International Journal of Operational Research

ISSN online: 1745-7653 - ISSN print: 1745-7645
https://www.inderscience.com/ijor

An efficient approach for solving a job shop scheduling problem with resources constraints: a case study iCIM 3000

Abdelkader Hadri, Aimade Eddine Bougloula, Fayçal Belkaid, Hacene Smadi

DOI: 10.1504/IJOR.2020.10053021

Article History:

Received:	08 November 2019
Accepted:	20 February 2020
Published online:	26 January 2023

An efficient approach for solving a job shop scheduling problem with resources constraints: a case study iCIM 3000

Abdelkader Hadri*

Department of Industrial Engineering,
University of Batna 02,
53, route de Constantine, Fesdis Batna 05078, Algeria
and
Department of Electrical and Electronic Engineering,
University of Tlemcen,
PB 230, Tlemcen, 13000, Algeria
Email: hadriabdelkader@yahoo.fr
Email: abdelkader.hadri@univ-tlemcen.dz
*Corresponding author

Aimade Eddine Bougloula

Department of Industrial Engineering,
University of Batna 02,
53, route de Constantine, Fesdis Batna 05078, Algeria
Email: imade_ed@yahoo.fr
Email: a.bougloula@univ-batna2.dz

Fayçal Belkaid

Manufacturing Engineering Laboratory of Tlemcen (MELT),
Department of Electrical and Electronic Engineering,
University of Tlemcen,
PB 230, Tlemcen, 13000, Algeria
Email: f_belkaid@yahoo.fr
Email: faycal.belkaid@mail.univ-tlemcen.dz

Hacene Smadi

Laboratory of Automation and Manufacturing (LAP),
University of Batna 02,
53, route de Constantine, Fesdis Batna 05078, Algeria
Email: h.smadi@hotmail.com

Abstract: In this work, we are interested in a job shop scheduling problem (JSSP) with resources availability constraints. The aim consists in scheduling a set of N jobs on M machines. To be processed in the system, each job needs an number of consumable resources that are available in a limited quantity. Solving such a problem means finding better jobs sequencing in order to minimise the maximum execution time. We suggest two different methods to solve the above-mentioned problem. We firstly propose a set of four heuristics based on priority rules. Then, we make call to genetic algorithm. Using a real job shop manufacturing system data, a large-scale experiment was performed in order to analyse the performance of the proposed methods. The studied system is called iCIM 3000. The simulation results reveal that the new proposed heuristics are better than genetic algorithms and achieve to good solutions in shorter time.

Keywords: job shop; non-renewable resources; heuristics; genetic algorithms; iCIM 3000.

Reference to this paper should be made as follows: Hadri, A., Bougloula, A.E., Belkaid, F. and Smadi, H. (2023) 'An efficient approach for solving a job shop scheduling problem with resources constraints: a case study iCIM 3000', *Int. J. Operational Research*, Vol. 46, No. 1, pp.73–92.

Biographical notes: Abdelkader Hadri is currently an Assistant Professor at Tlemcen University. He obtained his Engineering degree in Industrial Engineering (2007) and Magister degree in Industrial Engineering (2012) from Batna University, Algeria. He is currently working towards his PhD degree at the Industrial Engineering Department (University of Batna 2). His research interests include planning, scheduling, operation management, discrete optimisation methods and optimisation problems in manufacturing systems.

Aimade Eddine Bougloula is currently an Assistant Professor at Batna2 University. His research interests include job shop scheduling problem, optimisation problems in manufacturing systems, logistic engineering and optimisation methods.

Fayçal Belkaid is currently an Associate Professor at University of Tlemcen and member of the Manufacturing Engineering Laboratory of Tlemcen (MELT). He obtained his Doctorate degree in Manufacturing Engineering from University of Tlemcen in 2014 and his accreditation to supervise research (HDR) in Manufacturing Engineering from the same university in 2017. Furthermore, he received a Postdoctoral Certificate from Lorraine University (France) in 2014. He advised several Master's and Doctorate thesis. His research interests concern the planning and scheduling problems, optimisation methods (metaheuristic…), optimisation of logistic and production systems and supply chain management. He is author or co-author of more than 50 contributions with several book chapters and publications. He is a member of several international conference committees and international journal boards. He is a reviewer of many international journals.

Hacene Smadi is currently the Rector of Batna 2 University. His research interests include total quality management, maintenance management, and optimisation problems in manufacturing systems.

1 Introduction

Managing in flexible manufacturing systems (FMSs) is one of the most important challenges for companies in their production management and control systems. In this type of systems, the resources are generally numerically controlled by machines that can process several types of parts. These parts are stored and transported by an automated handling system (Mejía and Pereira, 2020; Zhou et al., 2019). Indeed, scheduling in this type of shop [job shop problem (JSP)] has a direct impact on the efficiency and the production costs of the corresponding manufacturing system. To this fact, it has attracted a lot of attention from researchers since 1956 (Zhang et al., 2019). However, this problem presents one of the most difficult problems in production management since it belongs to the NP combinatorial optimisation problems. His difficulty lies in the fact that we should process a variety of specific products on different types of machines with different sequences (Kassu and Eshetie, 2015).

To solve the JSSP, several exact and heuristic methods have been proposed in the literature. Most of these methods consider that the jobs execution requires only renewable resources as machines, operators, etc. (Fattahi et al., 2019; Sotskov et al., 2020; Zeng et al., 2020). However, in reality, several types of non-renewable resources can be used in the product manufacturing such as the components, energy, financial resources, moulds, etc. Indeed, the management of the resources to be consumed during the products manufacturing remains a difficult and an inherent phase because it offers many advantages, such as improving the production rate, increasing system performance and buffers optimisation (Carlier et al., 2018; Liu et al., 2018; Afshar et al., 2019). The impact of this problem clearly arises when we want to make important decisions regarding the choice of the component type as well as the optimal time to consume it, especially when it comes to quickly perishable products or those that are expensive in terms of storage.

The main objective of this paper is summarised in two key points. The first is to present and analyse the JSSP with non-renewable resources availability constraint in the FMS. Indeed, a mathematical model has been developed in order to meet this goal. The second point is to find an effective method for solving the JSSP by providing satisfactory solutions within a reasonable period of time. To achieve this goal, a metaheuristic that is based on genetic algorithms has been adapted. Moreover, four heuristics based on priority rules (PRs) are proposed. A case study of a FMS, called iCIM 3000, is presented in this paper. This system consists of two computer numerical control (CNC) machining centres that are dedicated to the machining of parts, a quality station to control the machined parts and an assembly station where the final product requires an amount of consumable resources (CRs) to be assembled.

