Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-12 (2016) 1478–1483

Heuristics for solving flow shop scheduling problem under resources constraints

Imane Laribi * **Farouk Yalaoui** ** **Fayçal Belkaid** * **Zaki Sari** ***

* *Manufacturing Engineering Laboratory of Tlemcen, University of Tlemcen B. P. N119, Tlemcen 13000, Algeria (e-mail: im.laribi@gmail.com,f_belkaid@yahoo.fr)*

** *ICD -LOSI (UMR 6281), Universit de Technologie de Troyes 12 rue Marie Curie, cs42060, 10004 Troyes Cedex, France (e-mail: farouk.yalaoui@utt.fr)*

*** *Department of Industrial Engineering, Izmir University of Economics, Izmir, 35330 Turkey (z_sari@mail.univ−tlemcen.dz)*

Abstract: Most of traditional scheduling problems deal with machine as the only resource, however, other resources such as raw materials is often disregards. Considering the second resource makes scheduling problems more realistic and practical to implement in manufacturing industries. Due to the applicability of flow shop environment in different manufacturing, scheduling of these types of shops are extensively studied by several authors. However, introducing an additional resource in this environment is not well studied. The present work deals with makespan minimization in flow shop scheduling problems where no renewable resources constraints are considered. The paper illustrates the importance of Johnson (1954) algorithm for the two machine flow shop under resources constraints. A mathematical model is also presented. Then a well-know heuristic is adapted to propose fast solution for the m machines flow shop problem subject to resource constraint.

Keywords: Scheduling, flow shop, heuristics, no renewable resource, makespan.

1. INTRODUCTION

Managing the use of non-renewable or consumable resources has become an important issue in many different settings. The term non-renewable implies that resources are limited for the whole time horizon of scheduling i.e. their availability may be sufficient or not, therefore, machine will stay idle for a period of time only if there are not enough quantities from the required resources when starting the job. As a result, the efficiency of scheduling will be affected. Scheduling under this constraint is an exciting field with enormous practical importance. In this context jobs have to be processed on machine in different environment while also respecting the availability of some non renewable resources that will be consumed by the various jobs but replenished at given dates from external resources with known quantities. The research in the area of scheduling problems with non renewable resources is rather limited. Carlier and Kan (1982) developed a polynomial time algorithms for various special case in order to solve precedence constraint scheduling in with each job requires certain amount of non-renewable resources. Slowinski (1984) presented a polynomial algorithm for preemptive problem on unrelated parallel machines with some renewable and non-renewable resources. Toker et al. (1991) studied the single machine non renewable resource constrained problem, they shown that when the amount of resource available at each time period is constant the problem is polynomially solvable. Some basic complexity results on a single machine subject to non renewable resources constraints has been studied by Grigoriev et al. (2005), the authors provided also approximation algorithms for the hard problems. Carrera (2010) studied a single machine consumable resources problem with staircase availability of resources, an integer linear program, branch and bound procedures are proposed. Belkaid et al. (2013) proposed a genetic algorithm to minimize the makespan for parallel machines problem with consumable resources. Laribi et al. (2014) studied hybrid flow shop scheduling research by considering the problem with consumable resources, the objective is the minimization of the makespan with one common testing machine at the first stage and identical parallel machine at second stage, several heuristics based on priority dispatching rules are compared and evaluated. In this paper, we extend flow shop scheduling research by considering the constraint of no renewable resources. Since Johnson algorithm is optimal for two machine flow shop, the main purpose of this paper is to calculate the performance of this algorithm as a heuristic applied to the two machine flow shop subject to no renewable resources constraints on the second machine; see section 4. Then we move from two machine to m machine flow shop problem (where m denote the number of machines) by adapting Nawaz et al. (1982) heuristic; see section 5. Problem formulation is given in section 2. Section 3 provides a mathematical model for the proposed problem. Some interesting conclusions and future studies are given in section 6.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.780

