Journal of Industrial and Systems Engineering
Vol. 10, Special Issue on Scheduling, pp. 58- 77
Autumn (November) 2017

A novel mathematical model for a hybrid flow shop scheduling problem under buffer and resource limitations-A case study

Ghasem Safari[1], Ashkan Hafezalkotob[2*], Mohammad Khalilzadeh[1]

[1]*Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*
[2]*Department of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran*

gh.safari@srbiau.ac.ir, a_hafez@azad.ac.ir, khalilzadeh@srbiau.ac.ir

Abstract
Scheduling problems play a big role in manufacturing and planning the production for increasing efficiency and assigning the most convenient resources to operations. Furthermore, in many manufacturing systems the existence of intermediate buffers is one of the essential factors for smoothing the material flow. In this study, a model is proposed for minimizing the makespan of a hybrid flow shop scheduling problem with intermediate buffers and resource constraints. These constraints exist in almost every realistic manufacturing system and have an imperative impact on improving the production cost, productivity, and sustainability. In this study, a hybrid algorithm based on genetic algorithm and variable neighborhood search is used, which in tuned with Taguchi's method solve the proposed model for a tire manufacturing company. The results show that the proposed mathematical model has a high ability for scheduling problems with resource and intermediate buffer constraints and is solvable by the hybrid genetic algorithm.
Keywords: Hybrid flow shop scheduling, buffer limits, resource constraints, hybrid genetic algorithm.

1- Introduction

Scheduling models and algorithms have a significant impact on enhancing manufacturing productivity and are essential for production planning in competitive environments. They assign resources to operations throughout the specified time intervals for optimizing different processes. One of the most important scheduling problems is the hybrid flow shop. This scheduling problem is a combination of series and parallel production lines and consists of a set of u stages with each stage k $(1 \leq k \leq u)$ have m_k parallel machines. All the jobs have to be processed through all the stages via the same path. Assume that there are n jobs i $(1 \leq i \leq n)$. Each job consists of a chain of u operations $(o_{i1}, \ldots, o_{iu})$so that they have to be executed with the following order:

- Each operation o_{ik}takes a processing time p_{ikm_k}on machinem_k at the stage k
- Unlike the classic flow shop problems, each stage can have more than one machine for processing the jobs.

*Corresponding author
ISSN: 1735-8272,

Hybrid flow shop is often observed in the electronic manufacturing environment such as IC packaging and PCB fabrication (Allaoui & Artiba, 2006). Furthermore, in some cases of the hybrid flow shop scheduling problem, there are intermediate buffers between two consecutive stages.

The capacity of the buffers can be finite or infinite (depending on the size of semi-manufactured). In general, the capacity of buffers is finite. When the capacity is zero, the problem is called blocking flow shop. In this case, when all the machines are busy in the next stage, the job is blocked at the current stage (i.e., the jobs will wait in the buffers).

Most of the flow shop scheduling studies considered machines as the only resource. However, in most real-world manufacturing systems, jobs may require resources (such as energy, operators, equipment and tools, material handling equipment, molds and industrial robots) for handling and processing (Edis & Oguz, 2011). Practically, every machine at each stage uses some resources for job processing. When the capacity of resources is limited, the resulting scheduling problems are known as Resource-Constrained Flow Shop Scheduling Problems (RCFSP) (Lovaa, Tormos, Cervantes, & Barber, 2009). Table 1 gives a classification of the resource constrained scheduling models as proposed by Blazewicz, et al.(2007).

Table 1. The classification of resources

Viewpoint	Class	Definition
Time of need	Processing resource	The resource is required during the processing of a job.
	Input–output resource	Resource is required either before or after the processing of a job
Renewability	Renewable	Total usage at every moment is constrained. Once it is used for a job, it may be used again for another job
	Non-renewable	Total consumption is constrained. In other words, once it is used by some job, it will not be available for any other job
	Doubly constrained	It is both renewable and non-renewable
Divisibility	Discrete resources	Can be allocated to jobs in discrete units
	Continuous resources	Can be allocated to jobs in arbitrary amounts within an interval

The literature review on these problems can be broadly grouped into two categories: (i) survey on intermediate buffer limits in the hybrid flow shop problems and (ii) survey on resource constraints in the hybrid flow shop problems.

1-1- Survey on intermediate buffer limits in the hybrid flow shop problems

Yaurima, Burtseva, and Tchernykh (2009) presented a genetic algorithm for an important production scheduling problem. Since the problem was NP-hard, they focused on suboptimal scheduling solutions for the hybrid flow shop with unrelated machines, sequence-dependent setup time, availability constraints, and limited buffers. The production environment of a television assembly line for inserting electronic components was considered. Shi Qiang Liu and Kozan (2009) considered four inter-machine buffer conditions (i.e. *no-wait*, *no-buffer*, *limited-buffer* and *infinite-buffer*) in a flow shop are investigated.

These four different buffer conditions were combined to generate a more generalized and more comprehensive scheduling problem, which could cover the *classical flow-shop scheduling* (FSS) with unlimited buffer, the *blocking FSS* (BFSS) without buffer, the *no-wait FSS* (NWFSS) and the *limited-buffer FSS* (LBFSS) problems. Belaid et al.(2012)investigated the problem of scheduling batches on buffers with non-negligible transfer and setup times in a real-life industrial scenario. Moslehi and Khorasanian(2014) developed a hybrid variable neighborhood search (VNS) algorithm with simulated annealing for the limited-buffer permutation flow shop scheduling problem. Liu et al.(2015)presented a hybrid algorithm for the permutation flow shop scheduling problem without intermediate buffers based on the scatter search and the variable neighborhood search. Abdollahpour and Rezaeian(2015) presented three new meta-heuristics for permutation flow shop scheduling problem with intermediate buffer. Liu and Kozan(2016)proposed an MIP model for Parallel-identical-machine job-shop scheduling under different buffering requirements.Ribas, Companys, and Tort-Martorell (2017) presented a mathematical model along with some constructive and improvement heuristics to solve the parallel blocking flow shop problem (PBFSP) and to minimize the maximum completion time among lines. Fu, Wang, Zhang, and Wang (2017) proposed a blocking flow shop deteriorating scheduling problem which had a widespread application in manufacturing and service systems.

