Scheduling of blocking and no wait job shop problems in robotic cells

Saad Louaqad
Laboratoire des Technologies Innovantes
ENSAT - University ABDELMALEK ESSADI
90 000 Tanger, Morocco
Email: louaqad@gmail.com

Oulaid Kamach
Laboratoire des Technologies Innovantes
ENSA - University ABDELMALEK ESSADI
90 000 Tanger, Morocco
Email: kamach@ensat.ac.ma

***Abstract*—This paper focuses on the study of job-shop problems with transportation, blocking and no-wait (BNWT JSSP). This problem is an extension of the job shop problem. We take into account transport operations of jobs between different machines using a limited number of robots. We also have blocking and no wait constraints. It means that after a completion of an operation, a job waits on its machine, thus blocking it, until the next machine and a robot are available. In the other hand, the job starts its processing on the machine immediately and without any interruption after the robot has accomplished the transportation operation, thus no wait constraints. The aim of our problem is to find a feasible order of machine operations and transport operations. Additionally to the problem description, we will present a method based on graph theory which illustrates blocking situations. As the main contribution, we will formalize this problem with an integer linear programming models and propose an heuristic based on priority rules.**

I. Introduction

The JSSPs are well-known combinatorial optimization problems, which consist of a finite number of jobs and machines. Each job consists of a set of operations that has to be processed in a given order, on a set of known machines, and where each operation has a known processing time. No machine can process more than one operation at the same time. The objectives usually considered in JSSPs are the minimization of makespan.

Considerable research has been devoted to this problem in the literature. An overview of history and main techniques used along with their reported results on the available benchmarking problems for JSP can be found in [1].

Several problems in practice cannot be modeled as (classical) Job Shop, due to additional features like storage space and transportation constraints. The classical model supposes that storage space has an infinite capacity and transport operations from one machine to another take almost no time. In many practical cases, the number of buffers is limited and transports need to be considered for various reasons. Buffers may be expensive or inadequate for technological or process reasons. Transports may take a considerable amount of time, or only a limited number of mobile devices are available to execute transports operations.

The jobs shop problems with transportation, blocking and no wait constraints are met for example in factories with robotic cells. A robotic cell is a flow-shop or job-shop scheduling problem in which the jobs must be transported from machine to machine. Transports are performed by one or more robots. We have to assign the transport operations to the robots and to schedule both the machine and robot operations. Usually there is no buffer or buffers with only limited capacity to store the jobs outside the machines or the robots. This means that our scheduling must deal with blocking and no wait constraints.

Job shop models with blocking constraints (BJS) have been discussed by several authors. [2] give a survey on machine scheduling problems with blocking and no-wait constraints. [3] describes several applications of machine scheduling with blocking and no-wait in process and reviews the computational complexity of a variety of related problems. [4] and [5] study several types of job shop problems including the ideal (classical) job shop, the Blocking Job Shop (with and without "swaps") and the no-wait job shop, and formulate these problems by means of alternative graphs; [6] investigate scheduling of flexible manufacturing systems, study a multi-resource job shop problem with blocking constraints. [7] develop a genetic algorithm for solving no-wait and Blocking Job Shop problems. [8] and [9] introduce a generalized disjunctive graph framework for modeling various types of scheduling problems and develop a local search approach for the generalized Blocking Job Shop problem with application in automated warehouses. [10] propose a tabu search algorithm to solve the BJS problem for cyclical scheduling. [11] proposes a combination of a branch and bound algorithm with alternative graphs and develops two methods based on genetic algorithms to solve the BJS problem.