2. PROBLEM DESCRIPTION

This paper deals with the flow shop scheduling problem subject to resource constraints, the resource is typically no renewable (such as raw materials, money...). The problem is to find a schedule to minimize the maximum completion time, or makespan, denoted by C_{max}. The problem under study can be described as follow. There are m machines $M1, M2, ..., Mm$ available from time zero for processing a set of n independents jobs $J1, J2, ..., Jn$. each job consists of m operations $Q_{ij}(i = 1, 2, ..., n, j = 1, 2, ..., m)$, which have to be processed by the machines in the same technology order. Besides machines, jobs for their processing on the last machine require additional non renewable resources. There are R types of non renewable resources, the resource of type $(l = 1, ..., R)$ is available in limited quantities at a specified time. The general form includes the following assumptions.

- Pre-emption is not allowed, i.e. a job once started on a machine, continues in processing until it is completed
- All jobs are independent and available for processing at time zero
- Jobs are allowed to wait between two machines, and the storage is unlimited
- The machines are continuously available from time zero onwards (no breakdowns).
- Each machine can process at most one job at a time, and each job can be processed only on one machine at a time.
- All jobs are assumed to be processed on each machine on the same order (permutation flow shop).
- The total resource requirements never exceed the resource capacity.
- Each job consumes a limited amount of resource l at the beginning of their processing in the last machine.
- Each job is assumed to be available for processing when all necessary quantities of resources are available.
- The delivery of each resource is represented as a cumulated stair curve.
- All parameters are deterministic.

The problem can be depicted by fig. 1

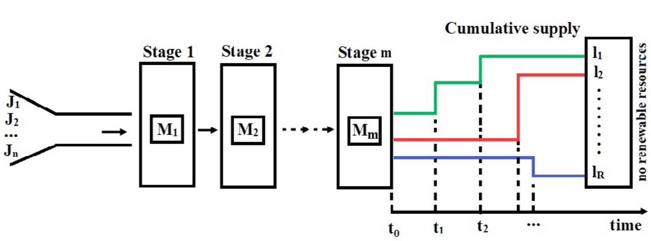

Fig. 1. Problem presentation.

Following the classification scheme used by Toker et al. (1991) and Xie (1997), we denote the problem considered in this paper by $Fm|NR : \alpha_{lt}|C_{max}$, where $NR : \alpha_{lt}$ indicates the existence of non renewable resources, that become available at an amount of α_{lt} in period t (l refers to the resource type). Since $Fm||C_{max}(form > 2)$ is NP-hard, $Fm|NR : \alpha_{lt}|C_{max}$ is obviously NP-hard.

3. INTEGER LINEAR PROGRAMMING FORMULATION

In order to determine an optimal solution for the flow shop problem under a non renewable resources constraints, we propose an integer linear model to minimize the makespan, this modeling is based on the model presented by Carrera (2010) to minimize makespan in the environment of single machine with consumable resource constraint. In the following, we provide an integer linear program formulation of the $Fm|NR : \alpha_{lt}|C_{max}$ problem using positional variables.

3.1 Notations

In order to present a mathematical model for problem described previously, we define several notations. These notations are explained below.

Table 1. Parameters

Parameter	Description
n	Number of jobs to be scheduled
m	Number of machines in the system
R	Number of resources
P_{ij}	Processing time of job i on machine j
Q_{il}	Quantity of resource l that job i consumes
Z_{tl}	Total amount of resource l arrived at time t
M	Large positive number
D	Time of the last delivery

Table 2. Indices

Index	For	Scale
i	Jobs	1...n
j	Machines	1...m
l	Resources	1...R
k	Position	1...n
t	Time	1...D

Table 3. Decision variables

Variable	Description
C_{kj}	Completion time of job in position k on machine j
S_{kj}	Start time of job in position k on machine j
W_{kl}	The amount of resource l required to process job in position k
X_{ik}	Binary variable that takes value 1 if job i is processed on position k, and 0 otherwise
Y_{kt}	Binary variable that takes value 1 if $S_{kj} \geq t$, and 0 otherwise

3.2 Model

Using the notation given above, the objective function and the constraints of the proposed mathematical model for the flow shop problem under non renewable resources are presented below.

