Simulated annealing approach to solve dual resource constrained job shop scheduling problems: layout impact analysis on solution quality

Maurizio Faccio*

Department of Management and Engineering,
University of Padova,
Stradella San Nicola, 3 36100 Vicenza, Italy
E-mail: maurizio.faccio@unipd.it
*Corresponding author

Jana Ries

Portsmouth Business School,
University of Portsmouth,
Portland Street PO1 3DE,
Portsmouth, Hampshire, UK
E-mail: jana.ries@port.ac.uk

Nicola Saggiorno

Department of Management and Engineering,
University of Padova,
Stradella San Nicola, 3 36100 Vicenza, Italy
E-mail: nicola.saggioro@gmail.com

Abstract: Real-world manufacturing systems are operating subject to a substantial level of resource constraints. One characteristic model that considers the combination of human and machine resource constraints is called dual resource constrained (DRC). In this context a number of machines n_{mach} is managed by a selection of operators n_{op}, with typically $n_{op} \leq n_{mach.}$. A real life case study for an Italian manufacturing company is introduced that uses a set of identical parallel machines being operated by a set of operators. Each job is scheduled to one machine with corresponding loading and unloading process times. A simulated annealing approach is proposed to solve the DRC job shop scheduling problem. A sensitivity analysis is conducted for a selection of algorithm-specific parameters used to solve characteristic DRC layouts. Being characteristic for the just-in-time (JIT) production environment, the high variability in job times has also been taken into account. The results show that the selected layout $n_{mach.}/n_{op}$ ratio strongly influences the production system performance. The impact of the ratio of constrained resources has been analysed for different layouts, showing that simulated annealing performs better for single resource constrained problems while also demonstrating that this trend is not symmetrical for different layouts, either operator or machine constrained.

Keywords: scheduling; dual resource constrained; DRC; resources utilisation; JIT production; job list variability.

Reference to this paper should be made as follows: Faccio, M., Ries, J. and Saggiorno, N. (xxxx) 'Simulated annealing approach to solve dual resource constrained job shop scheduling problems: layout impact analysis on solution quality', *Int. J. Mathematics in Operational Research*, Vol. X, No. Y, pp.xxx–xxx.

Biographical notes: Maurizio Faccio, Assistant Professor in Industrial Plants and Logistics at Padua University.

Jana Ries, Senior Lecturer in Quantitative Methods at Portsmouth Business School, University of Portsmouth

Nicola Saggiorno, Engineer and Researcher in Industrial Plants and Logistics at Padua University.

1 Introduction

Increasing economic pressure has forced companies to develop and amend production strategies that maximise manufacturing capacities while minimising corresponding costs. In contrast, integrated approaches have led to strategies driven by small batch sizes and an increasing level of product mix complexity that subsequently minimise in-process inventory costs. This leads to production systems continuously transforming into just-in-time (JIT) environments.

The task of scheduling resources forms the key challenge to increase efficiency and optimise capacity utilisation in production environments by reducing machine running times and consequently increasing profitability (Vinod and Sridharan, 2008). Resource scheduling is found in various applications including project scheduling (Najid and Arroub, 2010), processor scheduling (Venkataramana and Srinivasa Raghavan, 2010), maintenance scheduling (Hadidi et al., 2011), mixed model assembly scheduling (Azzi et al., 2012a, 2012b; Finetto et al., 2013), and production scheduling (Sharma et al., 2013). In case of different types of machines, the complexity of the problem increases due to the required sequencing of the jobs (Balin, 2012, Azzi et al., 2012c). With operators being in charge of loading and unloading jobs onto available machines, a significant challenge occurs with regards to improving efficiency in production systems. It requires the integration of various constraints to result in an optimal schedule that minimises idle times and reduces the overall makespan. Particularly challenging is the consideration of loading and unloading times between jobs that may change due to being e.g., job-specific. Subsequently, this may result in a higher risk of idle times within the sequence. Moreover, dual resource constrained (DRC) systems require two different decision making stages, namely identifying the job sequence and assignment of a set of operators (Xu et al., 2011). The interaction between both forms is an additional technical challenge in solving those systems. While the general form of job shop scheduling problems has been studied for more than 50 years (Johnson, 1954), it is the single resource constrained (SRC) problem which has received most consideration in the literature. However, most real-life production systems incorporate resource constraints in form of both machines and operators, transforming the real-life nature of job shop scheduling into a DRC system.

Treleven (1989) outlines a set of approaches to DRC problems categorising the problem formulations into design and operating decisions. Hottenstein and Bowman (1998) give a more in-depth coverage of sixteen different DRC scenarios. The authors outline five key DRC objectives namely worker flexibility, when to transfer operators, where to transfer an operator, queue discipline (job dispatching) and the cost of transferring workers. Those dimensions are reconsidered by Xu et al. (2011).

The *job release mechanism* (or planning system) defines the pattern of work released onto the shop floor. Some authors propose job release approaches that reduce the amount of variance by smoothing the load received (Araz and Salum, 2008; Salum and Araz, 2009). *Worker flexibility* is a typical element of DRC problems where operators move between workstations to process jobs when required. Worker flexibility is evaluated by an index which reflects the number of different machines each worker can operate. Kher and Fry (2001) outlined that an increase in worker flexibility positively affects the shop performance. Similarly, machine flexibility in terms of the set of jobs that can be performed may be considered. *Worker assignment* indicates when a worker should be transferred to a different work centre. The mode of control may be centralised (a worker is allowed to move to another work centre even if there are still some jobs waiting to be processed in the current one) or decentralised (no operator can move to a different work centre until all jobs are processed in the current one). It has been shown that the decentralised approach is to be preferred if transfer cost is taken into consideration (Gunther, 1981; Malhotra and Kher, 1994), while the centralised strategy performs better in the case of workers with different flexibility indices (Salum and Araz, 2009).

Job dispatching determines the order of jobs to be processed at a specific machine. This is the integral decision to be made in a scheduling environment. Dispatching rules are very often used because of simplicity. First-come first-served (FCFS), the early due date (EDD) or shortest processing time (SPT) use the scheduling system information only, while more complex scheduling rules can be implemented taking into account also future system information in order to improve the quality of the schedule. Fredendall et al. (1996) developed a set of such rules for order release, job dispatching and labour assignment, showing a significant increase in job shop performance. With respect to DRC systems, ElMaraghy (2000) tested six different dispatching rules to solve a DRC while Liao and Lin (1998) analysed the impact of different job dispatching and labour assignment rules on job shop performance. Sammarco et al. (2013) studied the effects of different rules and system parameters on a DRC flow shop behaviour and Kher (2000) analysed the problem of a DRC job shop environment with jobs of different priority levels.