A simulation model of this system, was constructed and a large-scale experiment was performed in order to evaluate the performance of the proposed methods. Statistical analysis of several simulations with different instances is also carried out. The simulation results reveal that the new proposed PRs-based heuristics are more efficient than GA and allow good solutions to be found in less time.

The remainder of this paper is organised as follows. Section 2 gives a short literature overview concerned with the JSP under non-renewable resources. The problem and the case study are described in Section 3. Section 4 describes the mathematical model developed to find exact solutions, while Section 5 describes the set of the meta-heuristic and heuristic approaches proposed in this paper. Section 6 outlines and discusses the results achieved by the five proposed approaches. Finally, Section 7 gives a short conclusion and describes the possibilities for the future work.

2 Literature review

For several years, many works have been focused on solving the job shop scheduling problem (JSSP) in a context of non-consumable resources. In Zhou et al. (1991), the authors proposed the use of neural networks as a resolution method. They showed how to simplify this NP-complete problem to a simple neural network. The work done in Caumond (2006) was also geared towards the same problem, but with consideration of several types of constraints like time lags, transport and other additional constraints.

Recently, series of works have dealt with the JSP in its classic version with or without constraints. In fact, this problem remains to this day among the real concerns of researchers given its crucial importance. For instance, Dabaha et al. (2018) presented a new approach based on the branch-and-bound method for solving the problem of job shop with blocking. Further, Nagata and Ono (2018) aimed to improve the makespan by proposing a method based on local research. Gong et al. (2019) proposed an effective mimetic algorithm (EMA) to solve the multi-objective JSSP. Deng et al. (2020) proposed a meta-heuristic algorithm to solve the JSP with no-wait jobs constraint.

The constraint of non-renewable resources presents a new challenge for researchers in the areas of scheduling. In project scheduling, Muritiba et al. (2018) proposed a path-relinking algorithm for the multi-mode resource-constrained project scheduling problem. The authors check to get the best schedule which minimises the total makespan of the project with the existence of renewable and non-renewable resources. In Tabrizi et al. (2019), and for the same objective (minimising the project completion), a mixed-integer programming model is developed to present simultaneous planning of project scheduling and material procurement problem.

Some studies have focused on single-machine problems with CRs. Toker et al. (1991) studied a single-machine scheduling problem with a single financial resource. The authors showed that this problem is equivalent to a flow shop composed of two machines without financial resources. They applied Johnson's algorithm to minimise the makespan.

Belkaid et al. (2016a, 2016b), in their works, dealt with the problem of production scheduling constrained by CRs in a parallel machine environment.

The first work in the job shop scheduling field introducing the CRs constraint is the work done in Grabowski and Janiak (1987). In this work, the processing time for each operation is represented by a function of the required quantity of the non-renewable resource. In order to solve this problem, the authors proposed an algorithm based on the disjunctive graph theory as well as a Branch-and-Bound technique. In the same context, Toker et al. (1994) developed an approximate algorithm and introduced two lower bounds to solve the problem of n job on m machine with non-renewable resource requirements.

Energy is another type of non-renewable resources that is considered in several recent works on JSSP. Actually, in most works, and unlike our work, energy consumption was not considered as a constraint, but rather as an objective to be optimised. In this context, we can cite Salido etr al. (2016) which dealt with this problem building on three objectives; energy efficiency, robustness and makespan. Kemmoé et al. (2015) studied the problem of job shop with a constraint of an energy consumption threshold not to be exceeded. A linear model has been proposed to simultaneously solve the power restriction and the traditional time objectives of the JSP. As for (Gondran et al. (2016), they dealt with a JSP under the constraint of energy consumption peaks. They proposed using a metaheuristic method to find a Pareto front solution based on the amount of instantaneous energy needed for planning as well as the total duration (makespan). Meng et al. (2019) studied a flexible JSP with the objective of minimising the total energy consumption. In this work, a new mixed integer linear programming (MILP) model is proposed. This model is based on two essential variables; the inactivity time variable and the inactivity energy variable.

Speaking of meta-heuristics, genetic algorithms have been widely used in solving NP-hard problems thanks to their efficiency. For instance, Seidgar et al. (2017) proposed a genetic algorithm and new self-adapted differential evolutionary for solving the two-stage assembly flow shop problems (TAFSP) by taking the machines breakdown into account. In the work of Rashid et al. (2018), a genetic algorithm has been proposed to obtain acceptable solutions for retailer's location problem with stochastic demand and service time consideration. Kumar et al. (2019) dealt with the flow shop problem (FSP) building on three constraints; the breakdown interval, the transportation time and the jobs' weights. The authors used new parameters in their algorithm in order to solve the FSP with these three constraints.

Dispatching rules are also applied in many fields of production and logistics in order to solve NP-hard scheduling problems, especially in the job shop environment (Đurasević and Jakobović, 2018). Two new heuristics, named NEHMSWG and NEHMS-PS, have been proposed by Rajendran et al. (2017), in order to solve the permutation flow-shop scheduling problem. The performance of these heuristics was investigated and the obtained results reveal that the proposed tie-breaking rules are simple and effective, and improve the solutions with respect to many problem instances. In Ozturk et al. (2019), the authors described two new approaches to extract PRs for the dynamic multi-objective JSSPs using simulation and gene expression programming.

Through this review of the literature, we note that there is little work that focuses on the JSSPs in the presence of CRs. Our work, on the other hand, emphasises this aspect by considering the CRs availability in a JSSP. Therefore, we propose, to solve the above-mentioned problem, a mathematical model, a meta-heuristic that is based on the genetic algorithms, and four new heuristics (based on PRs). Recall that the goal is to minimise the maximum execution time (makespan). Several experiments with different sizes were carried out and the obtained results are then compared to each other in order to assess the performance of the used methods.

3 Problem description and case study

3.1 System definition

The case study is represented by the FMS iCIM 3000 located in the Manufacturing Engineering Laboratory of Tlemcen (MELT) at the University of Tlemcen, Algeria. The iCIM 3000 is one of the latest relevant solutions proposed by Festo Didactic for the training of students and also to meet the scientific needs of the research centers interested in manufacturing. It is assumed to play an important role in illustrating complex topics such as production logistics and sequence planning in FMS s (Delgado Sobrino et al., 2014). This system, showed in Figure 1, is made up of machines and equipment such as:

1 Automatic storage/retrieval system (AS/RS), it is the main storage for all the materials used during production in the iCIM system, i.e., raw materials, semi-finished products as well as finalised products.