$$Minimize\ C_{nm} \tag{1}$$

Objective function (1) seeks for minimization of makespan criterion. Subject to

$$\sum_{i=1}^{n} X_{ik} = 1 \ \forall k \in \{1, ..., n\} \tag{2}$$

$$\sum_{k=1}^{n} X_{ik} = 1 \ \ \forall i \in \{1,...,n\} \tag{3}$$

$$C_{kj} \geq C_{(k-1)j} + \textstyle\sum_{i=1}^{n} X_{ik}.P_{ij} \quad \forall j \in \{1,...,m\} \, \forall k \in \{2,...,n\} \tag{4}$$

$$C_{kj} \geq C_{k(j-1)} + \textstyle\sum_{i=1}^{n} X_{ik}.P_{ij} \quad \forall j \in \{2,...,m\} \, \forall k \in \{1,...,n\} \tag{5}$$

$$C_{1j} \geq C_{1(j-1)} + \sum_{i=1}^{n} X_{i1}.P_{ij} \ \ \forall j \in \{2,...,m\} \tag{6}$$

$$C_{11} = \sum_{i=1}^{n} X_{i1}.P_{i1} \tag{7}$$

$$S_{kj} = C_{kj} - \sum_{i=1}^{n} X_{ik}.P_{ij} \ \ \forall j \in \{1,...,m\} \, \forall k \in \{1,...,n\} \tag{8}$$

$$W_{kl} = \sum_{i=1}^{n} X_{ik}.Q_{il} \ \ \forall k \in \{1,...,n\} \, \forall l \in \{1,...,R\} \tag{9}$$

$$\sum_{v=1}^{k} W_{vl} \leq \sum_{t=1}^{D} Z_{tl}.Y_{kt} \ \ \forall k \in \{1,...,n\} \, \forall l \in \{1,...,R\} \tag{10}$$

$$M(Y_{kt} - 1) \leq S_{km} - t \ \ \forall k \in \{1,...,n\} \, \forall t \in \{1,...,D\} \tag{11}$$

$$C_{kj} \geq 0 \ \ \forall k \in \{1,...,n\} \, \forall j \in \{1,...,m\} \tag{12}$$

$$S_{kj} \geq 0 \ \ \forall k \in \{1,...,n\} \, \forall j \in \{1,...,m\} \tag{13}$$

$$W_{kl} \geq 0 \ \ \forall k \in \{1,...,n\} \, \forall j \in \{1,...,m\} \tag{14}$$

$$X_{ik} \in \{0,1\} \ \ \forall i \in \{1,...,n\} \, \forall k \in \{1,...,n\} \tag{15}$$

$$Y_{kt} \in \{0,1\} \ \ \forall k \in \{1,...,n\} \, \forall t \in \{1,...,D\} \tag{16}$$

Constraint (2) ensures that every job must occupy exactly one position, constraint (3) forces that each position must be assigned exactly once. Constraint (4) ensures that each job can start only after the previous job assigned to the same machine (from the second order of sequencing). Constraint (5) controls that the processing of job in position k on machine j can only start when the processing of the same job on the previous machine is finished. Constraint (6) presents the completion time of job placed in position one on machine j, that is greater or equal the completion time of job placed in position one on the production scheduling on machine $j-1$. The completion time of job sequenced in position 1 on machine 1, is represented by constraint (7). Constraint (8) calculates the start time of job in position k on machine j. Constraint (9) calculates the quantity of resources l consumed by job in position k. Constraint (10) verifies that, the total quantity of resource l consumed by job at position k is less than or equal to the available quantity of this resource. Constraint (11) makes link between the decision variable Y_{kt} and the start time of job on position k. Constraints (12) and (13) ensures that the completion time, starting time of jobs at any position cannot be negative. Constraint (14) ensures that the amount of resource required to process a job must be positive. Constraint (15) X_{ik} is a Boolean variable. It is equal to 1 if job i is processed on position k, and 0 otherwise. Constraint (16) Y_{kt} is a Boolean variable variable that takes value 1 if S_{kj} ≥t, 0 otherwise.