1-2- Survey on resource constraints in the hybrid flow shop problems

Blazewicz (1978), Blazewicz and Ecker (1983), Blazewicz et al.(1987) and Ventura and Kim (2000)studied resource constraints on identical parallel machine scheduling problems (RCPMSPs) and proposed polynomial-time exact algorithms for solving them. Brucker and Kramer (1996) presented polynomial-time algorithms for multiprocessor task scheduling problems with resource constraints. They divided a set of tasks into a fixed number of classes and all the tasks belonging to the same class have the same processing time and resource requirements. Resource constraints on uniform and unrelated machines were studied by Kovalyov and Shafransky(1998), Ruiz-Torres et al.(2007), Grigoriev et al. (2007) and Edis and Oguz (2012).Daniels et al. (1996) investigated that job processing times were not fixed and were related to the quantity of additional allocated resources. Some studies assumed that the hybrid resources may be allocated freely among the machines whereas the resource capacity remains fixed in the entire scheduling horizon. Kellerer and Strusevisch (2008) defined a resource constraint for the parallel machine problem with binary resource allocation, speeding-up resource and makespan objective. Edis and Ozkarahan (2011) proposed a combined integer/constraint-programming approach to a resource constrained parallel machine scheduling problem with machine eligibility restrictions. Waldherr and Knust (2016) used decomposition algorithms for synchronous flow shop problems with additional resources and setup times. Laribi et al.(2016) proposed a heuristic algorithm for solving two-machine flow shop scheduling problems under resources constraints. Afzalirad and Rezaeian (2016) proposed a new pure integer mathematical model for resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. Mansouri and Aktas (2016) developed constructive heuristics and multi-objective genetic algorithms (MOGA) for a two-machine sequence-dependent permutation flow shop problem to address the trade-off between energy consumption (as a measure of sustainability) and makespan (as a measure of service level). They leveraged the variable speed of operations to develop energy efficient schedules that minimized the total energy consumption and makespan. Fanjul-Peyro et al. (2017) modeled a parallel machine scheduling problem in which the job processing by machines requires several units of a scarce resource. This number depends both on the job and machine where the availability of resources is limited and fixed throughout the production horizon.

1-3- Research gap and contributions

Although the limited buffer flow shop scheduling problem has not been studied as extensively as the hybrid flow shop problem and the majority of studies have focused on solving rather than modeling the problems with buffer limits, the number of published articles, addressing the former issue in order to minimize makespan, has increased in recent years. Furthermore, to the best of our knowledge, there are no studies considering two constraints on resources and buffers simultaneously.

In this study, a mixed integer programming (MIP) model for the hybrid flow shop problem under two simultaneous constraints on capacities of resources and buffers is proposed. These two constraints exist in almost every real manufacturing system. Since in our case study, the energy conduits are considered as resources then this model is effective for sustainable production. In other hand, with considering buffer limits then it is also helpful in reducing semi-production storage cost and in improving the production cost. The mathematical model is solved using a genetic algorithm that is hybridized with variable neighborhood search (VNS) and the parameters are tuned based on Taguchi's method.

In this article, Section 2introduces the proposed model and the hybrid meta-heuristics algorithm is described in Section 3. The case study and computational results are presented in Section 4and Section 5, respectively. Some managerial insights discussed in Section 6and finally the concluding remarks are compiled in Section 7.

2- Problem description and formulation

The problem are composed from a hybrid flow shop problems with one or more machine at each stage that consume resources for processing job and finally, every job for processing at each stage should go away to related buffer and waits for processing by a machine (Figure1). In this problem, every job requires a determined set of resources to be processed via each machine. These resources may differ from one machine to the other for the same stage. The capacity of each resource (all resources being renewable) is constrained in discrete units and these resources are consumed during job processing. It is supposed that between two successive stages there exists a buffer where each job enters, after processing from the first stage, and waits there to be assigned to a machine in the next stage. In this problem, it is assumed that buffer capacities are limited and predetermined. This problem can be denoted by HFS $|Limited\ buffer, Res|\ C_{max}$ based on three-field notation of Pinedo (2016).

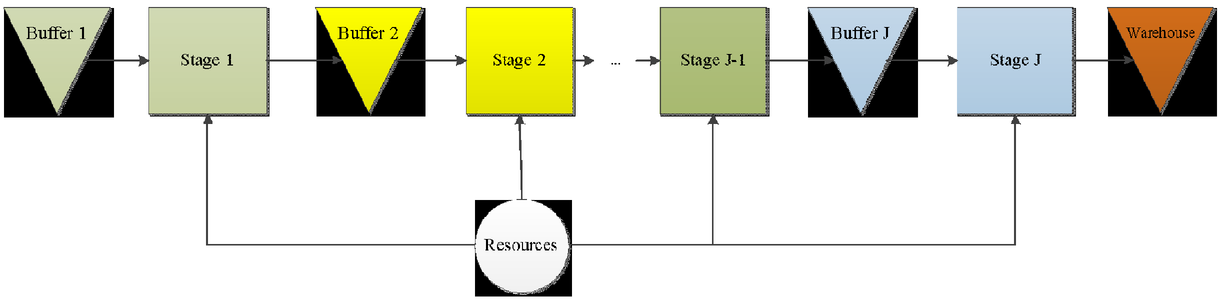

Figure 1. A schematic model of the problem

We make the following assumptions to establish the framework of the study for further mathematical modeling:

Assumption1. A job cannot be met more than one time at each stage. In other word, every job at each stage can be processed only by one machine and we do not have any rework to reprocess the jobs. Eventually, no machine can process more than one job at the same time. As a result, if a machine be busy then next job must wait until present job completes its processing time.

Assumption2. Resource usage by each job on each machine is deterministic.

Assumption3. Buffer capacity for every stage is pre-deterministic and is unlimited for the first stage and warehouse.

Assumption4. Setup times are considered as a part of processing time. In some cases, there are many settings for machines setup that can be dependent or independent from jobs and performing of these settings is time consuming which is called setup time. Setup times for a machine can be different from the others at each stage. In our case study, we neglect the setup times as a separate parameter from processing times. Let $r_{ijm}$and t_{ijm}respectively be setup time and cycle time of job i at stage j on machine m .Then the processing time is expressed as follows:

$P_{ijm} = r_{ijm} + t_{ijm}, \quad i = 1, \dots, n\ ;\ j = 1, \dots, J; m = 1, \dots, m_j$

Assumption5. At each stage, machines are always available. Sometimes it is possible that machines cannot process jobs due to planned or unplanned maintenance. In this condition, jobs must wait until the maintenance activities are finished.