2 Conveyor system (FMF-pallet transfer system), which is used to transfer the products from the stock to the stations and /or vice versa.

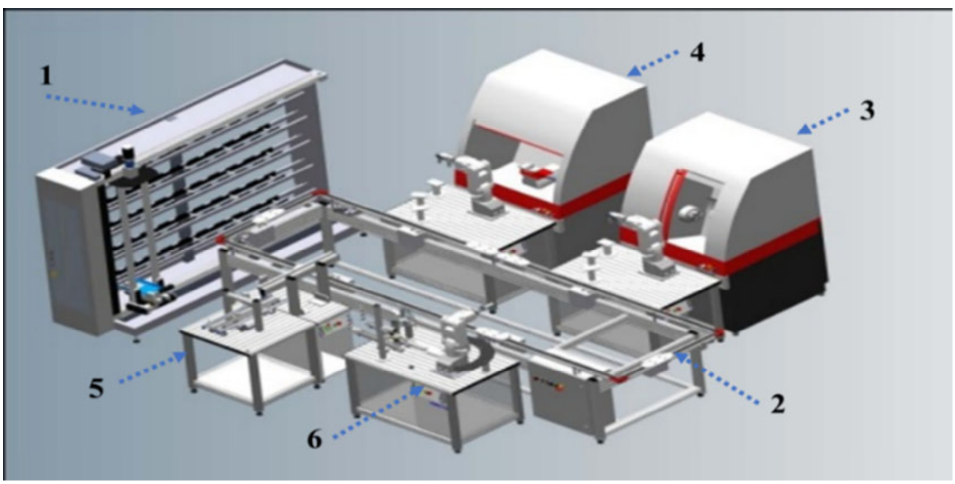
3 Concept milling 105 with flexible robot feeder, on the holes of different diameters can be made.

4 Concept turning 105 with flexible robot feeder is responsible for the production of the single rotary parts.

5 Quality and handling station (QH 200), which is responsible for the work pieces testing and the manual feeding of the system with pallets.

6 Flexible robot assembly cells (FAC), the function of this station is to assemble products from semi products created by the CNC machines or other products stocked in the AS/RS station.

Figure 1 3D configuration of iCIM 3000 system (see online version for colours)

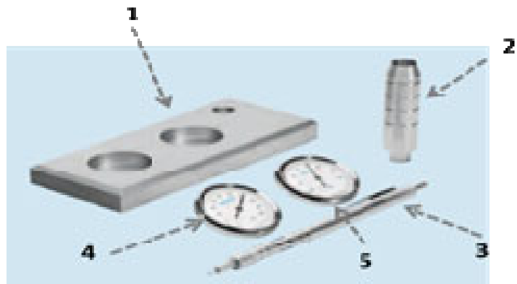


Source: Delgado Sobrino and Košťál (2013)

The final product produced and mounted by the iCIM 3000 system is a desk set (Figure 2). This product consists of five parts: base plate, pen holder, pen, thermometer device used for temperature measurement, and finally a hygrometer device used for humidity measurement.

Depending on the number and type of the components to be assembled as well as the number of holes in the base plate, the system allows production of several variations of this product. Each type of products has a specific material flow that contains a typical structure of operations to be performed. Table 1 shows an example of specific operating ranges for the realisation of three different parts: A, B and C.

Figure 2 Job information (see online version for colours)



Notes: (1): Base plate; (2): penholder; (3): pen; (4): thermometer device; (5): hygrometer device.

Table 1 Jobs information

Product	*Description*	*Flow*
A	Base plate with pen holder, pen and thermometer device.	Milling + Testing + Assembly
B	Base plate with pen holder and pen	Tour + Assembly
C	Base plate with thermometer device and hygrometer device.	Milling + Tour +Testing + Assembly

3.2 Notations and suppositions

The problem studied here can be represented as a JSSP with non-removable resources availability constraints. In this problem, it is assumed that a set of n jobs (parts) $N = \{J_1, J_1, \ldots, J_n\}$ must be executed, according to its own manufacturing range, on the four machines (stations) $M = \{M_1, M_1, M_3, M_4\}$; O_{ij}^k represents the operation i of the job J_j which will be executed on the machine M_k during a processing time P_{jk}. The machine M_4 represents the assembly station on which each job consumes an amount quantity of components which are stored in a limited capacity buffer. The arrival of the components in this stock is assumed periodic that is, arriving of a same quantity at each unit of time. The goal is to minimise the maximum execution time (makespan). According to the notation established by Graham et al. (1979), the problem can be defined as follows: $J4/NR/C_{\max}$. This problem is based on the following assumptions:

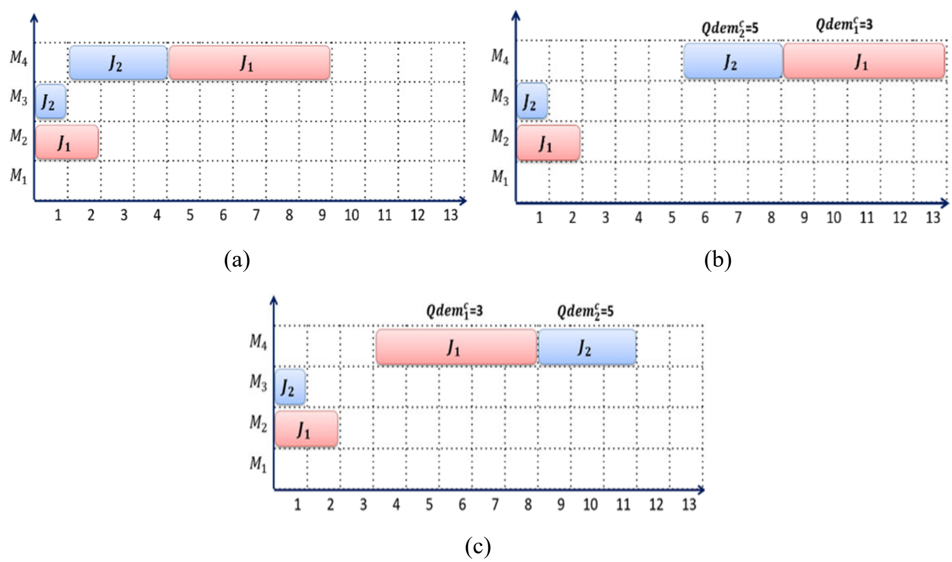
- Each job has its own routing, which is known before the beginning of the production.
- The processing time is known at the beginning and it includes tool change, transport, setup and machining times.
- Each machine can process without interruption one piece at a time.
- The blocking problem is not considered, i.e., each job can be moved to the next machine (without delay) once it is processed in the previous machine.
- Each job can be handled in one machine at a time.
- The CRs to consider are: pens, hygrometers and thermometers.