4. JOHNSON ALGORITHM FOR $F2|NR:\alpha_{lt}|C_{max}$ PROBLEM

Johnson (1954) algorithm is the most important result for the flow shop problem which has now become a standard in theory of scheduling, its primary objective is to reduce the idle time on the second machine in order to find an optimal sequence of jobs which the minimum makespan. Let now consider a two machine flow shop such that the second machine is subject to the availability of no renewable resources, the objective is to minimize the makespan, we try to adapt Johnson algorithm to perform this problem with respecting the availability of no renewable resources constraints. The problem $F2|NR:\alpha_{lt}|C_{max}$ is clearly NP-hard, because a special case $1|NR|C_{max}$ Gafarov et al. (2011) is already NP-hard.

Algorithm 1 Johnson algorithm for $F2|NR:\alpha_{lt}|C_{max}$

Inputs:
  $U = \{1...n\}$ set of unscheduled jobs.
  $U' = \{1...R\}$ set of resources.
  $V_{n\times 2}$ processing time matrix of n-job, two machines.
Initialize:
  $a \leftarrow 0$, $b \leftarrow n$
for $i = 1$ to n **do**
  for $j = 1$ to 2 **do**
    if $U = \emptyset$ **then**
      STOP
    else
      Find $e = min(minP_{i1}, minP_{i2})$
      if $e = P_{*1}$ **then**
        Schedule job∗ on the earliest position;
        $a \leftarrow a + 1$;
        $U \leftarrow U - \{J*\}$;
      else
        Schedule job∗ on the latest position;
        $b \leftarrow b - 1$;
        $U \leftarrow U - \{J*\}$;
      end if
    end if
  end for
end for
Allocate the sequenced jobs on the first and second machine respectively
for $l \in U'$ **do**
  for $k = 1$ to n **do**
    if $\sum_{v=1}^{k} W_{vl} \geq \sum_{t=1}^{H} Z_{tl}.Y_{kt}$ **then**
      Job∗ must wait until the resources will be available.
    end if
  end for
end for

4.1 Example

In order to illustrate this algorithm. Considering the following instance of a car painting factory, on the first machine we have a car degreasing and on the second machine a car painting, for painting a car we need paint that is no renewable resource. Processing time and resource requirement are presented in table 4 and table 5 respectively, the amount of paint arrived at time t is represented by figure

5 and the gantt chart of the optimal schedule is depicted by figure 3.

Table 4. Data for 2−machine 4−job problem

	Car1	Car2	Car3	Car4
Degreasing	5	2	4	3
Painting	2	1	3	4

Table 5. Data for paint requirement

	Car1	Car2	Car3	Car4
Paint needs	2	1	2	3

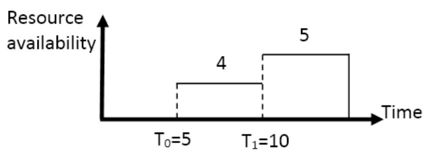

Fig. 2. Cumulated staircase curves for one resource delivery

The optimal sequence for this instance using Johnsons rule is as follow ($car4 \rightarrow car3 \rightarrow car1 \rightarrow car2$).
The schedule for this sequence is presented in fig. 3.

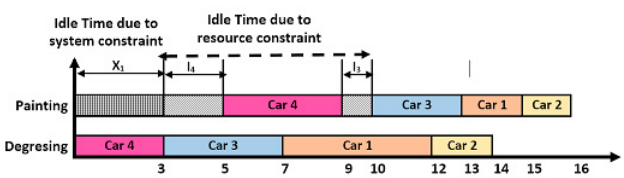

Fig. 3. Gantt chart of the optimal schedule

C_{max} = Sum of idle time due to the unavailability of resource + sum of idle time due to a system constraint (see constraint 4 and 5) + Sum of processing time on the second machine.
$C_{max} = (I_4 + I_3) + X_1 + \sum_{i=1}^{n} P_{i2}$
C_{max} = (2+1) +3+ (2+1+3+4) = 16