Most of the above authors emphasize that scheduling problems with blocking constraints (BJS) appear more difficult to solve than the classical job shop (JS). To illustrate the differences, two structural properties of the BJS shall be mentioned here, both being related to feasibility issues. First,

Xème Confrence Internationale : Conception et Production Intégrées, CPI 2015, 2-4 Décembre 2015, Tanger - Maroc.
Xth International Conference on Integrated Design and Production, CPI 2015, December 2-4, 2015, Tangier - Morocco.

in contrast to JS, a feasible partial schedule cannot always be extended to a feasible complete schedule. In fact, [4] and [5] established that deciding whether this is possible is NP complete (this was also shown independently by [8]). As a consequence, any heuristic that incrementally builds up a solution (e.g. based on priority rules) risks the chance of running into infeasibility. Second, it is not straightforward to construct feasible neighbor solutions in a local search approach as moves based on simple swaps of adjacent operations typically yield infeasible schedules (while in the JS case, it is well known that swapping critical adjacent operations leads to feasible neighbors).

Applications related to the BJS problem have been reported in the processing and logistics industries, such as scheduling for the manufacturing of concrete blocks by [12] , steelmaking by [13] , chemical batch production by [14] , container handling at a port by [15] and railway networks by [16].

Several researchers have devoted to study job shop scheduling problems with transportation constraints in various systems. However, the progress is limited as this kind of problem is difficult to solve even for simplified and small size cases.

[17] integrated transport constraints in the scheduling problem with one robot. [18] proposed a dynamic programming approach to construct optimal machine and vehicle schedules. [19] developed a mixed integer programming (MIP) formulation raising this constraint on the vehicles. [20] and [21] elaborated a genetic algorithm. [22] and [23] proposed, respectively, neural networks and tabu search approaches. [24] described a hybrid method composed of a genetic algorithm for the scheduling of machines and a heuristic for the scheduling of vehicles. [25] and [26] considered a job shop problem with several robots, with fixed operation times and fixed assignment of machine for each job's operation. [27] studied a two machine flow shop scheduling problem with intermediate transportation with a single transporter. [28] considered a cyclic hoist scheduling problem with a single hoist, but without assignment problem. [29] used a mixed integer linear program (MILP) to find optimal solutions for the FMS Scheduling Problem with one vehicle. [30] proposed a polynomial algorithm for finding the optimal cycle in a robotic cell with production of identical parts. [31] studied coupled task problem and one-machine robotic cell problems. It reported new algorithmic procedure for this problem with or without tolerances on the distance. [32] applied a decomposition method where the master problem (scheduling) is modeled with constraint programming and the subproblem (conflict free routing) with mixed integer programming.

To the best of our Knowledge there is no research that addressed the problem of job shop scheduling that take into account transfer time between machine performed by a limited number of robot and the non existence of buffers between machines that lead to blocking and no wait constraints.

Two common approaches to tackle the scheduling problems are the utilization of mathematical programming and heuristic approaches [33]. By Mathematical programming methods we formulates the scheduling problems as a mixed integer linear programming (MILP) problem and then tries to solve the formulated problem by a general purpose mixed integer linear programming solver. This methods are usually suitable for small instances. However and due to the great advance recently obtained in capacity of computers and creation of fast optimization software, presentation of MILP models is becoming more and more interesting among the researchers.

In this paper, we propose two methods to address the BNWT JSSP. We develop first an integer Linear programming model based on the model of [34] and secondly we construct an heuristic based on priority rules.

This paper is structured as follows. In the next section, we will present definitions and notations associated to our model. After that, in Section 3, we develop our integer programming formulation for the BNWT JSSP. In Section 4, we illustrate the theoretical tools and properties needed for developing our Heuristic. After that, we present the algorithm of our proposed heuristic. Finally, Section 5 gives some interesting conclusions and future studies.

II. Problem definition

We consider a job shop problem with several transport robots and no buffers. In this problem, a set of n jobs $\{J_1, J_2, ..., J_n\}$ are processed on a set of m machines $\{M_1, M_2, ..., M_m\}$ and transported by a set of k robots $\{r_1, r_2, ..., r_k\}$. Transportation times are robot-independent. Every job J_i require an operation order, $J_i = \{O_{i1}O_{i2}, .., O_{in_i}\}$, that must be executed according to its manufacture process. Operation O_{ij} of the job J_i requires the exclusive use of $M_l (l \in \{1, 2, ..., m\})$ for an uninterrupted duration p_{ij}, its processing time; the preemption is not allowed; each machine can process only one job at a time; the machine which execute the operation O_{ij} is denoted as M_{ij}.