The problem arises when we want to know the best sequencing of the jobs which gives the minimum makespan while ensuring the availability of resources to consume. As an illustration example, we consider two jobs described by the data in the following Table. In this example, the jobs J_1 and J_2 require respectively five and three units of the same resource to be processed. The arriving of this resource is limited by one resource at one-time unit.

Table 2 Jobs information

		Machine (processing time)		$Qdem_j^c$
Jobs	J_1	M_2 (2)	M_4 (5)	3
	J_2	M_3 (1)	M_4 (3)	5

Figure 3 Solutions obtained with and without consideration of resources, (a) optimal solution in classical JSSP case (b) solution not optimal in JSSP with CR case (c) optimal solution in JSSP with CR case (see online version for colours)



The optimal solution of the considered problem, in its classical form (without CRs), is given by the sequencing shown in Figure 3. This sequencing does not necessarily give an optimal solution when considering the availability of CRs, which is well noted in Figures 3(a) and 3(b).

This example shows that, in the case where the constraint of refutable CRs is considered within a classic version of JSSP, it is not enough to find an optimal order of jobs and to keep it as the optimal solution. This fact is declared in Toker et al. (1994).

4 Mathematical model

In this section, we present a mathematical model that is based on mixed integer variables. The main objective expected from the creation of this model is to define the JSSP and to find schedules with the best value of makespan. In addition, this optimal solution is used to evaluate the performance of the other solutions that are generated by the other proposed methods.

4.1 Clues and sets

i operation index

j job index

k machine index

p position index

c resource index

V a very great value

J set of n jobs

M set of m machines

C set of r types of resources

$T = [1...T_{\max}]$ time interval representing the arrival times of resources.

4.2 Parameters (data)

P_{jk} processing time of job j in machine k

nb_j number of operations per job

np_m number of positions per machines

$Qdem_j^c$ amount of resource c requested by the job j

$Qarr_t^c$ amount of resource c that arrives at time t

$Tarr_t^c$ time of arrival of resources

$$Y_{jk}^i = \begin{cases} 1 & \text{if operation } i \text{ the job } j \text{ will be performed in mcahine } k \\ 0 & \text{if not} \end{cases}$$

4.3 Variables

T_{jk} the start time of job j in machine k

F_k^p the end of the position p in machine k

Qst_k^{pc} total amount of requested resources until the end of the position p in machine k

$C_{\max}$ the makespan.

4.4 Decision variables

$$X_{jk}^{p} = \begin{cases} 1 & \text{if the job } j \text{ will be performed in the position } p \text{ of machine } k \\ 0 & \text{if not} \end{cases}$$

$$Z_{k}^{pt} = \begin{cases} 1 & \text{if } F_{k}^{p} \geq Tarr_{t}^{c} \\ 0 & \text{if not} \end{cases}$$

4.5 Objective function

The objective function to satisfy in our work is the minimisation of $C_{\max}$ (the total completion time)

$$Z = Min\left(C_{\max}\right) \tag{1}$$

4.6 Constraints

$$\sum_{j=1}^{n} X_{jk}^{p} = 1 \qquad \forall k \in M, \quad \forall p = 1 \ldots np_k \tag{2}$$

$$\sum_{p=1}^{np_k} X_{jk}^{p} = 1 \qquad \forall k \in M, \quad \forall j \in J \tag{3}$$

$$F_{k}^{1} \geq \sum_{j=1}^{n} P_{jk} * X_{jk}^{1} \qquad \forall k \in M \tag{4}$$

$$F_{k}^{p} \geq F_{k}^{p-1} + \sum_{j=1}^{n} P_{jk} * X_{jk}^{p} \qquad \forall k \in M, \quad \forall p = 2 \ldots np_k \tag{5}$$

$$\begin{aligned} &\sum_{k=1}^{m}\left(Y_{jk}^{i} * F_{k}^{p}\right) + \sum_{k=1}^{m}\left(Y_{jk}^{i+1} * P_{jk}\right) \leq \sum_{k=1}^{m}\left(Y_{jk}^{i+1} * F_{k}^{p'}\right) + V * \left(1 - \sum_{k=1}^{m}\left(Y_{jk}^{i} * X_{jk}^{p}\right)\right) \\ &+V * \left(1 - \sum_{k=1}^{m}\left(Y_{jk}^{i+1} * X_{jk}^{p'}\right)\right) \\ &\forall j \in J, \quad \forall i = 1 \ldots \left(nb_j - 1\right), \quad \forall p, p' = 1 \ldots np_k \end{aligned} \tag{6}$$

$$Qst_{k}^{pc} = \sum_{j=1}^{n}\left(X_{jk}^{p} * Qdem_{j}^{c}\right) \qquad \forall k \in M, \quad \forall p = 1 \ldots np_k, \quad \forall c \in C \tag{7}$$

$$\sum_{s=1}^{p} Qst_{k}^{sc} \leq \sum_{t=1}^{t\max} Z_{k}^{pt} * Qarr_{t}^{c} \qquad \forall k \in M, \quad \forall p = 1..np_k, \quad \forall c \in c, \quad \forall t \in T \tag{8}$$

$$F_k^p - \left(Tarr_t^c - \sum_{j=1}^{n} \left(P_{jk} * X_{jk}^p \right) \right) \geq V * \left(Z_k^{pt} - 1 \right) \tag{9}$$

$\forall k \in M, \quad \forall p = 1 \ldots np_k, \quad \forall c \in C, \quad \forall t \in T$

$$C_{\max} \geq F_k^{np_k} \qquad \forall k \in M \tag{10}$$

The constraint (2) determines that one position must be occupied by at most one operation of the job *j*; it means that a machine can execute only one operation at a time. The constraint (3) guarantees that an operation of the job *j* is assigned to one and only one position in a single machine at a time. The constraint (4) ensures that there is no overlap between positions. The constraint (5) ensures the precedence of jobs on the same machine (non-overlap between jobs). The constraint (6) makes sure that no operation starts processing before the accomplishment of the previous operation of the same job. The constraint (7) calculates the total amount of resources requested by a job assigned in each position. Constraint (8) indicates that the amount of resources must be less than or equal to the quantity arrived at the beginning of these positions. The constraint (9) makes the relationship between the time of resources arriving and the end of positions. Finally, the makespen is calculated by the constraint (10).