In addition to decision rules, DRC scheduling systems have been addressed using exact, heuristic and simulation approaches. The main drawback of exact methods is the high computational time required to reach the optimal solution for more complex real-life scheduling environments. Similarly, simulation methods are time-consuming. This supports the use of heuristic and meta-heuristics procedures that are proposed in DRC scheduling as they are capable of providing a feasible solution in a reasonable amount of time. Li et al. (2011) presented a metaheuristic to minimise the total completion time with permanently present operators in a flow shop production system. ElMaraghy (2000) developed a genetic algorithm where multiple performance measures and dispatching rules were taken into account. Tao et al. (2007) further developed the GA approach to solve DRC problems combining it with a simulated annealing (SA) algorithm in order to better deal with changing job shop environments, taking into account unpredictable events such as machine breakdowns or sudden urgent orders from vital customers. SA and neighbourhood search were proposed by Ruiz-Torres and Centeno (2008) as a tool to tackle the problem of minimising the number of tardy jobs in a flowshop with secondary resources.

While several sets of dispatching rules have been outlined in the literature, metaheuristic approaches have shown to be promising with respect to being able to integrate the set of problem-related constraints while being able to find good solutions in a reasonable amount of time. Metaheuristics are known to perform according to algorithm-specific parameters. It is therefore of further interest to identify the level of robustness of such a proposed metaheuristic in the problem context. Structural changes in instances may interact with the calibration of a metaheuristic (Ries et al., 2012) and hence it is important to understand the impact of potential changes in the problem with regards to heuristic performance.

In contrast to the existing literature which uses SA as a decision algorithm to choose amongst a set of scheduling rules depending on systems conditions, the paper proposes the implementation of a SA concept for the DRC. The proposed approach considers a production system characterised by a JIT strategy with a high variability in jobs time, ensuring that the scheduling procedure may be applied to various layouts considering different combinations of the number of machines and operators.

The remainder of the paper is organised as follows. Section 2 presents the DRC problem formulation and the selected meta-heuristic, while Section 3 describes a case study and the computational experiments. Section 4 presents the layout analysis and Section 5 shows the impact analysis of the $n_{mach.}/n_{op}$ ratio. Section 6 analyses idle times and costs while conclusions are reported in Section 7.

2 Problem formulation and SA

The considered production system is composed by a set of machines and human operators. A set of different jobs has to be performed with job times being composed of running, unloading and loading times performed at the workstation by an operator.

While it is important to understand the best daily job schedule, it is also vital to get an indication of the best job shop configuration (or layout) due to the high variability in the job shop environment. The model considers a given job shop layout composed of a number of identical machines n_{mach} and a number of workers n_{op} with a 100% flexibility level.

The following assumptions/notations are considered:

- The considered system is composed of a number of identical machines n_{mach} and workers n_{op}.
- Each operator can operate all machines.
- Each operator can perform only one job at a time.
- Each machine can perform only one job at a time.
- Holding costs are disregarded.

- Travel times between different machines and inspection times are not taken into consideration.
- Each job can be performed on all machines.
- For the considered time period, a given job list is to be scheduled and forms the solution of the problem. Each job i is characterised by a running time R_i that represents the processing time, a loading time L_i and an unloading time U_i representing the time for loading and unloading the job, respectively.

The solution framework is divided into two phases:

1. the design phase that determines the optimal ratio n_{mach}/n_{op} for a given list of jobs
2. the scheduling phase which determines the optimal job assignment sequence for each machine and operator for a given list of jobs and its corresponding ratio n_{mach}/n_{op}.

2.1 Design phase: n_{mach}/n_{op} ratio

A ratio r is defined as $r = n_{mach}/n_{op}$, with n_{jobs} different jobs being performed within a given day on a set of n_{mach} available machines and each job being characterised by R_i, L_i, U_i. This ratio characterises the layout of the production environment by identifying the number of machines per operator.

This design phase allows determining the layout that will result in the optimal combination of resources given the jobs to be performed. Subsequently, the best ratio r will allow for all jobs to be processed while minimising the risk of resources being idle.

The best ratio r has to satisfy the following condition:

$$\frac{n_{mach}}{n_{op}} - 0.5 \le \frac{\sum_{i=1}^{n_{jobs}} \left(L_i + R_i + U_i\right)}{\sum_{i=1}^{n_{jobs}} \left(L_i + U_i\right)} \le \frac{n_{mach}}{n_{op}} + 0.5, \text{ with } n_{mach} \in Z^+, n_{op} \in Z^+ \tag{1}$$

It allows defining the number of machines and operators to use. If the given job list correspond to a considered production period T, the minimum number of necessary machines to perform the whole job list in T is:

$$n_{mach}^* = \frac{\sum_{i=1}^{n_{jobs}} \left(L_i + R_i + U_i\right)}{T} \tag{2}$$

The corresponding minimum number of operators n_{op}^* to perform the given job list is easily derived by (1) and (2).

2.2 Scheduling phase: SA

Let Ω be the solution space, where every solution Ω is a particular permutation of n_{jobs} jobs, and let f: $\Omega \rightarrow r$ be the objective function to be minimised. The goal is to find the optimal solution ω^* such that $f(\omega^*) < f(\omega)$ for every $\omega \in \Omega$.

An initial solution ω is randomly generated and becomes the current solution $\omega^{current}$. A neighbourhood $N(\omega^{current}) = \{\omega_1', \omega_2', .., \omega_m'\}$ is defined as a set of neighbour solutions of $\omega^{current}$. If the objective function value of a new solution in $N(\omega^{current})$ has improved compared to $f(\omega^{current})$, the new solution becomes the current one. Otherwise the new solution is accepted based on a probability such that:

$$P\left(accepting\ \omega'\ as\ new\ solution\right) = \begin{cases} e-\left[f(\omega')-f(\omega)\right]/t_k & \text{if } f(\omega')-f(\omega)>0 \\ 1 & \text{if } f(\omega')-f(\omega)\leq 0 \end{cases} \tag{3}$$

where t_k is a so-called temperature value at iteration k. The algorithm is repeated for a given number of iterations until it terminates.

The implementation of SA consists of two iterative loops, an internal and an external one. The latter uses a cooling schedule: given a number of steps for the cooling schedule (i.e., number of iterations for the external loop) k_{max}, a vector $t(k)$, $k = 1, 2, \ldots..k_{max}$ is defined as the cooling schedule. A starting temperature value T_0 and a final value T_{end} must be chosen such that

$$t(0) = T_0; \quad t\left(k_{max}\right) = T_{end} \tag{4}$$

The initial value T_0 should be high enough that nearly any change of solution is accepted, while the last value T_{end} is set close to zero, so that worse solutions are no longer accepted. With vector $t(k)$ representing a cooling schedule, it is required that

$$t(k) > t(k+1), \quad \text{for } k = 1,2,\ldots\ldots k_{max}-1 \tag{5}$$

At any iteration of the cooling schedule, the algorithm runs an internal loop. The number of iterations *it* is kept constant for every $t(k)$, $k = 1, 2, \ldots..k_{max}$.