For describing the mathematical model, we consider$\pi = \{\pi(1), \dots, \pi(n)\}$as a feasible sequence of jobs. Other notations of the problem are as follows:

Indexes	**Description**
j	stage index
i, r, k, h	job indexes
m, e	machine index
v	resource index
m_j	number of machines in stage j
B_j	buffer capacity of stage j
P_{rmj}	processing time of job r with m-th machine at stage j
$r_{ij}^{v}(\pi)$	resource requirement for processing of job $\pi(i)$at stage j under solution π
R_v	capacity of resource v in every time
M	a big positive number
$s_{ij}(\pi)$	starting time for processing of job $\pi(i)$at stage j under solution π
$T_{ij}(\pi)$	departure time of job $\pi(i)$at stage j under solution π
$x_{irmj}(\pi)$	1; if job r be in position $\pi(i)$ at stage j and processed by machine m under solution π, 0; else
$z_{irj}(\pi)$	1; if jobs in two positions of $\pi(i)$ and $\pi(r)$ have overlapping in processing time at stage j, 0; else
C_{max}	makespan of the system

$$Min C_{max} \tag{1}$$

s.t.

$$s_{kj} - s_{ij} - \sum_{r=1}^{n} x_{irmj} P_{rmj} - M(1\text{-}z_{ikj}) \leq 0 \quad ; j = 1, \dots, J, i = 1, \dots, n-1, m = 1, .., m_j k = i+1, \dots, n \tag{2}$$

$$s_{ij} - s_{kj} \leq 0 \quad ; j = 1, \dots, J\ , \quad i = 1, \dots, n-1\ , \quad k = i+1, \dots, n \tag{3}$$

$\sum_{k=1}^{n} z_{ikj} r_{kj}^{v} + r_{ij}^{v} \leq R_j^{v}$	$; v = 1, \dots, V, \quad j = 1, \dots, J, i = 1, \dots, n-1$	(4)
$T_{i,j} = s_{i,j} + \sum_{r=1}^{n} x_{irmj} P_{rmj}$	$; i \leq B_j j = 1, \dots, J-1, m = 1, .., m_j$	(5)
$T_{i,j} \geq s_{i,j} + \sum_{r=1}^{n} x_{irmj} P_{rmj}$	$; i > B_j j = 1, \dots, J-1, m = 1, .., m_j$	(6)
$T_{i,j} \geq s_{k,j+1}$	$; i > B_j k = i-1, \dots, 1$	(7)
$s_{i,j+1} - T_{k,j} - M(2 - x_{irej+1} - x_{krmj}) \geq 0$	$; j = 1, \dots, J, i, r, k = 1, \dots, n, m = 1, .., m_j e = 1, .., m_{j+1}$	(8)
$\sum_{m=1}^{m_j} \sum_{r=1}^{n} x_{irmj} = 1$	$; j = 1, \dots, J, \; i = 1, \dots, n$	(9)
$\sum_{r=1}^{n} x_{irmj} + \sum_{h=1}^{n} x_{khmj} \leq M(1 - z_{ikj})$	$; j = 1, \dots, J, \; i, k = 1, \dots, n, m = 1, .., m_j$	(10)
$\sum_{i=1}^{n} \sum_{r=1}^{n} z_{irj} \leq m_j$	$; \; j = 1, \dots, J$	(11)
$C_{max} \geq T_{iJ}$	$; i = 1, \dots, n$	(12)
$z_{irj} = \quad 0 \quad or \quad 1$	$; \; j = 1, \dots, J, \; i = 1, \dots, n-1, r = i+1, \dots, n$	(13)
$x_{irj} = \quad 0 \quad or \quad 1$	$; \; j = 1, \dots, J, \; i = 1, \dots, n-1, r = i+1, \dots, n$	(14)
$s_{i,j} \geq 0$	$; \; j = 1, \dots, J, \; i = 1, \dots, n, \quad r = i+1, \dots, n$	(15)
$T_{i,j} \geq 0$	$; \; j = 1, \dots, J, \; i = 1, \dots, n, \quad r = i+1, \dots, n$	(16)

Equation (1) denotes makespan as an objective function, Based on constraints (2) if two jobs have overlap in the process at a stage then the starting time of one of them must be between starting and departure time of other where M is a big positive number, constraints (3) gives precedence between jobs in feasible solutions, constraints (4) ensure that when some of jobs have overlap in the process at a stage the resources consumed by these jobs must be is equal to or less than the resource capacity of that stage at each time, constraints(5) and (6) calculate the departure time of job $\pi(i)$ at each stage in two conditions: buffer has free space and buffer is full, constraints (7) states that every job must wait until the next buffer has free space, constraints (8) expresses that each job can enter the next stage only after completing its process at the current stage, constraints (9) enforce that each job must be processed at every stage by only one machine, constraints (10) states that when the jobs have overlap in processing at same stage then they can't processed by same machine, constraints (11) determines maximum number of jobs which are overlapping in process at each stage, constraints (12) represents the makespan of each feasible solution and constraints (13) to constraints (16) define the types of decision variables.

3-Solution methodology: Hybrid genetic algorithm

From the perspective of buffer limitation, Papadimitriou and Kanellakis (1980) proved the strong NP-hardness of the two-machine problem with one capacity buffer and the makespan criterion. In addition, Hall and Sriskandarajah (1996) showed that the three-machine BFSP with the makespan criterion is strongly NP-hard. Almost all the cases related to the flow shop problems with resource constraints are NP-hard. For instance, Garey and Johnson (1975) and Blazewicz et al.(1986), respectively, proved that a resource constrained scheduling problem with two and three parallel machines, where process time of each job equals to one, is NP-hard. In this article, because of NP-completeness of the hybrid flow shop scheduling problem (even with two stages) (Gupta, 1988) and NP-hardness of the simpler version of the problem discussed (with two identical parallel machines) (Lenstra, 1977),the hybrid GA algorithm was applied for determining the machines that were used for processing the jobs and to obtain the global optimum of the scheduling problem. In this method, variable neighborhood search was used for increasing the efficiency of the local search of the algorithm before selecting the phases of the genetic algorithm. The procedure of the hybrid GA algorithm is shown in figure 2. Details about this algorithm are given in the subsequent sub-sections but the general stepwise outline is described below:

Step 1: Set GA parameters such as population size, crossover, mutation rates and stopping criteria.
Step 2: Generate initial population.
Step 3: If stopping criteria are fulfilled: stop algorithm, else: go to step 4.
Step 4: Apply VNS for selecting best neighbors of the current population.
Step 5: Select parents for generating new population according to their fitness functions.
Step 6: Apply crossover operators.
Step 7: Apply mutation operators.
Step 8: Replace current population with a new generation.
Step 9: Return to step 3.