5 Approximate methods

In order to solve the studied problem; we adapted a meta-heuristic that is based on genetic algorithms further than four heuristics that are based on PRs. To show more of these methods we have chosen a problem of four jobs to be executed on four machines.

Table 3 presents the operational ranges of jobs as well as the quantity of resources to be consumed for each of these jobs. The arrival of the resources is one resource every five units, represented as: $Qarr_t^c = 1$ and $Tarr_t^c = 5$.

Table 3 Processing time and amount of resources required for jobs

		Machine (processing time)				$Qdem_j^c$
Jobs	J_1	M_3 (2)	M_1 (3)	M_2 (5)	M_4 (25)	10
	J_2	M_1 (15)	M_2 (10)	M_3 (20)	M_4 (8)	8
	J_3	M_3 (7)	M_2 (14)	M_1 (5)	M_4 (5)	4
	J_4	M_2 (9)	M_1 (4)	M_3 (3)	-	-

5.1 Heuristic methods

5.1.1 Job shop short processing time

For the Job shop short processing time (JSSPT) heuristic, it is assumed that the sequencing of the jobs on the machines is done according to the increase in the sum of the processing time for each job. In our example, we find that the sum processing time values of jobs J_1, J_2, J_3 and J_4 are respectively 35, 53, 31 and 16. Then, according to the heuristic JSSPT, the problem solution is defined by the following sequencing:

Table 4 Jobs order according to JSSPT

	M_1	M_2	M_3	M_4
Order 1	J_4	J_4	J_4	J_3
Order 2	J_3	J_3	J_3	J_1
Order 3	J_1	J_1	J_1	J_2
Order 4	J_2	J_2	J_2	-

Table 4 shows the order of jobs execution on machines. The order indicates that machines must first process job J_4 then J_3 after that J_1 and finally J_2. So, the $C_{\max}$ calculated from this sequencing is equal to *118* units of time.

5.1.2 Job shop longest processing time

Unlike the previous heuristic, the heuristic job shop longest processing time (JSLPT) requires that the jobs sequencing on each machine is done in a descending order of the sum corresponding to each job processing time. That is, the job execution on each machine must be started by the job having the smallest value of the sum of durations of these operations. Applying this heuristic on the same example gives us the order shown in Table 5. This provides a value of $C_{\max}$ which is equal to *120* units of time.

Table 5 Jobs order according to JSLPT

	M_1	M_2	M_3	M_4
Order 1	J_2	J_2	J_2	J_2
Order 2	J_1	J_1	J_1	J_1
Order 3	J_3	J_3	J_3	J_3
Order 4	J_4	J_4	J_4	-

5.1.3 Job shop short accumulation processing time

Job shop short accumulation processing time (JSSPTcum) is a heuristic that consists of ordering jobs in the increasing order of the cumulative processing time of jobs on each machine. For this heuristic, when a machine is released, the job chosen to be processed first is the one with the lowest accumulate value. This accumulate value is calculated by the sum of durations of all previous operations (that is, operations up to the operation to be performed on that machine). This technique aims to place at the beginning of the scheduling the jobs that consume the least execution time to be out of the machine. Table 6 shows the cumulative durations on each machine (according to the operations order). We can see, for example for the job J_3 which will be executed with its second operation on the machine M_2, that the accumulate value on this machine is equal to 21, i.e., the duration of the first operation (7) plus the duration of the second operation (14).

According to the heuristic JSSPTcum the scheduling of jobs for our example is mentioned in Table 7. The $C_{\max}$ obtained in this case is 118 units of time.

Table 6 Cumulative of processing time on machines

	M_1	M_2	M_3	M_4
J_1	5	10	2	35
J_2	15	25	45	53
J_3	26	21	7	31
J_4	13	9	16	-

Table 7 Jobs order according to JSSPTcum

	M_1	M_2	M_3	M_4
Order 1	J_1	J_4	J_1	J_3
Order 2	J_4	J_1	J_3	J_1
Order 3	J_2	J_3	J_4	J_2
Order 4	J_3	J_2	J_2	-

5.1.4 Job shop short resources consumption

The principle of the job shop short resources consumption (JSSRC) heuristic is to schedule jobs according to the increasing order of the quantity of resources required for each job. This heuristic aims to put the jobs that consume the least components at the beginning of the scheduling. Applying this heuristic on the same example gives the order shown in Table 8. The $C_{\max}$ obtained in this case is *135* units of time.

Table 8 Jobs order according to JSSRC

	M_1	M_2	M_3	M_4
Order 1	J_4	J_4	J_4	J_3
Order 2	J_3	J_3	J_3	J_2
Order 3	J_2	J_2	J_2	J_1
Order 4	J_1	J_1	J_1	-

5.2 Method based on genetic algorithms

This type of meta-heuristics is a research technique proposed by Holland in 1975 (Holland, 1992). Meta-heuristics serve to solve a wide range of difficult optimisation problems in a reasonable time, for which no more effective traditional methods are known. In a genetic algorithm, a representation must be chosen for the individuals in a population. Therefore, each solution is stored in an artificial chromosome represented by a code.

In fact, when using genetic algorithms, no information is required to solve problems. It is only necessary to provide a coding function of each solution in the form of genes, and to provide a function to evaluate the relevance of the obtained solution to this problem. For our problem, we have developed a genetic algorithm that has the following structure:

5.2.1 Coding

The selection of a string format for the individuals is a first and a very important step to a successful implementation of a genetic algorithm. Therefore, we try to find an efficient coding of individuals, which satisfies all constraints. We proposed to use a representation that is based on the jobs execution order on machines. This means, each individual of the population is illustrated by a $n \times M$ matrix. However, the columns represent the machines and the lines represent the jobs order in each machine. The example shown in Table 9 presents the chromosome of four jobs and four machines.

Table 9 Chromosome illustration

M_1	M_2	M_3	M_4
J_4	J_4	J_3	J_2
J_1	J_2	J_2	J_1
J_2	J_3	J_4	J_3
J_3	J_1	J_1	-

In this example, we can notice that the first job that will be processed on the machine M_1 is the job J_4 followed by the job J_1, then the job J_2 and finally the job J_3 (same thing for the other machines).