Hence, the total number of iterations run by the algorithm is

$$tot\ iter = k_{max} \cdot it \tag{6}$$

The problem considers a single objective function $f(\omega)$ with the aim to minimise the makespan of the final assignment ω of the set of jobs to a set of machines. The SA implementation uses a memory-based approach by remembering the best solution found during the run of the algorithm. The current solution is thereby defined as $\omega^{bestsofarmksp}$, which is initialised as the worst case scenario makespan. Any time a solution ω' results in a shorter makespan than the current solution $\omega^{current}$, it is accepted and $\omega^{current} = \omega'$.. In this case, the original $\omega^{current}$ is not automatically deleted: if $f(\omega^{current}) < f(\omega^{bestsofar})$, $\omega^{current}$ is stored as $\omega^{bestsofar}$ solution. The final solution is chosen to be the best solution overall. SA has been implemented in MATLAB.

2.3 Neighbourhood functions for SA

The neighbourhood $N(\omega^{current})$ is created using a *swap* strategy: two jobs of the current solution are selected and swapped, leaving the others unchanged. Two swap operators are introduced, swapping jobs according to the operator chart (Figure 1) and swapping jobs according to the machine chart (Figure 2).

Figure 1 Example of operator-based (three operators) neighbourhood strategy (operators chart) (see online version for colours)

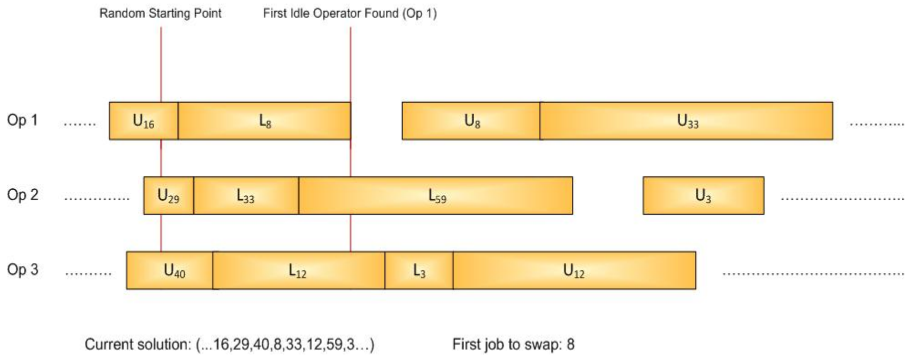
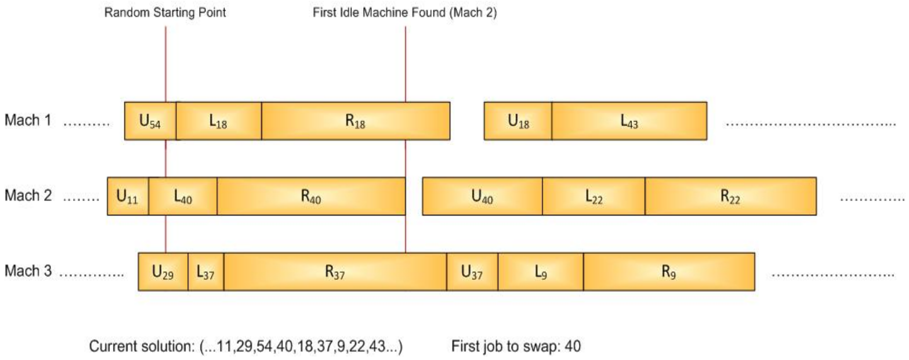


Figure 2 Example of machine-based (three machines) neighbourhood strategy (machines chart) (see online version for colours)



Given a random starting job, two swap strategies are being considered with respect to choosing the second job to be swapped with:

- selecting the next job in the solution
- random selection.

During one iteration the algorithm generates four new solution candidates using all four swap operators. The best candidate is accepted as new solution and compared to the current one. Preliminary testing has shown that a range of operators leads to better results than applying a single one only. Furthermore, it is appropriate to introduce both machine-based and operator-based functions into SA, as the algorithm is more likely to perform well regardless of which one is the critical resource (machines or operators).

3 Case study and computational experiments

The considered case study is based on an Italian manufacturing company that produces mechanical components, see Figure 3.

Figure 3 Case study production environment (see online version for colours)

Every day, a list of jobs needs to be performed by three operators on nine parallel identical machines (work centres) in a shop floor. Due to a JIT production strategy, the job release mechanism to the shop floor continuously changes.

The base case scenario uses nine identical parallel machines (m_1, m_2, …. m_9), operated by three operators (o_1, o_2, o_3) that perform a list of n_{jobs} jobs every day. Figure 4 summarises the case study layout.

In order to understand the impact of the layout on the performance of the shop floor, a set of other layouts has been tested (Table 1). The layouts have been selected according to different common industrial scenarios. Consequently, the case study represents a particular case for the tested computational experiments.

Table 1 Tested layouts

n_{mach}	n_{op}	r	n_{jobs}
9	3	3	60
15	5	3	100
6	2	3	40
8	4	2	60
8	2	4	35

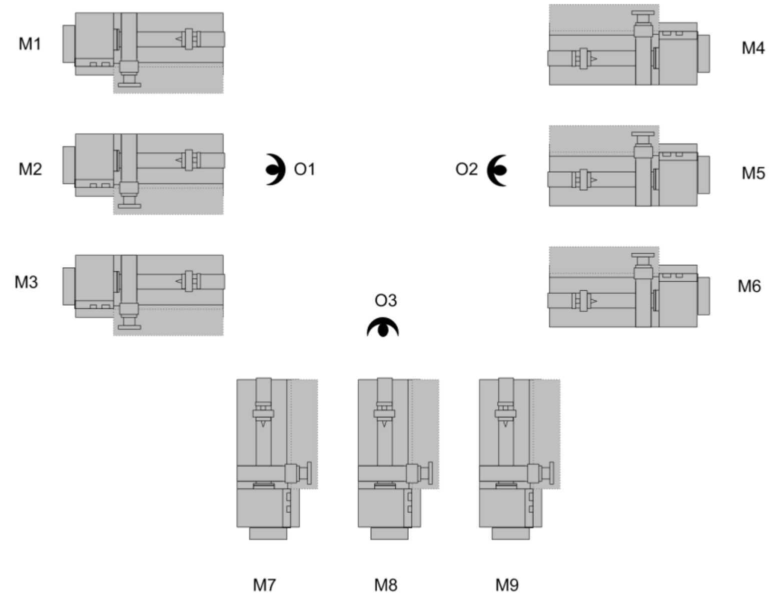
The job list has been generated according to the company case conditions as follows:

$$R_{min} = 10\,\text{min} \le R_i \le 80\,\text{min} = R_{max} \quad \forall i \tag{7}$$

$$L_{min} = 2\,\text{min} \le L_i \le \rho \cdot R_i = L_{i,max} \quad \forall i \tag{8}$$

$$U_{min} = 2\,\text{min} \le U_i \le \rho \cdot R_i = U_{i,max} \quad \forall i \tag{9}$$

Figure 4 Case study layout



Running, loading and unloading times are uniformly distributed according to a set of intervals, shown in equations (7) to (9), respectively. This is most appropriate to represent the duration of the set of jobs to be processed in a real-life scenario. Table 1 reports the list of layouts and corresponding scenarios tested including $r = n_{mach}/n_{op}$.

Performance is measured as follows in form of computational time and solution quality. Times are expressed in minutes. The considered time frame is one production shift, i.e., eight hours.