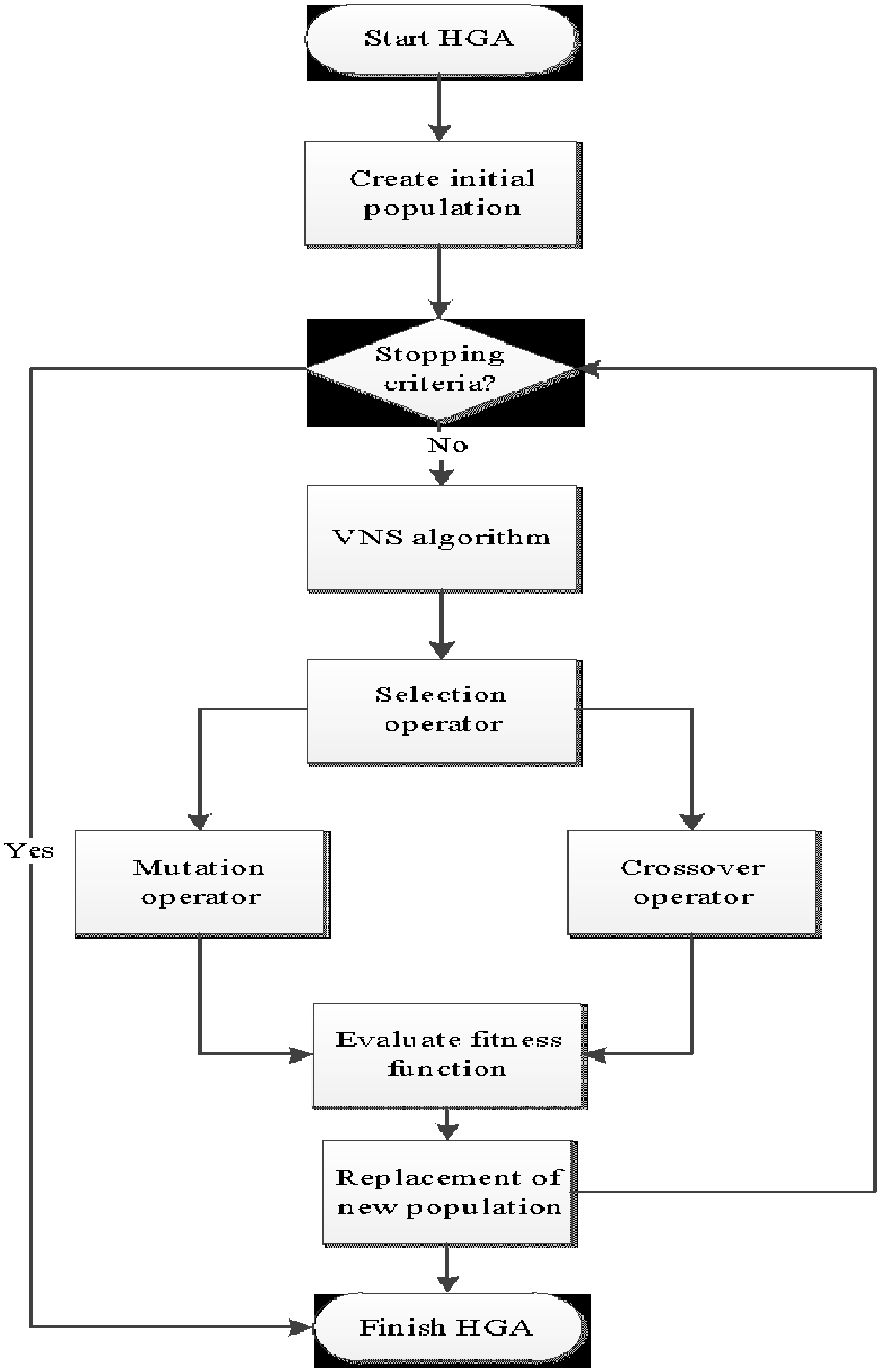

Figure 2. Hybrid GA workflow

3-1- Genetic operators

Genetic algorithm generally has three operators: selection, crossover, and mutation. It is crucial to suitably set these operators. In this paper, the chromosomes are considered as a permutation of the operations. This means that a number is assigned to all operations of all the jobs at each stage. If there are 10 jobs and 9 stages, then the total number of operations is 90. Every number between 1 and 90 represents an operation. Hence, a sequence of job can be represented by the permutation of these numbers where each number corresponds to specific job processing at a specific stage (Figure 3).

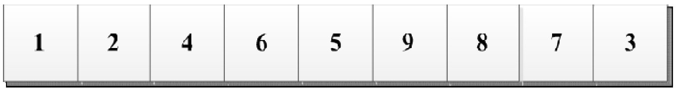

Figure 3. Solution representation

3-2- Selection

In GA, the selection operator is used to select individuals according to fitness (Li & Gao, 2016). In this paper, Roulette wheel method was used for selection of parent to create the offspring. Roulette wheel approach allows a tradeoff between exploration and exploitation. We Apply VNS with GA to improvement search in the solution space to find the parents who have better fitness for creation a better generation. In this study, VNS used two kinds of neighborhood search: (i) the scramble operator which randomly selects two positions in the permutation and reorders all the positions between them randomly (Back, Fogel, & Michalewicz, 2000) (Figure 4) and (ii) the insertion operator that selects two positions in the permutation randomly and inserts them together, as illustrated in figure 5(Gen & Cheng, 1997).

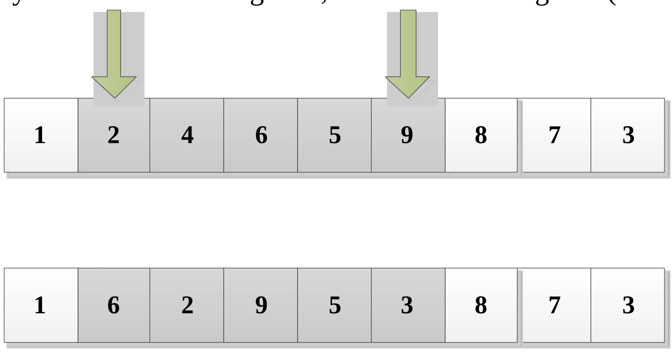

Figure 4. Scramble operator

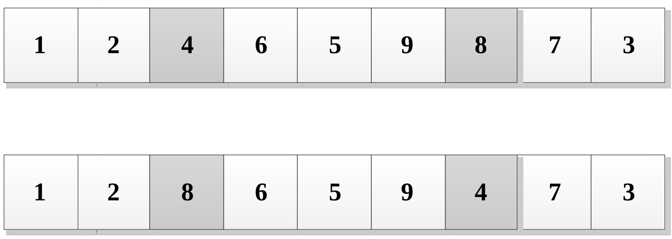

Figure 5. Insertion operator

3-3- Crossover

In this paper, three different motion operators were used: Single-point, Two-point, and Mask. These operators are chosen by a random number. The process of the crossover operators is shown in figure 6 to figures 8.