4.2 Computational results

To determine whether the previous heuristic on the $F2|NR : \alpha_{lt}|C_{max}$ problem would give good results, we compare it with the solution obtained by solving the integer programming model. There is no set of benchmark instances available to test our problem. Thus, we generate a set of instances which differ with respect to several important characteristics of two machine permutation flow shop subject to no renewable resources, including number of resources, the quantity of each resource required to perform jobs and the amount in which they are available. Processing time and resource requirement of jobs in the testing instances are generated randomly as follow:

- Processing time follows a uniform distribution in the range of $[1, 100]$ time units as usual in the scheduling literature, this distribution is even for the first and second machine.
- Resource requirement follows a uniform distribution in the range of $[1, 10]$.
- We perform tests when the number of no renewable resources R varies in the range $[1, 2]$.

Note that, the total amount of resource arrived must be greater than or equal to the total amount of resource requirement. Table 6, table 7 and table 8 present the comparison of the objective function values (makespan) and the computational times (in seconds) obtained by the heuristic and integer mathematical model for small, medium and large instances respectively. We estimate the gap according to the optimal solution reported by the mathematical model. The gap, in percent, is calculated as follow:

$$GAP = \frac{F[S(H)] - F[S^*]}{F[S^*]} \times 100$$

S^*: solution given by CPLEX, H corresponds to the value obtained by the heuristic. CPLEX v12.2 was used to solve the mathematical model, while all heuristics processes in this paper are programmed in JAVA language and run on i3 PC with 1, 5 GHz. We limit the simulation time for CPLEX to 1200 second.

Table 6. Computational experiment for small instances

Data		CPLEX		Heuristic		GAP
n	R	C_{max}	CPU_{time}	C_{max}	CPU_{time}	%
10	1	124	0,48	124	0,06	0
	2	125	0,03	127	0,09	1,6
30	1	417	0,44	417	0,08	0
	2	418	0,42	429	0,08	2,63

Table 7. Computational experiment for medium instances

Data		CPLEX		Heuristic		GAP
n	R	C_{max}	CPU_{time}	C_{max}	CPU_{time}	%
50	1	826	15	855	0,15	3,51
	2	910	11	1263	0,14	38,8
70	1	1008	5,80	1132	0,13	12,30
	2	1308	143	1778	0,12	35,93

Table 8. Computational experiment for large instances

Data		CPLEX		Heuristic		GAP
n	R	C_{max}	CPU_{time}	C_{max}	CPU_{time}	%
100	1	1632	5,16	1671	0,10	2,40
	2	1576	138	2009	0,10	27,50
300	1	–	≥ 1200	11276	0,459	–
	2	–	≥ 1200	13532	0,645	–

Results presented in table 6 show that for small sized problems $(10 - 30 jobs)$ with single no renewable resource requirement, the heuristic gives the same results as CPLEX. While when the number of resource increase the heuristic produces solutions that are very near from the optimal solution. For the medium instances, average gap of the solution found by the heuristic with respect to CPLEX are less than $12,3\,\%$ for single resource constraint. Whereas, for up to one resource the heuristic produces solutions that are less than $35,93\,\%$. From table 8, we observe that CPELX cannot solve instances over 300 jobs even for one or two resources. However, the heuristic is able to find solutions during a reasonable amount of computational time.

5. NEH ALGORITHM FOR $FM|NR : \alpha_{lt}|C_{max}$

In this section, we move from the special case two machines problem to the general case m machines problem under no renewable resource constraint with applying NEH heuristic. Nawaz et al. (1982) heuristic did not transform the original m machine problem into one artificial two machines problem like Campbell et al. (1970) heuristic, but instead build the final sequence in constructive way by adding a new job at each step and finding the best partial sequence. The authors have demonstrated that their heuristic outperforms those due to Campbell et al. (1970) and Dannenbring (1977). Taillard (1990), Ruiz and Serifo (2008), Semanco and Modrk (2012) and other researchers claim that NEH heuristic produce better solutions than other heuristics.