In addition, we consider transportation operations between two machines. Consider two successive operations of the same job O_{ij} and O_{ij+1} to execute in two machines M_{ij} and M_{ij+1}. T_{ij} and t_{ij} are respectively used to denote transport operation and transfer time of job J_i from machine M_{ij} to machine M_{ij+1}. Each robot can handle at most one job at one time. Loaded transfer times does not depend on the job transported, but only on the travel routes and the robot which perform the transportation operation. This times are given by $C^r_{p,l}$ where r represents the robot and (p, l) represents the route between machine M_p and M_l. It is assumed that the triangle inequality is satisfied:

$\forall p, l \in \{1, 2, ..., m\}$ machine indexes.
$\forall r \in \{r_1, r_2, ..., r_k\}$

$$C^r_{p,h} + C^r_{h,l} \geq C^r_{p,l} \tag{1}$$

(1) means that the direct way between two machines is at least as short as the detour through a third machine. Otherwise, the robot always takes the shorter way through the third machine.

Note that a sequence of loaded transport operations indirectly induces necessary empty moves. Empty transfer time from machine M_p to M_l is denoted $V^r_{p,l}$. It is assumed that:

$\forall r, r' \in \{r_1, r_2, ..., r_k\}$
$\forall p, l \in \{1, 2, ..., m\}$ machine indexes

$$\left\{\begin{array}{l} V^r_{p,p} = 0 \\ V^r_{p,h} + V^r_{h,l} \geq V^r_{p,l} \\ V^r_{p,l} \leq C^{r'}_{p,l} \end{array}\right. \quad (2)$$

The first assumption means that no empty transfer time is considered if a robot waits at the same machine the next transportation operation. The second one is the triangular inequality for empty moves. The third one means that empty transfers between two machines by a robot r take less time than loaded transfers between this two machines by another robot r'. (It is also valid if $r = r'$). In the other hand, we consider the blocking constraint because there is no machine buffer. This means that after finishing its processing on the machine, a job has to stay there until it is unloaded by the robot. During this stay, the machine is blocked and not available for processing any other job. We also consider the no wait constraint that means if the robot transporting the job J_i reaches machine M_{ij}, the operation O_{ij} must start immediately without any interruption.

The scheduling problem objectives are:

– To determine the starting time for each machine operation O_{ij}

– To assign a handling robot to each transport operation T_{ij} and to determine its starting time

– To minimize the Makespan denoted $C_{max} = max(Ci)$ where Ci denotes the completion time of the last operation of job J_i.

All data are assumed to be non-negative integers.

III. Integer Programming Model

This section presents the ILP models to formulate BNWT JSSP. Our following formulation is based on the model of [34]. We used their ideas to model the Blocking Job shop with no wait and transportation constraints.

To model the blocking job shop with no wait and transportation constraints, the following notations are used.

List of parameters :

– p_{ij}: the processing time of operation O_{ij}
– C^r_{pl}: on load transfer time of robot r between machine p and machine l
– $V^r_{p,l}$: empty transfer time of robot r between machine p and machine l
– H a large number

List of variables:

– d_{ij}: the start time of machine operation O_{ij}
– f_{ij}: the completion time of machine operation O_{ij}. (Time when operation O_{ij} leave machine M_{ij}
– d'_{ij}: the start time of transport operation T_{ij}
– f'_{ij}:the completion time of transport operation T_{ij}

List of decision variables:

– $\alpha_{ij;lq}$: Binary variable that takes value 1 if O_{ij} is processed before O_{lq} and 0 otherwise.
– $\beta_{ij;r_s}$: Binary variable that takes value 1 if T_{ij} require processing on robot r_s, and 0 otherwise.
– $\delta_{ij;lq}$: Binary variable that takes value 1 if T_{ij} is processed before T_{lq}, and 0 otherwise.
– $\lambda_{ij;lq}$: Binary variable that takes value 1 if T_{ij} and T_{lq} are processed by the same robot, and 0 otherwise.