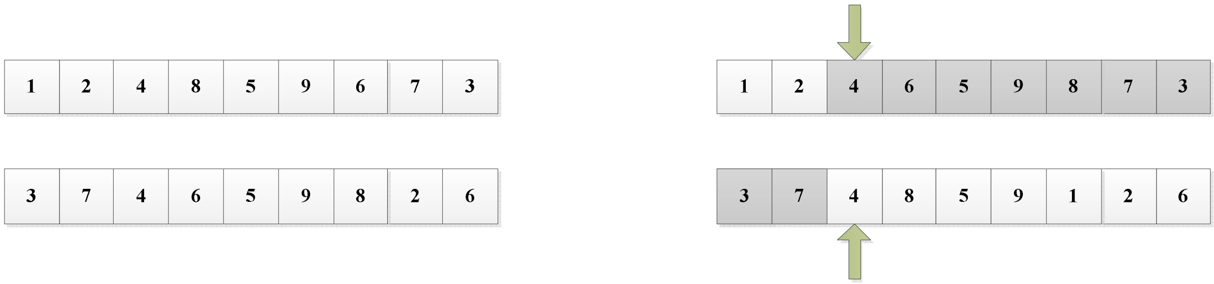

Figure 6. Single-point crossover

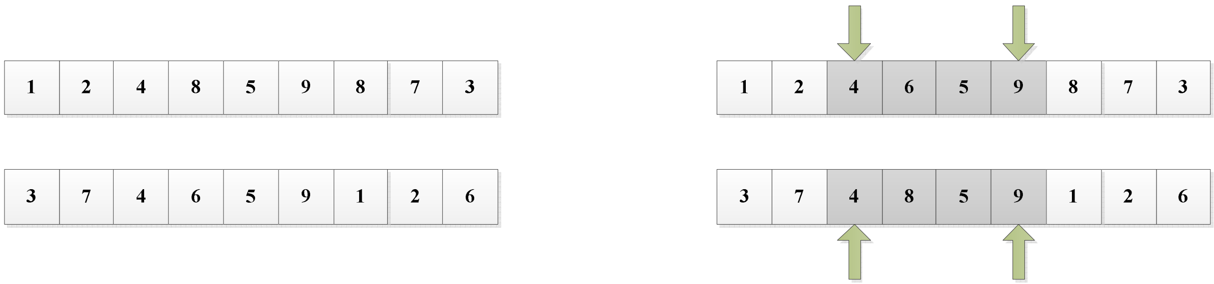

Figure 7. Two-point crossover

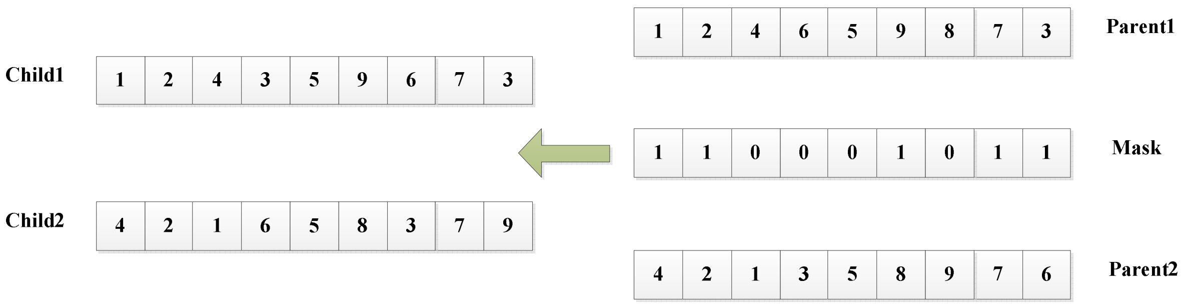

Figure 8. Mask crossover

3-4- Mutation

Mutation operators play an important role in escaping from the local optimum and making a random change in the search space (Abdollahpour & Rezaeian, 2015). In this study, two operators were applied for mutating the parents: Swap and reversion (Figures 9 and 10).

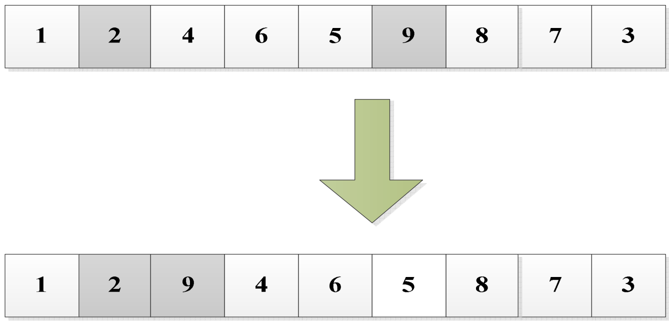

Figure 9. Swap mutation

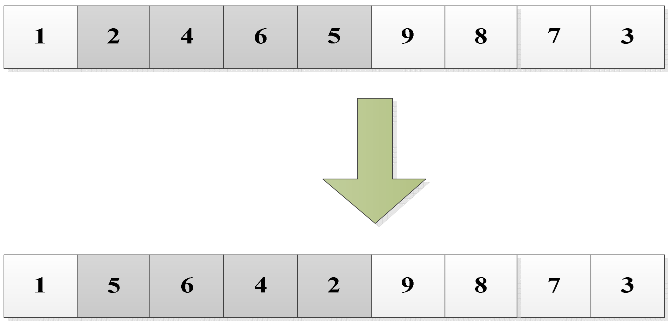

Figure 10. Reversion mutation

4- Case study

The case study is about a tire manufacturing company in Iran that has 30% market share in heavy tire market segment. All of tire production stages are done with automation and computer aided manufacturing. Although tires manufacturing processes regarding to their types (heavy or light) and used technology (bias or radial) have some differences but in general we can categorize them to five following stages:

i. Master batch production: in this stage, all chemical materials are mixed in the mixer machine (ban bury) for producing “master” batches based on specific sequence. These materials include rubber (natural and synthetic), carbon blacks, oils and other chemical products.
ii. Final batch production: the master batches must mix with sulphur in the ban buries that is called final ban buries. The output of this stage is called “final” batch and used in different objectives at next stages.
iii. Extruder: one of the key elements of tires construction is tread. Tread covers the tire plies and has friction with ground when automotive is in motion. The treads composed from two or more different of final batches depending on type of technology that used for extruding of rubbers.
iv. Green tire building: in tire building process all rubber plies are assembled together with specific timing and instructions and finally tread installed on the plies. The final production of this stage is called green tire
v. Tire curing: the final stage of tires manufacturing those green tires attains the final shape and mechanical-dynamical specification is tire curing or tire vulcanization. Hence, in this stage, the green tires puts in curing press machines and formed by molds heat and pressure.

According to above description the figure 11 represents a schematic relationship between tire manufacturing stages and table 2 is assigned number to each stage.

Table 2. Stage numbers

Stage number	Stage description
1	Master banbury
2	Final banbury
3	Extrusion
4	Tire building
5	Curing

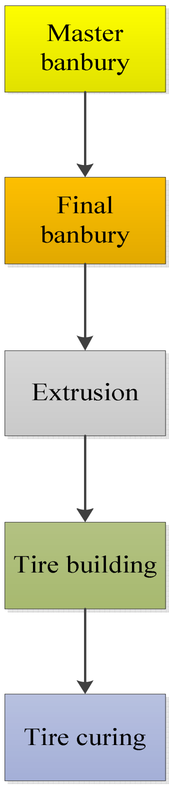

Figure 11. Schematic relationship between tire manufacturing stages

4-1- Data of the case study

In this case study, every stages of one to three have 2 machines with two different speeds; stage four has 20 parallel machines that type1 and type2 include 17 and 3 identical machines and finally, stage five has 55 parallel machines with 25 identical machines of type1 and 30 identical machines of type2. More details about processing time for each size of ordered tires on every machines at each stage is as table 3. This tire company manufactures its products according to the orders shown in table 4. This model simulated for four orders among the various customers; automotive company, two sell dealers, and one real person and horizontal time is one working day with three eight hour shifts.