Algorithm 2 NEH Algorithm for $Fm|NR:\alpha_{lt}|C_{max}$

Inputs:
$U^{'}=\{1...n\}$set of jobs ordered by decreasing total processing time value.
$V_{n\times m}$ processing time matrix of n job, m machines.
seq : partial sequence.
$seq^{'}$: permutation sequence of a new placed job.
C_{pmax} : partial makespan $(completiontimeofanewplacedjob)$.
Initialize:
$C_{pmax}\leftarrow\infty$
for $i=1$ to n **do**
  for $j=1$ to m **do**
    Calculate $P_i^{'}=sumP_{ij}$;
  end for
end for
for $i=1$ to n **do**
  Sort jobs in decreasing order of their $P_i^{'}$
end for
for $i=1$ to n **do**
  Select job i from $U^{'}$
  for $k=1$ to i **do**
    Move job i to position k in a partial sequence seq, thus obtaining a new permutation $seq^{'}$;
    Verifies that, $\sum_{v=1}^{k} W_{vl}\leq\sum_{t=1}^{H} Z_{tl}.Y_{kt}$;
    Calculate the corresponding C_{max} value ;
    if $C_{max}<C_{pmax}$ **then**
      $seq\leftarrow$seq$^{'}$;
      $C_{pmax}\leftarrow C_{max}$;
    end if
  end for
  Place job i at position k in partial sequence
end for

In order to show if this heuristic yields a good results, we adopt it to m machines flow shop with respecting the availability of no renewable resources constraints

5.1 Computational results

In the computational experiment, we use the problem instances described earlier for m machines problems with m take the value of $5, 10$ and 20 for each job size. Note that we perform tests when the number of resources R varies in the range $[1-2]$. For all above problems sizes, the CPU times are noted and the gap, in percents, which refer to the difference between the NEH makespan and the optimal solution reported by CPLEX are calculated. Table 7 shows the results obtained by NEH heuristic and the mathematical model adapted to our problem. The first column contains instances, that is divided itself into three others, the first sub-column shows the number of jobs. The second, the number of machines and the last sub-column indicates the number of resources.

The second and the third columns show the makespan and the execution time obtained by respectively the CPLEX model and the NEH heuristic, the last column of the table calculate the gap. If we analyse the results which correspond to the first set job size (10 jobs), we can state that NEH heuristic gives the same results as CPLEX expectable for $10\times10\times2$ (10 jobs on 10 machines with two resource constraint) and $10\times20\times2$ instances where the results of NEH are near from the optimal solution with an average less than $1,55\,\%$. For $50\times5, 50\times10$ instances the results are suitable even for one or two resources with an average less than $7,32\,\%$ but when the numbers of machines increase (>10 machines) we observe that CPLEX was unable to solve instances. This later is due to the constraint of no renewable resources which increased the complexity of the problem. For the large size problem, the implemented NEH heuristic can efficiently solve the studied problem in considerably short time. However, CPLEX was unable to solve these instances. So, we can remark that, when the size of problem is small both NEH heuristic and CPLEX can solve it in an acceptable time, however, as the number of resources increases and the number of machines increases, the computation time of CPLEX increased exponentially.

Table 9. Comparison of the heuristic performance

Problem instance			Optimal solution		NEH		GAP
n	m	R	C_{max}	CPU_{time}	C$_{max}$	CPU_{time}	%
10	5	1	916	0,53	916	0,06	0,00
		2	513	0,69	513	0,04	0,00
	10	1	1148	1,62	1148	0,09	0,00
		2	1558	0,34	1577	0,22	1,22
	20	1	1065	2,70	1065	0,09	0,00
		2	1475	0,62	1498	0,04	1,55
50	5	1	5392	26,83	5413	0,09	0,34
		2	6738	69,44	7231	0,06	7,32
	10	1	6586	792,5	6667	0,19	1,22
		2	8176	825,1	8326	0,22	1,83
	20	1	–	≥ 1200	6832	0,23	–
		2	–	> 1200	7109	0,14	–
100	5	1	–	≥ 1200	6385	0,09	–
		2	–	≥ 1200	7142	0,13	–
	10	1	–	≥ 1200	7109	0,19	–
		2	–	≥ 1200	8654	0,17	–
	20	1	–	≥ 1200	7893	0,20	–
		2	–	≥ 1200	9432	0,19	–