The machines' lower bound for each scenario is determined as follows:

$$LB_m = \left\lceil \frac{\sum_{i=1}^{N_{jobs}} \left(L_i + R_i + U_i \right)}{N_{mach}} \right\rceil \tag{10}$$

Considering the operators' availability, the lower bound is given by the total operator time needed divided by the number of operators

$$LB_O = \left\lceil \frac{\sum_{i=1}^{N_{jobs}} \left(L_i + U_i \right)}{N_{op}} \right\rceil \tag{11}$$

The final lower bound for a job list will therefore be

$$LB = max\left(LB_m, LB_O \right) \tag{12}$$

Hence, solution quality is defined as the excess between the makespan of the solution and the lower bound.

$$\Delta\%LB = \frac{MKSP - LB}{LB} \cdot 100 \tag{13}$$

4 Layout analysis and simulate annealing model calibration

It is known that SA performs according to the setting of its algorithm-specific parameters. Hence, a factorial design was implemented to test the influence on both solution quality and running time. Table 2 shows the different levels investigated for each parameter based on a set of preliminary tests.

Table 2 SA tested parameters

Parameters	*Values*
k_{max}	10; 50
It	10; 100
T_0	200; 50; 2
Cooling schedule	Linear; geometrical

Due to its stochastic nature, the algorithm was run 100 times for each one of these 24 combinations. A test set of 18 different job lists was chosen and the experimental set-up was conducted for each layout shown in Table 1.

Figures 5 to 9 report solution quality (i.e., Δ%*LB*) versus computational time (in minutes) for each layout. Each dot in the graph is the average value obtained for 100 algorithmic runs of each of the 18 job list scenarios for a particular combination of the four parameter considered shown in Table 2. Hence, every dot is representing 1,800 algorithmic runs. Different layouts have been tested, as reported in the figures' legend.

Figure 5 Δ%*LB*-running time, layout 9 machines-3operators (see online version for colours)

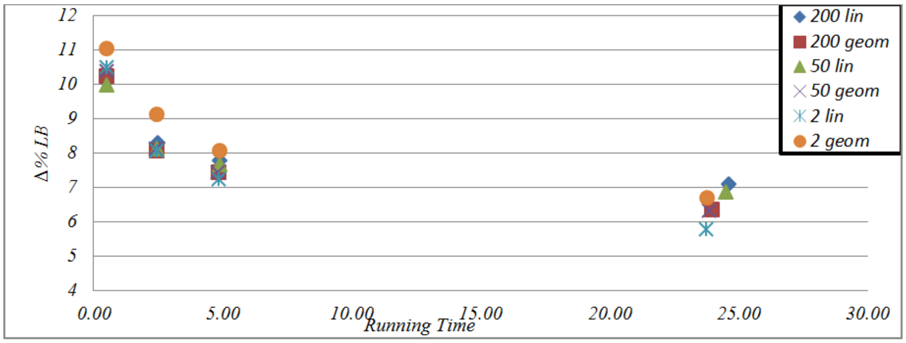


Figure 6 Δ%*LB*-running time, layout 15 machines-5operators (see online version for colours)

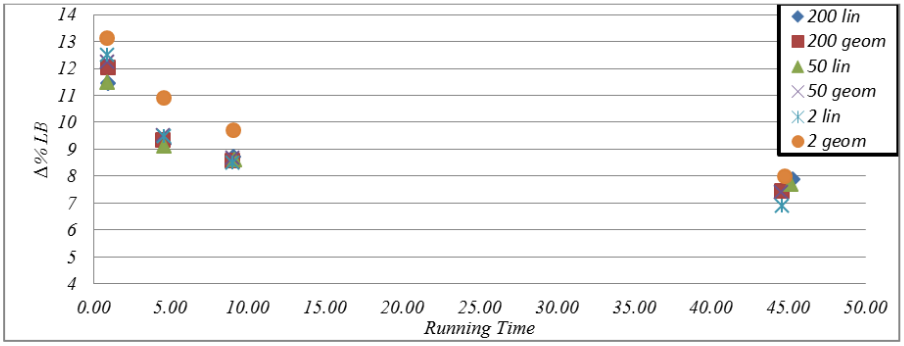


Figure 7 Δ%*LB*-running time, layout 6 machines-2operators (see online version for colours)

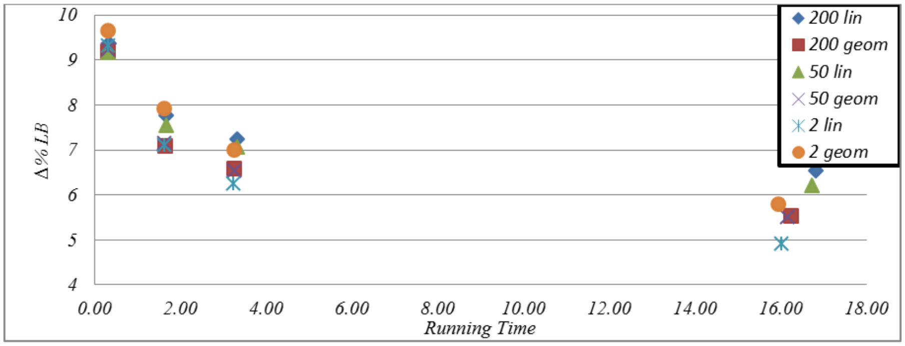


Figure 8 Δ%*LB*-running time, layout 8 machines-4operators (see online version for colours)

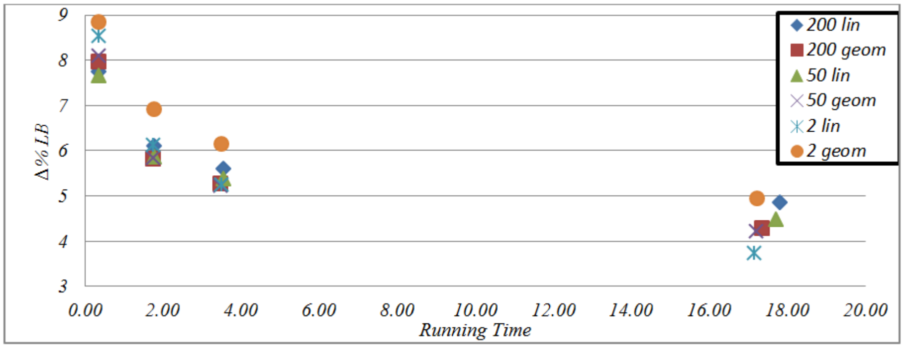
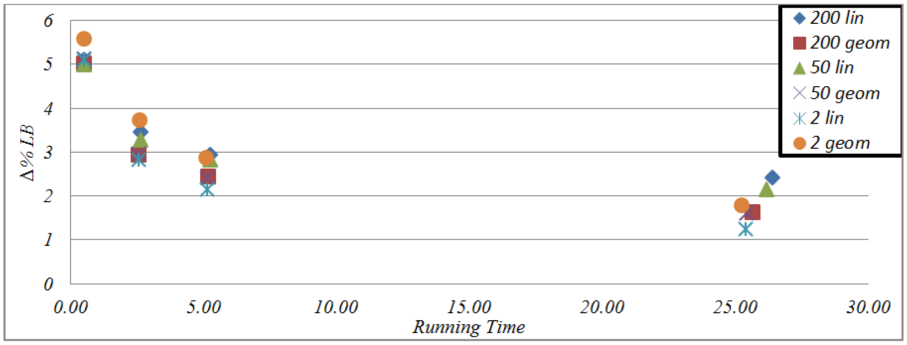


