, 1983.
- [51] Law AM, Kelton WD, Kelton WD. *Simulation modeling and analysis*. 3rd ed. New York, USA, McGraw-Hill, 1991.