Table 3. The database of processing times

Stage number	Machine/Type number	1	2	3	4
1	1	55	32	55	55
	2	53	30	53	53
2	1	51	30	51	51
	2	50	31	50	50
3	1	22	22	22	22
	2	26	26	26	26
4	1	1239	1239	1239	1239
	2	1229	1229	1229	1229
5	1	3920	3200	4340	4340
	2	3915	3195	4335	4335

Table 4. Order quantity

Product (Job)	1	2	3	4
Quantity	25	25	25	25

5- Results

We consider the case study as a hybrid flow shop problem with 5 stages and 4 types of resources. Other characteristics of the numerical example are as listed in table 5.The proposed algorithm was run with MatLab on a 2.5 GHz processor with 1 G-Byte RAM for testing and evaluation. To achieve improved robustness of the algorithms without producing functional variance under the influence of external environment, the parameter design can be applied to the process design. Taguchi's method describes that the optimal operator combination is to minimize the quality characteristic variances resulting from the S/N ratio, which explains why the parameter design is called the robust design. The data required for the HGA consists of the population size (npop), crossover rate (c_p) and mutation rate (m_p). Table 6 shows the factors and their levels. To consider the effect of varying levels on averaged S/N ratio, the main effect plot was used in the results of average of fitness function for the problem. Based on the results, the population size, crossover and mutation rate were set to 40, 0.5 and 0.4, respectively.

Table 5. Parameters of numerical example

Parameters	1	2	3	4	5
Number of machines	2	2	2	20	55
Buffer capacity	100	50	70	65	90
Capacity of electricity power (KW/h)	1000				
Capacity of Water (m^3)	80				
Capacity of steam (bar)	20				
Capacity of compressed air (bar)	18				

Table 6. Factor levels

Factor	Level index	Level
Population size (npop)	1	20
	2	30
	3	40
Crossover rate (c_p)	1	.5
	2	.6
	3	.7
Mutation rate (m_p)	1	.5
	2	.4
	3	.3

The full factorial design for the above three factors required 3^3=27 experiments of the algorithm. To determine optimum parameter levels, these experiments were conducted at the three scales of the problem. The problem was repeated 9 times. Therefore, the total number of required experiments for the factorial designs was 27×9=243 but considering cost and time, this type of an experimental design was not economical. In statistical theories, it is not required to experiment all the combinations of the factors. Hence, fractional replicated designs are used. The robust parameter design uses Taguchi designs (orthogonal arrays) which allowed us to analyze many factors with a few runs only.

5-1- Analysis of the performance ratios of S/N for quality characteristics

The S/N ratio suggests the best and strangest conditions with using the variations in experiments results. This ratio expresses the variation around a specific value and the higher level for this ratio is equivalent of less variation. S/N value for each quality characteristic of problem is shown in table 7. To consider the effect of varying levels on averaged S/N ratio, the main effect plot was used. Therefore, the total S/N ratio and its average are compiled in table 8 and shown in figure 12.

Table 7.S/N Value for GA parameters

Control parameters				SNRA
Run Order	npop	c_p	m_p	
1	20	0.5	0.5	-55.8478
2	20	0.6	0.4	-55.7634
3	20	0.7	0.3	-56.1236
4	30	0.5	0.4	-55.7916
5	30	0.6	0.3	-55.9454
6	30	0.7	0.3	-55.9730
7	40	0.5	0.3	-55.8758
8	40	0.6	0.3	-55.9592
9	40	0.7	0.3	-55.7634

Table 8. Average levels of S/N

Levels	npop	c_p	m_p
1	-55.91	-55.84	-55.94
2	-55.90	-55.89	-55.78
3	-55.87	-55.95	-55.85

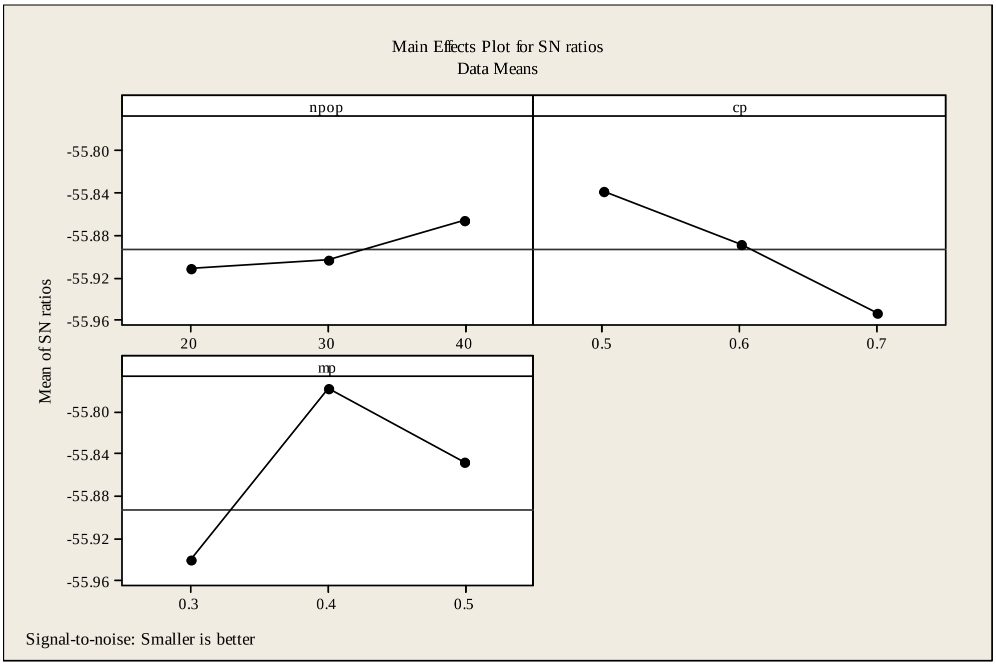

Figure 12. Effect of factors on the performance of S/N

5-2- Analysis of the fitness function

To determine the optimum design of GA parameters, the average of fitness function must be considered after the analysis of S/N ratio. Whenever the S/N ratio for different levels of the parameter has a negligible difference, the optimum level of the parameter is determined using the average of data in the levels. The average ratios of mean among all the levels were calculated and are shown in figures 13.