5.2.2 The initial population

It is the set of chromosomes representing different solutions of the problem. At the beginning of the study, we proposed that all chromosomes be generated in a random way. Each chromosome is represented by a random assignment of a job to a position on the machine.

5.2.3 Fitness function

Using GA in an optimisation problem requires the assignment of a fitness value to a chromosome. In many studies, this value consists in the value of the objective function (Atabaki and Mohammadi, 2017; Yazdian et al. (2017). As the objective function of the studied problem is the minimisation of the total execution time [defined in equation (1)], the best solutions are those resulting in the lower $C_{\max}$.

5.2.4 Selection

The selection mechanism adopted here is based on a two-player tournament to select parents for reproduction. We use the elitist selection technique and we choose the reproduction chromosomes building on the evaluation of their fitness function.

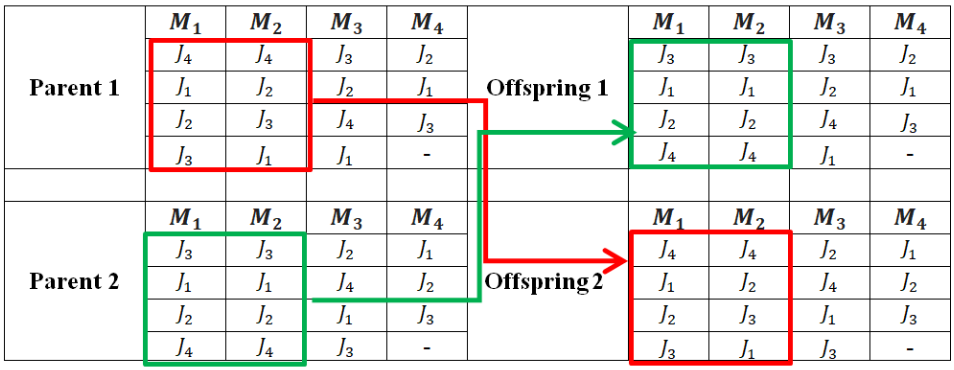
5.2.5 Crossover

In this step, and as shown in Figure 4, we select two random chromosomes and apply a cross of two columns to produce two new chromosomes.

The choice of two columns (machines) for crossing is based on two reasons:

- To generate two new chromosomes, because if we choose more or less than two machines it is very possible to fall into the same father chromosomes.
- To ensure the eligibility of the solutions, because if we choose lines, the probability that the generated chromosomes present inadmissible solutions is very high.

Figure 4 Crossing way (see online version for colours)



5.2.6 *Mutation*

The mutation in the genetic algorithm method is carried out to maintain the diversity of the population through the disturbances caused in a solution. In our problem, mutation was not taken into account.

5.2.7 *Stop criterion*

The stop criterion used here refers to the number of performed iterations.

6 Experimental results

To evaluate the performances of the adapted GA as well as the four previously proposed heuristics, we used them to solve several examples with different sizes. For this purpose, we have proposed a code on MATLAB that allows us to calculate the makespan ($C_{\max}$) further than finding the best solution.

Indeed, we proposed to study ten different examples for each size. These examples are generated in a random way through a subroutine which randomly defines the operating ranges, the amount of jobs requested resources as well as the amount and the resources arrival time.

To examine the effectiveness of the proposed approaches, it was proposed that two important criteria be considered. The first is the relative deviation percentage ratio (RE) defined by equation (11). This ratio determines the percentage reconciliation of the makespan value calculated by a given approach ($C_{\max}(app)$) and the best value found by all approaches ($C_{\max}(best)$).

$$\text{RE} = \frac{C_{\max}(app) - C_{\max}(best)}{C_{\max}(app)} \times 100 \tag{11}$$

The second criterion is the CPU time, which shows the time elapsed during the execution of certain algorithms. This criterion is used to test the efficiency of approaches and to find the best one that gives satisfactory solutions in a reasonable time.

6.1 Obtained results for small instances

We studied three small size problems; (2, 3 and 4 jobs to be performed on four machines), through which, we made a comparison between the proposed approaches and the exact method defined by the mathematical model. Indeed, to obtain the optimal solution using an exact method (EX), we had programmed and optimised the mathematical model by CPLEX. In this section, processing time is generated randomly between 1 and 30 units of time.

Table 10 Obtained results for small instances

*J*M*	*JSSPT*		*JSLPT*		*JSSPTcum*	
	RE	*CPU time*	*RE*	*CPU time*	*RE*	*CPU time*
2*4	28,60	0.07	28.86	0.05	0.94	0.11
3*4	40.23	0.07	45.01	0.06	5.05	0.10
4*4	35.83	0.07	36.03	0.06	0.00	0.13
*J*M*	*JSSRC*		*GA*		*EX*	
	RE	*CPU time*	*RE*	*CPU time*	*RE*	*CPU time*
2*4	25.23	0.08	10.87	1.86	0	0.358
3*4	41.37	0.07	25.89	9.56	0	0.655
4*4	30.62	0.09	21.32	0.54	-	-

Through the results presented in Table 10, and by comparing the RE values calculated through the application of each approach, we notice that the solutions provided by the JSSPTcum heuristic are very close to the optimum, this is defined by the percentages of 0.94% and 5.05% of the two problems 3*4 and 3*4, respectively. This approximation to the optimum demonstrates very well the efficiency of this heuristic. In addition, we deduce that the exact method is unable to solve the 4*4 problem in a reasonable time, which demonstrates the complexity of the studied problem.

6.2 Obtained results for middle size problem instances

In this part, problems of 8, 10 and 15 jobs are selected. The variation of operations processing time follows a uniform law between 1 and 50.

Through the results mentioned in Table 11, it is well-noticed that the JSSPTcum heuristic, which is based on the choice of the small value of the cumulated processing time, gives the best solutions when compared with the solutions found by the other approaches. This is defined by the null value of RE calculated by the application of JSSPTcum. On the other hand, the CPU time values presented in the table also demonstrate the efficiency of JSSPTcum and the other heuristics compared to the GA in terms of time. In addition, the exact method in this part of expertise remains unable to find a solution within a reasonable time, which also supports the use of other approaches.