6. CONCLUSION

In this paper, we investigated an extension of the classical flow shop scheduling problem to the case where jobs for their processing need additional non renewable resources, the goal is to minimize the makespan. After introducing some related works in the literature, we proposed a mathematical model. This model is tested on various instances with CPELX optimization software. Since Johnson algorithm works optimally for the classical two machines flow shop, we adopted it for the two machines flow shop subject to no renewable resources constraints on the second machine in order to illustrate the performance on our problem. Experimental results show that this heuristic is very efficient for one no renewable resource, on small sized problem, while when the number of resource increase the heuristic produces solutions that are very near from the optimal solution. Among the most effective heuristics, NEH was taken to improve their performance on the m machines flow shop under resources constraints, computational experiments have been performed to demonstrate the efficiency of this heuristic to our problem. Results show that we can obtain optimal solutions in some cases within a reasonable amount of computation time.

As future research, we try to study the complexity of the special cases of this problem. It could be interesting to adapt a metaheuristic for the problem and compare its performances with the NEH heuristic. The multi-objective case of the flow shop scheduling problems subject to resource constraint could be considered as an interesting topic for future research.

REFERENCES

Belkaid, F., Sari, Z., and Souier, M. (2013). A genetic algorithm for the parallel machine scheduling problem with consumable resources. *International Journal of Applied Metaheuristic Computing*, 4, 17–30.

Campbell, H.G., R.A.Dudek, and , M.S. (1970). A heuristic algorithm for the n job , m machine sequencing problem. *Management science*, 16, B630–B637.

Carlier, J. and Kan, A. (1982). Scheduling subject to nonrenewable-resource constraints. *Operations Research Letters*, 1, 2–5.

Carrera, S. (2010). *Planification et Ordonnancement de Plateformes Logistiques*. Ph.D. thesis, Ecole doctorale IAEM Lorraine.

Dannenbring, D.G. (1977). An evaluation of flow shop sequencing heuristics. *Management Science*, 23, 1174–1182.

Gafarov, E.R., Lazarev, A., and Werner, F. (2011). Single machine scheduling problems with financial resource constraints: Some complexity results and properties. *Mathematical Social Sciences*, 62, 7–13.

Grigoriev, A., Holthuijsen., M., and Klundert, J. (2005). Basic scheduling problems with raw material constraints. *Naval Research Logistics*, 52, 527–535.

Johnson, S.M. (1954). Optimal two- and three-stage production schedules with set-up times included. *Naval Research Logistics Quarterly*, 1, 61–68.

Laribi, I., Belkaid, F., Yalaoui, F., and Sari, Z. (2014). Investigation for solving hybrid flow shop problem with consumable resource. *International Conference On Metaheuristics And Nature Inspired Computing, META*.

Nawaz, M., Enscore, E., and Ham, I. (1982). A heuristic algorithm for the m-machine , n-job flow-shop sequencing problem. *The International Journal of Management Science*, 11, 91–95.

Ruiz, R. and Serifo, F. (2008). Modeling realistic hybrid flexible flowshop scheduling problems. *Computers Operations Research*, 35, 1151–1175.

Semanco, P. and Modrk, V. (2012). A comparison of constructive heuristics with the objective of minimizing makespan in the flow-shop scheduling problem. *Acta polytechnica Hangarica*, 9, 77–190.

Slowinski, R. (1984). Preemptive scheduling of independent jobs on parallel machines subject to financial constraints. *European Journal of Operational Research*, 15, 366–373.

Taillard, E. (1990). Theory and methodology some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 47, 65–74.

Toker, A., Kondakci, S., and Erkip, N. (1991). Scheduling under a non-renewable resource constraint. *Operational Research Society*, 42, 811–814.

Xie, J. (1997). Polynomial algorithms for single machine scheduling problems with financial constraints. *Operations Research Letters*, 21, 39–42.