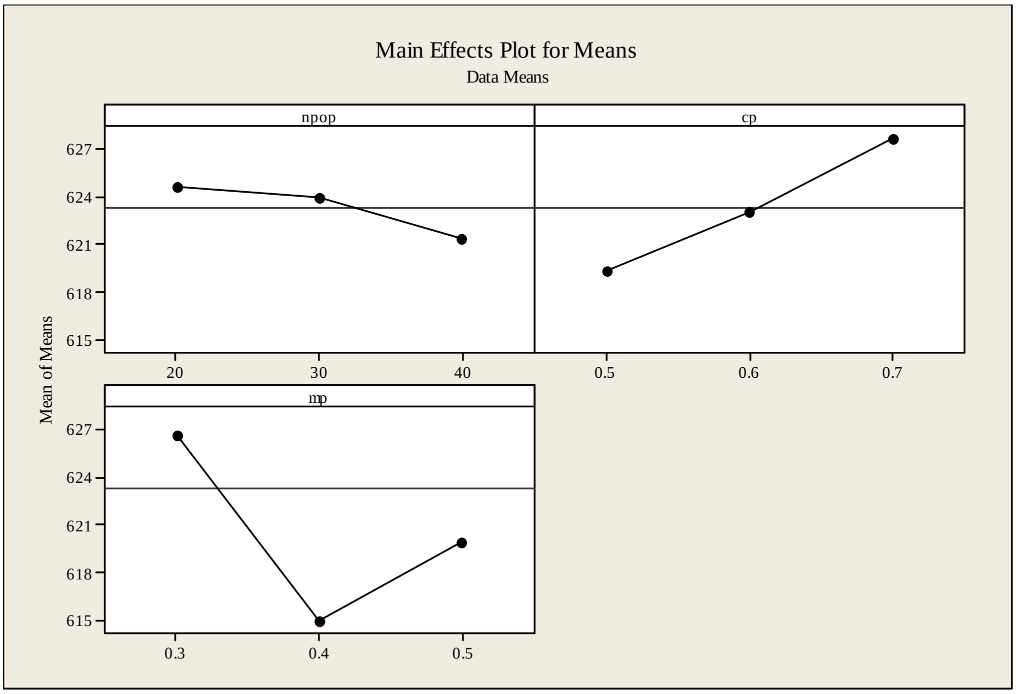

Figure 13. Effect of factors on the fitness function

The results of optimal design for GA parameters according to the S/N ratio and mean of the fitness function for problem is shown in table 9. It can be seen that there is no significant difference between the parameter levels for all the problems.

Table 9. Optimum design of parameters

Factor	Optimum level
npop	40
c_p	.5
m_p	.4

The algorithm converges after 73 iterations to 60300. Since the dimension of processing times in this case study is based on 100 seconds then we can say the makespan of production for this order is about 16.75 hours that can be processed at one working day and subsequently, placed at intended horizontal time. Figure 14 represents the trends of best and mean of solutions repetitions of algorithm.

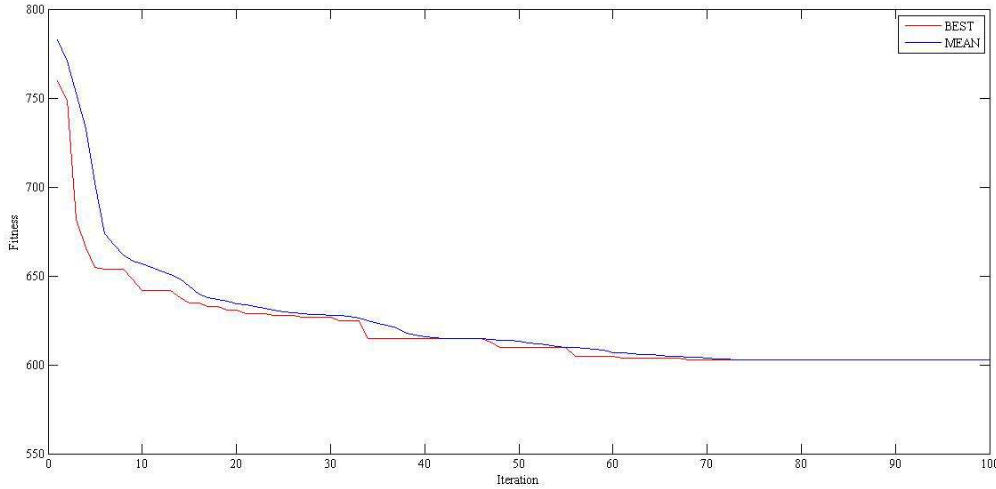

Figure 14. Best vs. mean of the solutions

6- Discussion

The case study discussed above gives the following managerial insights:

- In some energy-intensive industries such as the tire manufacturing, existence of sufficient energy is critical for processing, especially in the curing stage. In some cases (at this stage), simultaneous processing causes a sudden fault in the evaporating supply system (such as boilers) which may lead to final products being damaged. On applying the proposed model, such companies will be able to schedule the curing machines so as to avoid this problem.
- In many industries, assigning optimum space for intermediate buffers, due to budgeting and physical constraints plays an important role in designing the production lines. With the proposed model, these constraints can be considered while scheduling and estimating the required space.
- One of the major problems in production scheduling is to increase the semi productions inventory and consequently enhance the working capital cost. The proposed model enables the organization to reduce this cost with an emphasis on minimizing the makespan of products.
- Sometimes due to adequate resources, the companies have to reduce their energy usage. For example in the warm climatic regions of Iran, most of the companies are forced to reduce their electrical power consumption to prevent interruption in electrical supplies for household consumptions. In such cases, the organization is liable to pay fine if it offends the government's directives. Using the proposed mathematical model, these types of organizations can consider the constraints in production scheduling.

7- Conclusion

The main idea in this study is to simultaneously deploy two constraints on the capacity of buffers and resources due to which there arises a need to formulate a new mathematical model for the scheduling problem for a real-case study. Using a real case study we demonstrate the applicability of the proposed model. The proposed model has a strong potential for applications in other real manufacturing environments such as textile or automotive industries. In most of the manufacturing systems, machines require specific resources for job processing. When the available capacity of resources is limited, some jobs have to wait in the buffers for processing. On the other hand, because of limited buffer space, it is possible that there is not enough space for the storage of jobs. In this situation, the jobs must remain on the machine until some space is released in the buffer of the next stage. According to NP-hardness of the problem, a hybrid genetic algorithm hybridized with variable neighborhood search (VNS) was used for solving this problem. Genetic algorithm typically has a slow convergence speed due to which a method is required for improving the neighborhood search. Using VNS method improves the exploitation of the algorithm and helps in selecting the best solutions for transfer to the next generation. The statistical results clearly show that the quality of obtained solutions is acceptable for the case study; hence the hybrid GA has a high performance.

There are several future directions to pursue further studies in these problems. The first one is to consider other types of constraints such as availability of machines in the problem and modeling with them. The second one is adding few assumptions to the problem for instant setup times for the jobs. Also, this problem can be formulated with multi-objective function.

References

Abdollahpour, S., & Rezaeian, J. (2015). Minimizing makespan for flow shop scheduling problem with intermediate buffers by using hybrid approach of artificial immune system. *Applied Soft Computing, 28*, 44–56.

Afzalirad, M., & Rezaeian, J. (2016). Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. *Computers & Industrial Engineering, 98,* 40–52.