Figure 9 Δ%*LB*-running time, layout 8 machines-2operators (see online version for colours)



Given the parameter domains tested, the best parameter combination with regards to solution quality for the particular layout considered is: T_0 = 2, *linear cooling schedule*, k_{max} = 50 and *it* = 100 (Figures 5 to 9). This shows that although solution quality levels and computational times vary for different layouts, the algorithm is not significantly affected by the change in algorithmic parameters and performs in a somewhat robust way across different layout modifications. Table 6 reports the solution quality (Δ%*LB*) for the best parameter setting for each layout. A linear regression analysis with IBM SPSS software has been conducted to investigate the impact of a change in algorithm parameters on computational time. Results are reported in Tables 3 and 4.

Table 3 Running time linear regression coefficients for layouts with different ratios

r	*Layout*	*Constant*		*kmax*		*it*		T_0		*Cooling*		R^2
		B	*Sign.*	*B*	*Sign.*	*B*	*Sign.*	*B*	*Sign.*	*B*	*Sign.*	*(%)*
3	9-3	–7,774	0.000	0.265	0.000	0.144	0.000	0.001	0.813	–0.121	0.773	78.7
	15-5	–14,576	0.000	0.492	0.000	0.269	0.000	0.000	0.950	–0.105	0.899	76.9
	6-2	–5,228	0.000	0.179	0.000	0.098	0.000	0.001	0.701	–0.125	0.662	78.6
2	8-4	–5,623	0.000	0.191	0.000	0.104	0.000	0.000	0.805	–0.088	0.774	78.4
4	8-2	–8,272	0.000	0.283	0.000	0.154	0.000	0.001	0.743	–0.164	0.716	78.6

Table 4 Δ%*LB* for the best parameter setting for the different tested layouts

n_{mach}	n_{op}	*r*	Δ%*LB*
9	3	3	5.76
15	5	3	6.89
6	2	3	4.91
8	4	2	1.23
8	2	4	3.74

Table 3 highlights that there is no significant impact of the algorithm-specific parameters T_0 and the cooling schedule on computational times, while it underpins that the number of iterations are significant due to the known trade-off between solution quality and computational times.

Considering the outlined results, it is important to highlight that the multiple linear regression results on solution quality (Δ%LB) showed a significant effect of the number of iterations on running times, while no significant effects were found related to the starting temperature or the cooling schedule (Table 4). It is important to underline that the model explains almost 90% of the variation in the data for all the five layouts tested.

5 Impact analysis of n_{mach}/n_{op} ratio on solution quality

In particular in JIT production systems, the job list characteristics can be far from the nominal ratio $r = n_{mach}/n_{op}$ determined in Phase 1.

Let r^* be the actual ratio for a given job list defined as shown in (1)

$$r^* = \frac{\sum_{i=1}^{n_{jobs}} \left(L_i + R_i + U_i\right)}{\sum_{i=1}^{n_{jobs}} \left(L_i + U_i\right)} \tag{14}$$

and $\overline{r^*}$ the average of r^* over the 18 job lists considered for each layout in the analysis (Table 5).

Table 5 Values of $\overline{r^*}$ for different layouts

Layout	*r*	$\overline{r^*}$
9-3	3	3.09
15-5	3	2.99
6-2	3	3.07
8-4	2	1.91
8-2	4	3.63

The difference between the actual ratio r^* and the nominal ratio r is notated by:

$$\Delta r = r^* - r \tag{15}$$

Using Δ*r*, (1) can be expressed as

$$|\Delta r| \leq 0.5 \tag{16}$$

Δ*r* represents the discrepancy between the job list characteristics and ratio *r* identified in Phase 1 where the optimal value of Δ*r* is 0. A specific analysis of the effects of different values of Δ*r* for each of the five tested layouts is carried out, considering $k_{max} = 50$, $it = 100$, $T_0 = 2$ and a linear cooling schedule. A new test set is created consisting of 21 new instances, with $-0.5 < \Delta r < 0.5$. The algorithm is run 100 times for each instance.

Figures 10 to 14 report the solution quality Δ%*LB* plotted against Δ*r* for each shop floor layout. The graphs show a similar pattern for each of the five considered layouts. When $\Delta r \to +\infty$ a pure machines constrained problem is described, while $\Delta r \to -\infty$

characterises a pure operator constrained problem. If $\Delta r \to 0$, a perfect double resource constrained problem is at hand.

The algorithmic behaviour was found to be consistent with regards to solution quality for all different layouts. It becomes apparent that solution quality decreases if the actual ratio r^* is close to the nominal one ($\Delta r \cong 0$), while better solutions are obtained for large values of $|\Delta r|$. If $|\Delta r|$ increases, one of the two constraints (machines and operators availability) becomes more relevant than the other, hence the DRC problem gradually turns into a SRC one which is easier to solve. With Δr decreasing, the operator availability gradually becomes the dominant constrained resource. In contrast, as Δr increases, machine availability becomes the dominant constrained resource. Figures 10 to 14 show how the corresponding performance curves are not symmetrical, as the quality of the solutions improves faster when the problem turns into being operator-constrained rather than machine-constrained. It can be seen that for a job list with $\Delta r = –0.5$ (borderline scenario of operator-constrained system) the algorithm is able to find very good solutions with $\Delta\%LB \cong 0$.

Figure 10 Impact of Δr variation on $\Delta\%LB$, layout 9 machines, 3 operators (see online version for colours)

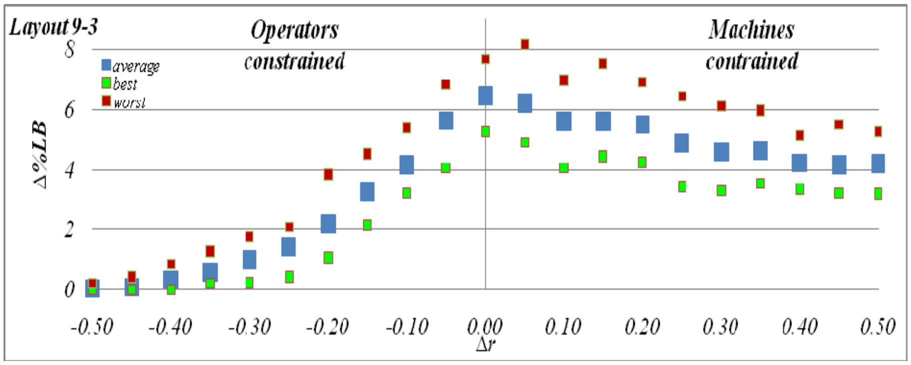


Figure 11 Impact of Δr variation on $\Delta\%LB$, layout 15 machines, 5 operators (see online version for colours)