Table 11 Obtained results for middle size problem instances

*J*M*	*JSSPT*		*JSLPT*		*JSSPTcum*	
	RE	*CPU time*	*RE*	*CPU time*	*RE*	*CPU time*
8*4	44.59	0.07	45.42	0.06	0.00	0.13
10*4	45.38	0.08	43.98	0.05	0.00	0.07
15*4	48.08	0.06	49. .01	0.06	0.00	0.07
*J*M*	*JSSRC*		*GA*		*EX*	
	RE	*CPU time*	*RE*	*CPU time*	*RE*	*CPU time*
8*4	44.27	0.09	21.66	0.54	-	-
10*4	44.08	0.05	21.93	17.35	-	-
15*4	50.61	0.06	32.21	26.31	-	-

6.3 Obtained results for larger size problem instances

Problems selected in this part are those of size 20, 50 and 100 jobs which are to be performed on four machines. As in the previous part, the processing time follows a uniform law between 1 and 50.

According to Table 12, we notice that the best solutions are obtained also by the application of the heuristic JSSPTcum, which validates the performance of this proposed method.

Table 12 Obtained results for larger size problem instances

*J*M*	*JSSPT*		*JSLPT*		*JSSPTcum*		*JSSRC*		*GA*	
	RE	*CPU time*	*RE*	*CPU time*	*RE*	*CPU time*	*RE*	*CPU time*	*RE*	*CPU time*
20*4	55.38	0.05	54.84	0.04	0.00	0.05	54.50	0.04	36.15	29.76
50*4	57.73	0.14	58.72	0.12	0.00	0.12	58.68	0.11	51.26	126.60
100*4	58.48	0.27	55.92	0.21	0.00	0.26	58.11	0.21	53.85	319.95

Again, it can be noted here that the closest BR value to the best value (value given by JSSPTcum) is given by the GA in the three studied problems. This shows that the GA is the most efficient approach compared to the other heuristics.

7 Conclusions

In the field of scheduling, researchers aim to get much closer to reality by adding one or more constraints to each of the problems they are studying. In this paper, we dealt with a JSSP by taking into account the non-renewable resources availability constraint. This problem is strongly encountered in the FMSs. The FMS includes one or more assembly stations where some jobs require an amount of CRs to be performed. Our work is based on a real case study represented by iCIM 3000 FMS system of the MELT laboratory. To solve the scheduling problem existing in this system, we developed a mathematical model that is based on the integer mixed linear programming. The aim is to find the optimal jobs sequencing by taking into account the required resources in each assembly operation. Due to this problem complexity as well as the large number of variables and constraints, two different resolution methods have been proposed. We firstly adapted a meta-heuristic that is based on the genetic algorithms, and secondly, we proposed four heuristics that appeals to PRs. Several simulations have been performed on instances of different sizes. The obtained results show that the JSSPTcum heuristic, based on cumulative processing time, gives solutions that are very close to the exact method ones in case of small instances. However, the JSSPTcum surpasses the AG and the other approaches in case of large instances. These results demonstrate the effectiveness of the proposed JSSPTcum approach. However, the work carried out in this document deserves more attention, and it should be enriched by proposing other practical solutions in the following areas:

- The conducted research focused on minimising makespan in job shop environment. However, many objectives can be well-justified in practice such as maximum tardiness minimisation. Consequently, it is noticed that a multiple objective optimisation study is necessary and represents an important future direction.
- Furthermore, In the FMS studied here, jobs require non-renewable resources only on one station (machine). However, in many other FMSs, this is not always the case. For this, an improvement of the proposed method is essential to cover this generalisation in terms of the number of machines using CRs.
- The experimental results are strongly dependent on the structure of the studied system. Thus, the proposed algorithm may not lead to good results if certain operating conditions change such as the arrival function of the non-renewable resources or the speed of the machines. To overcome these drawbacks, we propose to address this problem with other criteria and constraints, in order to analyse and better understand the system behavior.
- Finally, to improve the found results using the JSSPTcum heuristic, it is proposed to integrate this heuristic into the initial population generation of the genetic algorithm in order to further improve the best solution search.

References

Afshar, M.R., Shahhosseini, V. and Sebt, M.H. (2019) 'A genetic algorithm with a new local search method for solving the multimode resource-constrained project scheduling problem', *International Journal of Construction Management*, pp.1–9.

Atabaki, M.S. and Mohammadi, M. (2017) 'A genetic algorithm for integrated lot sizing and supplier selection with defective items and storage and supplier capacity constraints', *International Journal of Operational Research*, Vol. 28, No. 2, pp.183–200.

Belkaid, F., Yalaoui, F. and Sari, Z. (2016a) 'An efficient approach for the reentrant parallel machines scheduling problem under consumable resources constraints', *International Journal of Information Systems and Supply Chain Management*, Vol. 9, No. 3, pp.1–25.

Belkaid, F., Yalaoui, F. and Sari, Z. (2016b) 'Investigations on performance evaluation of scheduling heuristics and metaheuristics in a parallel machine environment', in Talbi, E. et al. (Eds.): *Metaheuristics for Production Systems*, pp.191–222, Springer, Cham.

Carlier, J., Moukrim, A. and Sahli, A. (2018) 'Lower bounds for the event scheduling problem with consumption and production of resources', *Discrete Applied Mathematics*, Vol. 234, pp.178–194.

Caumond, A. (2006) *Le problème de job shop avec contraintes: modélisation et optimisation*, Doctoral dissertation, Blaise-Pascal University, France.

Dabaha, A., Bendjoudi, A., AitZaib, A., El-Bazc, D. and Taboudjemat, N.N. (2018) 'Hybrid multi-core CPU and GPU-based B&B approaches for the blocking job shop scheduling problem', *Journal of Parallel and Distributed Computing*, Vol. 117, pp.73–86.

Delgado Sobrino, D.R. and Košťál, P. (2013) 'A few analysis and customization issues of a new iCIM 3000 system: the case of the material flow, its complexity and a few issues to improve', *Academic Journal of Manufacturing Engineering*, Vol. 11, No. 4, pp.36–41.

Delgado Sobrino, D.R., Košťál, P. and Vavruška, J. (2014) 'On the analysis and customization of an iCIM 3000 system: a take on the material flow, its complexity and a few general issues to improve', *Applied Mechanics and Materials*, Vol. 474, pp.42–48.

Deng, G., Su, Q., Zhang, Z., Liu, H., Zhang, S. and Jiang, T. (2020) 'A population-based iterated greedy algorithm for no-wait job shop scheduling with total flow time criterion', *Engineering Applications of Artificial Intelligence*, Vol. 88, pp.103–369.

Đurasević, M. and Jakobović, D. (2018) 'Comparison of ensemble learning methods for creating ensembles of dispatching rules for the unrelated machines environment', *Genetic Programming and Evolvable Machines*, Vol. 19, Nos. 1–2, pp.53–92.