Allaoui, H., & Artiba, A. (2006). Scheduling two-stage hybrid flow shop with availability constraints. *Computers & Operations Research, 33,* 1399–1419.

Back, T., Fogel, D. B., & Michalewicz, Z. (2000). *Evolutionary Computation 1: Basic Algorithms and Operators* (Vol. 1). New York, USA: Tylor and Francis Group.

Belaid, R., T'kindt, V., & Esswein, C. (2012). Scheduling batches in flowshop with limited buffers in the shampoo industry. *European Journal of Operational Research, 223*(2), 560–572.

Blazewicz, J. (1978). *Complexity of computer scheduling algorithms under resource constraints.* Paper presented at the Proceedings of the First Meeting AFCET-SMF on Applied Mathematics.

Blazewicz, J., Barcelo, J., Kubiak, W., & Röck, H. (1986). Scheduling tasks on two processors with deadlines and additional resources. *European Journal of Operational Research, 26*(3), 364–370.

Blazewicz, J., & Ecker, K. (1983). A linear time algorithm for restricted bin packing and scheduling problems. *Operations Research Letters, 2*(2), 80–83.

Blazewicz, J., Ecker, K. H., Pesch, E., & Schmidt, G. (2007). *Handbook on Scheduling*. New York: Springer-Verlag.

Błażewicz, J., Kubiak, W., Röck, H., & Szwarcfiter, J. (1987). Minimizing mean flow-time with parallel processors and resource constraints. *Acta informatica, 24*(5), 513-524.

Brucke, P., & Kramer, A. (1996). Polynomial algorithms for resource-constrained and multiprocessor task scheduling problems. *European Journal of Operational Research, 90*, 214–226.

Daniels, R. L., Hoopes, B. J., & Mazzola, J. B. (1996). Scheduling parallel manufacturing cells with resource flexibility *Management Science, 42*(9), 1260–1276.

Edis, E. B., & Oguz, C. (2011). *Parallel machine scheduling with additional resources: a Lagrangian-based constraint programming approach*. Berlin, Germany: Springer-Verlag.

Edis, E. B., & Oguz, C. (2012). Parallel machine scheduling with flexible resources. *Computers and Industrial Engineering, 63*, 433–447.

Edis, E. B., & Ozkarahan, I. (2011). proposed a combined integer/constraint-programming approach to a resource-constrained parallel machine scheduling problem with machine eligibility restrictions. *Engineering Optimization, 43*(2), 135–157.

Fanjul-Peyro, L., Perea, F., & Ruiz, R. (2017). Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources. *European Journal of Operational Research*.

Fu, Y., Wang, Z., Zhang, J., & Wang, Z. (2017). A blocking flow shop deteriorating scheduling problem via a hybrid chemical reaction optimization. *Advances in Mechanical Engineering, 9*(6), 1687814017701371. doi:10.1177/1687814017701371

Garey, M. R., Johnson, D.S. (1975). Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing, 4*(4), 397–411.

Gen, M., & Cheng, R. (1997). *Genetic Algorithms and Engineering Design* (Vol. I). New York, USA: John Wiley & Sons.

Grigoriev, A., Sviridenko, M., & Uetz, M. (2007). Machine scheduling with resource dependent processing times. *Mathematical programming, 110*(1), 209-228.

Gupta, J. N. D. (1988). Two-Stage, Hybrid Flowshop Scheduling Problem. *Journal of the Operational Research Society, 39*(4).

Hall, N. G., & Sriskandarajah, C. (1996). A Survey of Machine Scheduling Problems with Blocking and No-Wait in Process. *Operations Research, 44*(3), 510 - 525.

Kellerer, H., & Strusevisch, V. A. (2008). Scheduling parallel dedicated machines with the speeding-up resource. *Naval Research Logistics, 55*(5), 377–389.

Kovalyov, M. Y., & Shafransky, Y. M. (1998). Uniform machine scheduling of unit-time jobs subject to resource constraints. *Discrete Applied Mathematics, 84*, 253–257.

Laribi, I., Yalaoui, F., Belkai, F., & Sari, Z. (2016). *heuristics algorithm for solving flow shop scheduling problem under resources constraints.* Paper presented at the 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM, Troyes.

Lenstra, J. K., Rinnooy Kan, A. H. G., and Brucker, P.,. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics, 1*(1), 343-362.

Li, X., & Gao, L. (2016). An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. *Int. J.ProductionEconomics, 174*, 93–110.

Liu, S. Q., & Kozan, E. (2009). Scheduling a flow shop with combined buffer conditions. *International Journal of Production Economics, 117*(2), 371-380. doi:https://doi.org/10.1016/j.ijpe.2008.11.007

Liu, S. Q., & Kozan, E. (2016). Parallel-identical-machine job-shop scheduling with different stage-dependent buffering requirements. *Computers & Operations Research, 74*, 31–41.

Liu, X., Li, K., & Ren, H. (2015). A Hybrid Algorithm for the Permutation Flowshop Scheduling Problem without Intermediate Buffers. *Discrete Dynamics in Nature and Society, 2015*.

Lovaa, A., Tormos, P., Cervantes, M., & Barber, F. (2009). An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. *International Journal of Production Economics, 117*(2), 302–316.

Mansouri, S. A., & Aktas, E. (2016). Minimizing energy consumption and makespan in a two-machine flowshop scheduling problem. *Journal of the Operational Research Society, 67*(11), 1382–1394.

Moslehi, G., & Khorasanian, D. (2014). A hybrid variable neighborhood search algorithm for solving the limited-buffer permutation flow shop scheduling problem with the makespan criterion. *Computers & Operations Research, 52*, 260–268.

Papadimitriou, C., & Kanellakis, P. (1980). Flow shop scheduling with limited temporary storage. *Journal of the ACM, 27*(3), 533-549.

Pinedo, M. L. (2016). *Scheduling: Theory, Algorithms, and Systems*: Springer International Publishing.

Ribas, I., Companys, R., & Tort-Martorell, X. (2017). Efficient heuristics for the parallel blocking flow shop scheduling problem. *Expert Systems with Applications, 74*(Supplement C), 41-54. doi:https://doi.org/10.1016/j.eswa.2017.01.006

Ruiz-Torres, A. J., Lopez, F. J., & Ho, J. C. (2007). Scheduling uniform parallel machines subject to a secondary resource to minimize the number of tardy jobs. *European Journal of Operational Research, 179*(2), 302–315.

Ventura, J. A., & Kim, D. (2000). Parallel machine scheduling about an unrestricted due date and additional resource constraints. *Iie Transactions, 32*(2), 147-153.

Waldherr, S., & Knust, S. (2016). Decomposition algorithms for synchronous flow shop problems with additional resources and setup times. *European Journal of Operational Research*.

Yaurima, V., Burtseva, L., & Tchernykh, A. (2009). Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. *Computers & Industrial Engineering, 56*(4), 1452-1463. doi:https://doi.org/10.1016/j.cie.2008.09.004