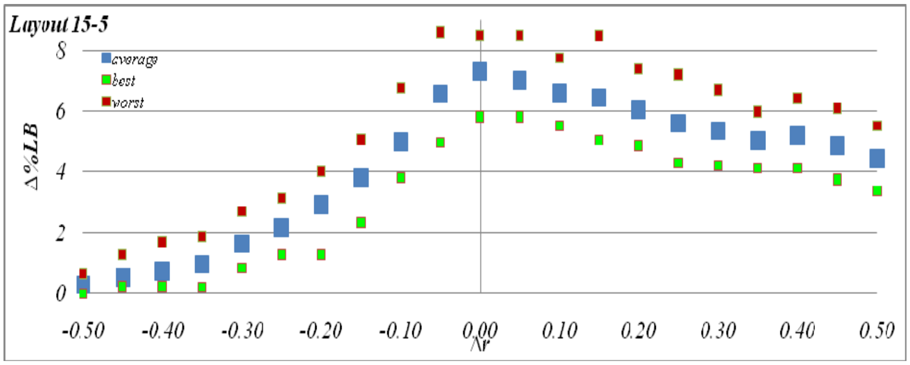


Figure 12 Impact of Δ*r* variation on Δ%*LB*, layout 6 machines, 2 operators (see online version for colours)

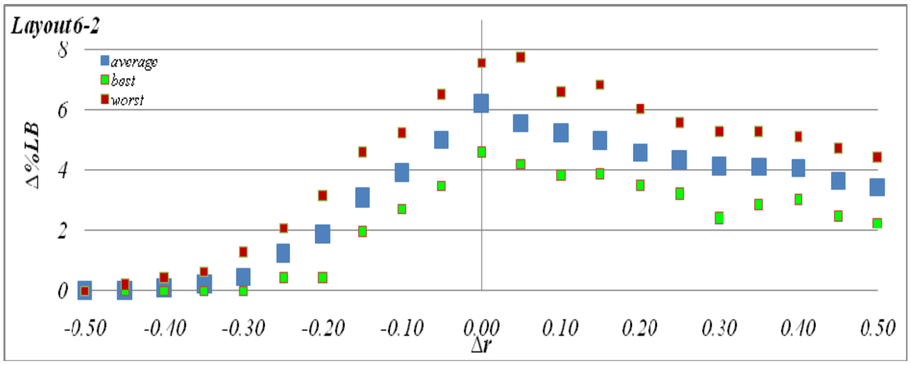


Figure 13 Impact of Δ*r* variation on Δ%*LB*, layout 8 machines, 4 operators (see online version for colours)

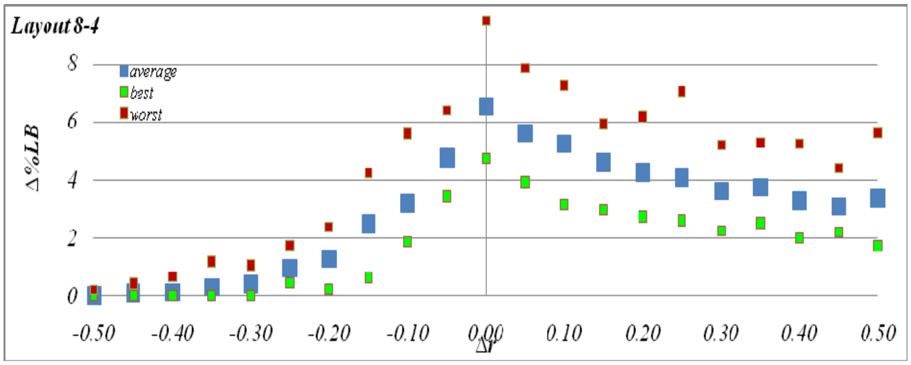


Figure 14 Impact of Δ*r* variation on Δ%*LB*, layout 8 machines, 2 operators (see online version for colours)

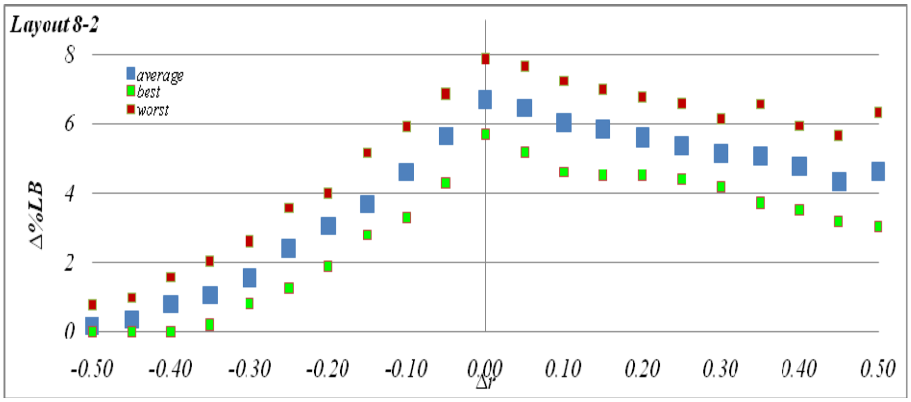


Table 6 reports the results for the three main scenarios: operator constrained ($\Delta r = -0.5$), perfect DRC ($\Delta r = 0$) and machine constrained ($\Delta r = 0.5$).

Table 6 $\Delta\%LB$ for different Δr over different layouts

Layout	$\Delta\%LB$		
	$\Delta r = -0.5$	$\Delta r = 0$	$\Delta r = 0.5$
9-3	0.05	6.47	4.21
15-5	0.28	7.31	4.43
6-2	0.00	6.23	3.43
8-4	0.00	6.53	3.38
8-2	0.19	6.69	4.63

A possible reason for the asymmetric performance behaviour can be found by looking at the two borderline cases of SRC problems for the case under study. In a pure machine constrained problem ($\Delta r \to +\infty$), segments of $L_i + R_i + U_i$ time units (total time to perform job i) must be allocated to different machines in order to find the solution with the lower makespan. In contrast, in a pure operator constrained problem ($\Delta r \to -\infty$), segments of $U_i + L_j$ time units have to be allocated to different operators. These segments are shorter than the ones in the machine constrained case. Furthermore, job i can be any of the remaining jobs, and this drastically increases the number of possible combinations. This may be a reason for the existence of smaller gaps between actual makespan and lower bound being more likely in the case of an operator-constrained problem than in a machine-constrained one.