The problem is formulated as follows :

Minimize $Cmax$
Subject to:

– Finish time of machine operations:
$For\ i \in [1, n]; j \in [1, m]$

$$f_{ij} \geq d_{ij} + p_{ij} \quad (3)$$

– Precedence Constraints between transport operations and machine operations:
$For\ i \in [1, n]; j \in [1, m-1]$

$$\left\{\begin{array}{l} f'_{ij} = d_{ij+1} \\ d_{ij+1} = d'_{ij} + \sum_{s=1}^{k} \beta_{ij,r_s} C^{r_s}_{M_{ij} M_{ij+1}} \end{array}\right. \quad (4)$$

– Precedence constraints between machine operations and transport operations:
$For\ i \in [1, n]; j \in [1, m-1]$

$$d'_{ij} = f_{ij} \quad (5)$$

– Disjunctive constraints between machine operations:
$For\ i, l \in [1, n]; j, q \in [1, m]\ / M_{ij} = M_{lq}$

$$\left\{\begin{array}{l} d_{ij} \geq f_{lq} - H * \alpha_{ij;lq} \\ d_{lq} \geq f_{ij} - H * (1 - \alpha_{ij;lq}) \\ \alpha_{ij;lq} + \alpha_{lq;ij} = 1 \end{array}\right. \quad (6)$$

– Robot assignment constraints:
$For\ i \in [1, n]; j \in [1, m-1], s \in [1, k]$

$$\sum_{s=1}^{k} \beta_{ij,r_s} = 1 \quad (7)$$

– Disjunctive constraints between robot:
  For $i, l \in [1, n]; j, q \in [1, m-1]; s \in [1, k]$

(3) Ensures that each operation is processed at least for its process duration
(4) ensures that each machine operation starts immediately after the finish of the transport operation that precede it.
(5) As there is no buffer in machines, transport operation starts immediately when the job leaves a machine .
(6) ensures that each machine process one operation at a time.
(7) ensures that each transport operation is performed by only one Robot.
(8) ensures that each robot process one operation at a time. no two transport operations are performed by the same robot at any time

IV. Heuristic

A. *Blocking Situation*

In this section, we will look for situations that could lead to blocking states in order to avoid them during the construction of our algorithm. For this purpose, we will identify blocking situations by using a graph $G_s = (M, J)$ that we will call the graph of last scheduled operations. This graph is defined below: Consider the graph of last scheduled operations $G_s = (M, J)$. A set of vertices M represents machines; A set of arcs J represents the last scheduled operations O_{ij} of job J_i. The starting point of the last scheduled operation O_{ij} of job J_i is the machine M_{ij} and its end point is the machine M_{ij+1}.
Consider for example a problem with 5 machines $\{M_1, M_2, ..., M_5\}$ and 4 jobs $\{J_1, J_2, J_3, J_4\}$. The last scheduled operations are: $J_1 : M_1 -- > M_2$; $J_2 : M_2 -- > M_3$; $J_3 : M_3 -- > M_1$, $J_4 : M_4 -- > M_5$. The associated graph G_s is modeled as follows:

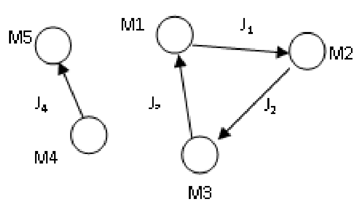

Fig. 1. Associated graph of last scheduled operations

Following the topology of $G_s = (M, J)$, the graph contains a cycle of length $p = 3$ (p equals the number of jobs that forms the cycle. Under this cycle, the necessary condition of blocking (C1) is satisfied because job J_1 that is processed on the machine M_1 cannot pass to machine M_2 occupied by the job J_2, as well as for job J_2 that is processed on machine M_2 cannot pass to machine M_3 occupied by job J_3 as well as for job J_3 that is processed on machine M_3 cannot pass to machine M_1 occupied by the job J_1.

Blocking condition (C1) can be formulated as follows: "Systems may confront a blocking situation if the graph of the last scheduled operations $G_s = (M, J)$ contains a cycle of length $p \geq 2$." Thereafter, we will check if the condition (C1) is a sufficient condition to blocking situation in BNWT JSSP.