Fattahi, P., Keneshloo, S. and Daneshamooz, F. (2019) 'A hybrid genetic algorithm and parallel variable neighborhood search for jobshop scheduling with an assembly stage', *International Journal of Industrial Engineering and Production Research*, Vol. 30, No. 1, pp.25–37.

Gondran, M., Kemmoé-Tchomté, S., Lamy, D. and Tchernev, N. (2016) 'Une approche bi-objectif au problème du job-shop sous contrainte de pics de consommation énergétique', Paper presented at the *17th ROADEEF International Conference*, France, 10–12 February.

Gong, G., Deng, Q., Chiong, R., Gong, X. and Huang, H. (2019) 'An effective memetic algorithm for multi-objective job-shop scheduling', *Knowledge-Based Systems*, Vol. 182, p.104840.

Grabowski, J. and Janiak, A. (1987) 'Job-shop scheduling with resource-time models of operations', *European Journal of Operational Research*, Vol. 28, No. 1, pp.58–73.

Graham, R.L., Lawler, E.L., Lenstra, J.K. and Kan, A.R. (1979) 'Optimization and approximation in deterministic sequencing and scheduling: a survey', *Annals of Discrete Mathematics*, Vol. 5, pp.287–326.

Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, Cambridge.

Kassu, J. and Eshetie, B. (2015) 'Job shop scheduling problem for machine shop with shifting heuristic bottleneck', *Global Journal of Research in Engineering*, Vol. 15, No. 1, pp.21–25.

Kemmoé, S., Lamy, D. and Tchernev, N. (2015) 'A job-shop with an energy threshold issue considering operations with consumption peaks', *International Federation of Automatic Control*, Vol. 48, No. 3, pp.788–793.

Kumar, H., Kumar, P. and Sharma, M. (2019) 'A genetic algorithm for a flow shop scheduling problem with breakdown interval, transportation time and weights of jobs', *International Journal of Operational Research*, Vol. 35, No. 4, pp.470–483.

Liu, S.Q., Kozan, E., Masoud, M., Zhang, Y. and Chan, F.T. (2018) 'Job shop scheduling with a combination of four buffering constraints', *International Journal of Production Research*, Vol. 56, No. 9, pp.3274–3293.

Mejía, G. and Pereira, J. (2020) 'Multi objective scheduling algorithm for flexible manufacturing systems with Petri nets', *Journal of Manufacturing Systems*, Vol. 54, pp.272–284.

Meng, L., Zhang, C., Shao, X. and Ren, Y. (2019) 'MILP models for energy-aware flexible job shop scheduling problem', *Journal of Cleaner Production*, Vol. 210, pp.710–723.

Muritiba, A.E., Rodrigues, C.D. and da Costa, F.A. (2018) 'A Path-Relinking algorithm for the multi-mode resource-constrained project scheduling problem', *Computers & Operations Research*, Vol. 92, pp.145–154.

Nagata, Y. and Ono, I. (2018) 'A guided local search with iterative ejections of bottleneck operations for the job shop scheduling problem', *Computers and Operations Research*, Vol. 90, pp.60–71.

Ozturk, G., Bahadir, O. and Teymourifar, A. (2019) 'Extracting priority rules for dynamic multi-objective flexible job shop scheduling problems using gene expression programming', *International Journal of Production Research*, Vol. 57, No. 10, pp.3121–3137.

Rajendran, S., Rajendran, C. and Leisten, R. (2017) 'Heuristic rules for tie-breaking in the implementation of the NEH heuristic for permutation flow-shop scheduling', *International Journal of Operational Research*, Vol. 28, No. 1, pp.87–97.

Rashid, R., Arani, S.D., Hoseini, S.F. and Omran, M.M. (2018) 'A new supply chain network design approach, regarding retailer's inventory level and supplier's response time', *International Journal of Operational Research*, Vol. 31, No. 4, pp.421–241.

Salido, M.A., Escamilla, J., Barber, F., Giret, A. and Tang, D. (2016) 'Energy efficiency, robustness, and makespan optimality in job-shop scheduling problems', *AI EDAM*, Vol. 30, No. 3, pp.300–312.

Seidgar, H., Rad, S.T. and Shafaei, R. (2017) 'Scheduling of assembly flow shop problem and machines with random breakdowns', *International Journal of Operational Research*, Vol. 29, No. 2, pp.273–293.

Sotskov, Y.N., Matsveichuk, N.M. and Hatsura, V.D. (2020) 'Two-machine job-shop scheduling problem to minimize the makespan with uncertain job durations', *Algorithms*, Vol. 13, No. 1, p.4.

Tabrizi, B.H., Ghaderi, S.F. and Haji-Yakhchali, S. (2019) 'Net present value maximisation of integrated project scheduling and material procurement planning', *International Journal of Operational Research*, Vol. 34, No. 2, pp.285–300.

Toker, A., Kondakci, S. and Erkip, N. (1991) 'Scheduling under a non-renewable resource constraint', *Journal of the Operational Research Society*, Vol. 42, No. 9, pp.811–814.

Toker, A., Kondakci, S. and Erkip, N. (1994) 'Job shop scheduling under a non-renewable resource constraint', *Journal of the Operational Research Society*, Vol. 45, No. 8, pp.942–947.

Yazdian, S.A., Shahanaghi, K. and Naini, S.G. (2017) 'Integrated marketing and operational decisions in remanufacturing of end-of-life products: model and hybrid solution algorithms', *International Journal of Operational Research*, Vol. 29, No. 1, pp.34–66.

Zeng, C., Tang, J., Fan, Z.P. and Yan, C. (2020) 'Scheduling of a job-shop problem with limited output buffers', *Engineering Optimization*, Vol. 52, No. 1, pp.53–73.

Zhang, J., Ding, G., Zou, Y., Qin, S. and Fu, J. (2019) 'Review of job shop scheduling research and its new perspectives under Industry 4.0', *Journal of Intelligent Manufacturing*, Vol. 30, No. 4, pp.1809–1830.

Zhou, D.N., Cherkassky, V., Baldwin, T.R. and Olson, D.E. (1991) 'A neural network approach to job-shop scheduling', *IEEE Transactions on Neural Networks*, Vol. 2, No. 1, pp.175–179.

Zhou, Y.D., Shin, J.H. and Lee, D.H. (2019) 'Loading and scheduling for flexible manufacturing systems with controllable processing times', *Engineering Optimization*, Vol. 51, No. 3, pp.412–426.