6 Idle times and cost analysis

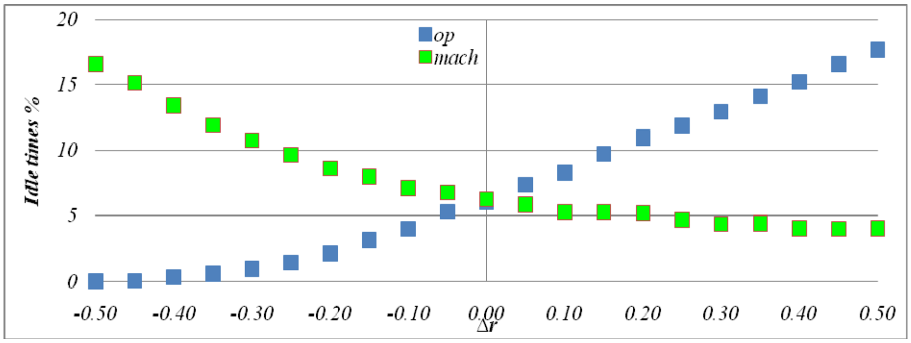
In production environments, the focus is on maximising the productivity of resources which in a DRC system are both machines and operators. The maximum productivity is obtained when there is no idle time. Recalling expressions (10) and (11) used for the lower bounds, it can be noted that they correspond to ideal solutions where there is no machine or operator idle time, respectively. It is therefore possible to use them to calculate the machine and operator idle times.

$$I_m = \frac{MKSP - LB_m}{MKSP} \cdot 100; \ \% \text{ of machines idle time} \tag{17}$$

$$I_o = \frac{MKSP - LB_o}{MKSP} \cdot 100; \ \% \text{ of operators idle time} \tag{18}$$

Idle times are strongly influenced by the ratio r of the particular job list considered. As an example, Figure 15 shows this relation for the case of the layout 9-3.

Figure 15 Impact of Δr variation on machines and operators idle times, layout 9-3 (see online version for colours)



The following considerations are outlined focusing on this particular layout only for brevity, but may be generalised. It can be noted that in the case of operator constrained systems ($\Delta r = -0.5$) there is no operator idle time ($I_o \cong 0\%$) while the machine idle time is at its maximum ($I_m \cong 17\%$). Moving towards the perfect DRC system, the idle time for the operator increases while there is a reduction in machine idle time: the two become eventually very similar for $\Delta r = 0$, where $I_o \cong I_m \cong 6\%$. Increasing Δr even further, it can be seen how values for operator idle times increase while machine idle times decrease. For the borderline case of a machine constrained system ($\Delta r = 0.5$) the values are $I_o \cong 18\%$ and $I_m \cong 4\%$. With I_o and I_m referring to the average idle times, corresponding cost of idle resources can be estimated such that:

$$I_{m,tot} = n_{mach} \cdot I_m \tag{19}$$

$$I_{o,tot} = n_{op} \cdot I_m \tag{20}$$

If c_m and c_o are the cost per hour for one machine and one operator respectively, the total cost C of the idle time is

$$C = c_m \cdot I_{m,tot} + c_o \cdot I_{o,tot} \tag{21}$$

The total cost of the idle time is plotted against Δr in Figure 16. The curves refer to different values of ratio $r = c_o/c_m$.

The Δr of minimum cost (from here on referred to as Δr_{min}) varies for different ratios c_o/c_m. Table 7 shows the values for all cases taken into consideration.

Since the goal is the minimisation of the idle time cost, Figure 16 can be used as an effective tool to determine it. Given the particular ratio c_o/c_m of the job shop under investigation, the objective is to have job lists with Δr as near as possible to Δr_{min}. Recalling equation (15), Δr is given by the difference between the actual ratio r^* of the job lists and the ratio of the layout r: $\Delta r = r^* - r$. For each instance, all job processing times are given, meaning that the corresponding r^* is fixed and cannot be changed. The only way to change Δr is to modify the layout, e.g. by acquiring a new machine or hiring a new operator. If a new machine is added to the job shop, Δr decreases and the system moves towards an operator constrained one.

Figure 16 Impact of Δr variation on machines and operators idle times, layout 9-3 (see online version for colours)

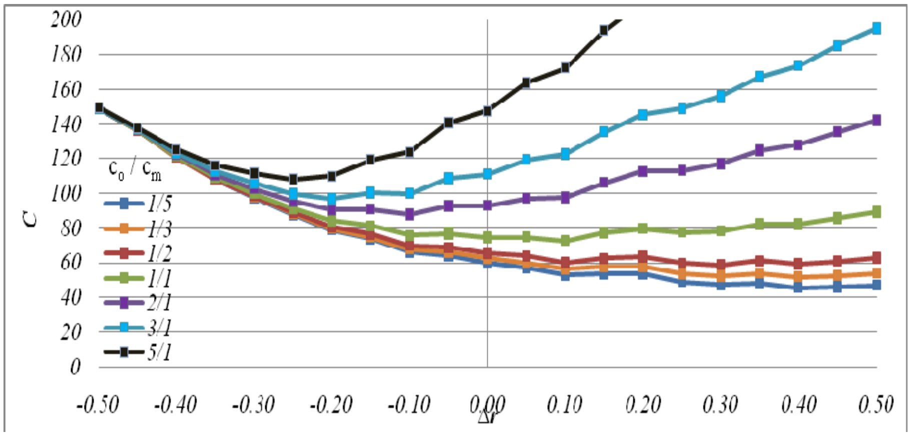


Table 7 Values of Δr_{min} in function of c_o/c_m

c_o/c_m	Δr_{min}
1/5	0.40
1/3	0.40
1/2	0.30
1/1	0.10
2/1	–0.10
3/1	–0.20
5/1	–0.25

7 Conclusions and further research

This paper presents a solution framework for the DRC problem considering a job shop layout composed of a number of identical machines n_{mach} and n_{op} workers with a 100% flexibility level. The framework is divided into two phases allowing for the determination of a ratio between the number of machines and number of operators, followed by a solution concept using SA. Distinctive features of the proposed SA algorithm are the implementation of multiple neighbourhood operators and a memory-based iterative strategy. The research was motivated by the lack of scientific contribution related to this specific area and the wide presence of the dual resources constrained problem in industry, especially in job shop production environments such as the presented case study.

The results reported in the paper can be summarised as follows:

- The main parameters of SA have been identified as the number of iterations for the external loop, the number of iterations for the internal loop, the starting temperature and the cooling schedule. The analysis has been developed for different selected layouts. It was shown that algorithm performance has not been affected substantially by changes in the cooling schedule and the starting temperature. The proposed SA algorithm proves to be an effective tool to solve the problem, with acceptable computational time required.
- The high variability in job times is a characteristic of JIT production environments and forms the basis for the presented analysis using different job shop layouts. As demonstrated by the results, the selected layout determined by the ratio n_{mach}/n_{op} strongly influences production system performance. The required n_{mach}/n_{op} ratio, function of the jobs list and sequence, versus the n_{mach}/n_{op} ratio given by the production system layout strongly influence the solution quality of the proposed DRC scheduling procedure. A perfect DRC problem is substantially more difficult to solve compared to problems that are more similar to SRC problems. Results demonstrate also that this trend is not symmetrical as the quality of the solutions improves faster with the problem turning into being operator-constrained rather than machine-constrained.
- As companies aim to minimise costs, a further extension to the work was investigated regarding the minimisation of idle time costs. Considering the cost per hour for one machine c_m and one operator c_o, it is shown that the optimum setting of the $n_{mach.}/n_{op}$ changes as function of the ratio c_o/c_m.