<u>1st Case : $p = 2$</u>
Consider the graph G_s with a cycle of length $p = 2$.

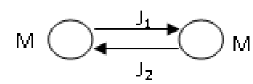

Fig. 2. Graph with a cycle of length $p = 2$

When the processing of job J_1 on machine M_1 has finished, job J_1 remains blocked on machine M_1 until its transportation operation T_1 from M_1 to M_2 start. We assume that a robot r_1 is available and will be assigned to transportation operation T_1. This operation can start and liberate the machine M_1. No wait condition imply that the processing of job J_1 on machine M_2 must start immediately after the termination of transport operation T_1 otherwise J_1 cannot be transported by r_1 and then machine M_1 remains blocked. M_2 is blocked by J_2 and must be liberated before the termination of transport operation T_1. Therefore, we must have a second robot to execute transport operation T_2 of J_2 before that J_1 arrives at M_2.

Proposition 1: If the graph of last scheduled operations $G_s = (M, J)$ contains a cycle of length $p = 2$, the system is eternally blocked if it has one handling robot ($k = 1$) and partially blocked if $k \geq 2$.

<u>Second Case: $p \geq 3$</u>
Consider the graph G_s which a cycle of length $p \geq 3$

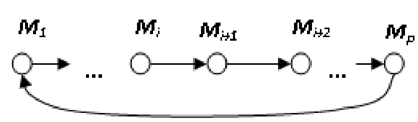

Fig. 3. graph with a cycle of length $p \geq 3$

We assume that a robot r_1 is available. Robot r_1 will be assigned to operation T_i. Before that operation T_i has finished, operation T_{i+1} must start. For this reason, we have a second robot r_2 which will be assigned to operation T_{i+1}.

Assumption 1:
Assume that operation T_{i+1} starts at the time that T_i finishes: $d'_{i+1} = f'_i$. At the time f'_i, r_1 arrives in front of machine M_{i+1} and unload job J_1. Then it can drive empty to machine

$$
\begin{cases}
d'_{ij} \geq d'_{lq} + \sum_{s=1}^{k} \beta_{lq,r_s} (C^{r_s}_{M_{lq}M_{lq+1}} + V^{r_s}_{M_{lq+1};M_{ij}}) + (\lambda_{ij;lq} - 1)H - \delta_{ij;lq}H \\
d'_{lq} \geq d'_{ij} + \sum_{s=1}^{k} \beta_{ij,r_s} (C^{r_s}_{M_{ij}M_{ij+1}} + V^{r_s}_{M_{ij+1};M_{lq}}) + (\lambda_{ij;lq} - 1)H + (\delta_{ij;lq} - 1)H \\
\delta_{ij;lq} + \delta_{lq;ij} = 1
\end{cases} \tag{8}
$$

M_{i+2}. r_1 reaches M_{i+2} at a time equal to $f'_i + V^{r_1}_{M_{i+1}M_{i+2}}$. The robot r_2 which is performing transport operation T_{i+1} reaches machine M_{i+2} at a time equal to $d'_{i+1} + C^{r_2}_{M_{i+1}M_{i+2}}$. According to (2) and (Assumption 1) we have:
$f'_i + V^{r_1}_{M_{i+1}M_{i+2}} \geq d'_{i+1} + C^{r_2}_{M_{i+1}M_{i+2}}$. This means that the robot r_1 reaches machine M_{i+2} and liberate this machine before that the robot r_2 which is executing T_{i+1} reaches this machine and so on until the blocking cycle is totally liberated.

Proposition 2:
If the graph of last scheduled operations G_s contains a cycle of length $p \geq 3$, the system is eternally blocked if it has a single handling robot ($k = 1$) and partially blocked if $k \geq 2$.