For its practical implications in many industrial situations, future research on the DRC scheduling problem is certainly needed. Possible extension of the presented study may consider broadening the set of parameter values tested, taking also into consideration a more extensive set of different layouts. As it was seen to have a major influence on the quality of the solution, the impact of the job list characteristics should also be extended. Furthermore, it would be interesting to see an integrated decision modelling strategy that seeks to take into consideration a full cost analysis linked to acquiring new resources in order to improve the overall efficiency of the system schedule.

References

Araz, O.U. and Salum, L. (2008) 'A multi-criteria adaptive control scheme based on neural networks and fuzzy inference for DRC manufacturing systems', *International Journal of Production Research*, Vol. 48, No. 1, pp.251–270.

Azzi, A., Battini, D., Faccio, M. and Persona, A. (2012a) 'Mixed model assembly system with multiple secondary feeder lines: layout design and balancing procedure for ATO environment', *International Journal of Production Research*, Vol. 50, No. 18, pp.5132–5151.

Azzi, A., Battini, D., Faccio, M. and Persona, A. (2012b) 'Sequencing procedure for balancing the workloads variations in case of mixed model assembly system with multiple secondary feeder lines', *International Journal of Production Research*, Vol. 50, No. 21, pp.6081–6098.

Azzi, A., Faccio, M., Persona, A. and Sgarbossa, F. (2012c) 'Lot splitting scheduling procedure for make span reduction and machines capacity increase in a hybrid flow shop with batch production', *International Journal of Advanced Manufacturing Technology*, Vol. 59, Nos. 5–8, pp.775–786.

Balin, S. (2012) 'Non-identical parallel machine scheduling with fuzzy processing times using genetic algorithm and simulation', *International Journal of Advanced Manufacturing Technology*, Vol. 61, Nos. 9–12, pp.1115–1127.

ElMaraghy, H. (2000) 'Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms', *Journal of Manufacturing Systems*, Vol. 19, No. 3, pp.186–198.

Finetto, C., Faccio, M., Rosati, G. and Rossi, A. (2013) 'Sequencing optimization for an automated single station fully flexible assembly system (F-FAS)', *International Journal of Advanced Manufacturing Technology*, Vol. 70, Nos. 5–8, pp.797–812.

Fredendall, L.D., Melnyk, S.A. and Ragatz, G. (1996) 'Information and scheduling in a dual resource constrained job shop', *International Journal of Production Research*, Vol. 34, No. 10, pp.2783–2802.

Gunther, R.E. (1981) 'Dual response parallel queues with server transfer and information access delays', *Decision Sciences*, Vol. 12, No. 1, pp.197–211.

Hadidi, L.A., Al-Turki, U.M. and Rahim, M.A. (2011) 'An integrated cost model for production scheduling and perfect maintenance', *International Journal of Mathematics in Operational Research*, Vol. 3, No. 4, pp.395–413.

Hottenstein, M.P. and Bowman, S.A. (1998) 'Cross-training and worker flexibility: a review of DRC system research', *The Journal of High Technology Management Research*, Vol. 9, No. 2, pp.157–174.

Johnson, S.M. (1954) 'Optimal two-and three-stage production schedules with set up times included', *Naval Research Logistics Quarterly*, Vol. 1, No. 1, pp.61–68.

Kher, H.V. (2000) 'Examination of worker assignment and dispatching rules for managing vital customers priorities in dual resource constrained job shop environments', *Computers and Operations Research*, Vol. 27, No. 6, pp.525–537.

Kher, H.V. and Fry, T.D. (2001) 'Labour flexibility and assignment policies in a job shop having incommensurable objectives', *International Journal of Production Research*, Vol. 39, No. 11, pp.2295–2311.

Li, X., Baki, M.F. and Aneja, Y.P. (2011) 'Flow shop scheduling to minimize the total completing time with permanently present operator: models and ant colony optimization metaheuristic', *Computers & Operations Research*, Vol. 38, No. 1, pp.152–164.

Liao, C.J. and Lin, H.T. (1998) 'A case study in a dual resource constrained job shop', *International Journal of Production Research*, Vol. 36, No. 11, pp.3095–3111.

Malhotra, M.K. and Kher, H.V. (1994) 'An evaluation of worker assignment policies in dual resource constrained job shops with heterogenous resources and worker transfer delays', *International Journal of Production Research*, Vol. 32, No. 5, pp.1087–1103.

Najid, N.M. and Arroub, M. (2010) 'An efficient algorithm for the multi-mode resource constrained project scheduling problem with resource flexibility', *International Journal of Mathematics in Operational Research*, Vol. 2, No. 6, pp.748–761.

Nelson, R.T. (1967) 'Labor and machine limited production systems', *Management Science*, Vol. 13, No. 9, pp.648–671.

Ries, J., Beullens, P. and Salt, D. (2012) 'Instance-specific multi-objective parameter tuning based on fuzzy logic', *European Journal of Operational Research*, Vol. 218, No. 2, pp.305–315.

Ruiz-Torres, A.J. and Centeno, G. (2008) 'Minimizing the number of late jobs for the permutation flowshop problem with secondary resources', *Computers and Operations Research*, Vol. 35, No. 4, pp.1227–1249.

Salum, L. and Araz, O.U. (2009) 'Using the when/where rules in dual resource constrained systems for a hybrid push-pull control', *International Journal of Production Research*, Vol. 47, No. 6, pp.1661–1677.

Sammarco, M., Fruggiero, F., Lambiase, A. and Neumann, W.P. (2013) 'Agent-based modelling of movement rules in DRC systems for volume flexibility: human factors and technical performance', *International Journal of Production Research*, Vol. 52, No. 3, pp.633–650.

Sharma, S., Gupta, D. and Sharma, S. (2013) 'Bicriteria scheduling on parallel machines: total tardiness and weighted flowtime in fuzzy environment', *International Journal of Mathematics in Operational Research*, Vol. 5, No. 4, pp.492–507.

Tao, Z., Xiao, T. and Hao, C. (2007) 'Petri net and GASA based approach for dynamic JSP', *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation*, Art. No. 4304196.

Treleven, M. (1989) 'A review of the dual resource constrained system research', *IIE Transactions* (Institute of Industrial Engineers), Vol. 21, No. 3, pp.279–287.

Venkataramana, M. and Srinivasa Raghavan, N.R. (2010) 'Ant colony-based algorithms for scheduling parallel batch processors with incompatible job families', *International Journal of Mathematics in Operational Research*, Vol. 2, No. 1, pp.73–98.

Vinod, V. and Sridharan, R. (2008) 'Scheduling a dynamic job shop production system with sequence-dependent setups: an experimental study', *Robotics and Computer-Integrated Manufacturing*, Vol. 24, No. 3, pp.435–449.

Xu, J., Xu, X. and Xie, S.Q. (2011) 'Recent developments in dual resource constrained (DRC) system research', *European Journal of Operational Research*, Vol. 215, No. 2, pp.309–318.