B. The proposed Heuristic

In this subsection we propose a heuristic dedicated to the BNWT JSSP. During construction of the algorithm, we complete iteratively a partial schedule S. U denotes the set of non-scheduled operations. At each iteration of the heuristic, two operations are selected, namely a machine operation and a transportation operation. When a machine operation is chosen on the basis of selection rules, the transportation operation which precedes this machine operation is automatically selected and a robot is assigned according to another priority rules.
The Iteration starts with the construction of the set E of eligible machine operations. An operation is eligible for partial schedule S if it is a non scheduled operation that can start without violating any constraints.

- Rule 1: Eligible operations $S(M_p)$ that need to be processed on machine M_p cannot be scheduled as long as machine M_p is occupied by another job. These operations will be eliminated from the set E.
- Rule 2 : Eligible operations that could lead to eternal blocking situations or even a cycle in the graph $G_s(M, J)$ with $p \geq 3$ will also be eliminated from the set E.

After the determination of eligible operations, it remains to appoint the machine operation to be scheduled. For this purpose, we associate each operation $O_{ij} \subset E$ to a pair (M_p, g) where $M_p = M_{ij}$ and g is the total time of operations $S(M_p) \subset U$, g can be seen as the weight of M_p on the set of non scheduled operations U.

Selection rules of machine operations: Machine operation to be scheduled is the one associated to a pair (M_p, g) with the largest g $(g = gmax)$

- Rule 3: If we have on the set E two or more operations with the same pair $(M_p, gmax)$, we choose the operation that has the longest queue.
- Rule 4: If we have on the set E two operations with the same $gmax$ but with different machines, we schedule that terminate first.

Selection rules of robots:

- Rule 5: To select the robot that will perform the transportation operation, we opted to choose the robot which provides the minimal completion time of the transportation operation, which involves exploring all robots for each assignment.
- Rule 6: In the case of a cycle of length $p = 2$, we choose the robot that has the earliest availability time. This time correspond to the empty robot arrival time at the departure machine for the loaded move.

V. Conclusion

In this work, we described the job shop with transportation and subject to no wait and blocking constraints. We extended an existing model from the literature and showed how an integer programming formulation can be applied to solve the problem. We illustrated the blocking situations that the system could confront and finally proposed an algorithm for the construction of a heuristic. Experimental results for the proposed algorithm are not yet provided. On future researches, we will focus on a hybrid methods based on this article proposed heuristic. Experimental results will be established to show the efficiency of our developed methods.

References

[1] A.S. Jain, S. Meeran, Deterministic job-shop scheduling: Past present and future, European Journal of Operational Research 113 (1999) 390-434.

[2] N.G. Hall, C. Sriskandarajah, A survey of machine scheduling problems with blocking and no-wait in process, Operations Research 44 (3) (1996) 510-525.

[3] O. Candar, Machine scheduling problems with blocking and no-wait in process, Working Paper [April-99], Department of Industrial Engineering, Bilkent University, Ankara, Turkey, 1999.

[4] Mascis, D. Pacciarelli, Machine scheduling via alternative graphs, Research Report, RT-DIA-46-2000, Italy, 2000.

[5] A. Mascis, D. Pacciarelli, Job-shop scheduling with blocking and no-wait constraints, European Journal of Operational Research 143 (2002) 498-517.

[6] Y. Mati, N. Rezg, X. Xie, Geometric approach and taboo search for scheduling flexible manufacturing systems, IEEE Transactions on Robotics and Automation 17 (6) (2001) 805-818.

[7] C.A. Brizuela, Y. Zhao, N. Sannomiya, No-wait and blocking job-shops: Challenging problems for GA's, IEEE 0-7803-77-2/ 01 (2001) 2349-2354.

[8] A. Klinkert, Optimization in design and control of automated high-density warehouses, Doctoral Thesis No. 1353, University of Fribourg, Switzerland, 2001.

[9] H. Groflin, A. Klinkert, Local search in job shop scheduling with synchronization and blocking constraints, Internal working paper [04-06], Department of Informatics, University of Fribourg, Switzerland, 2004.

[10] P. Brucker, T. Kampmeyer, Cyclic job shop scheduling problems with blocking,Ann. Oper. Res. 159 (2008) 161181.
[11] A. AitZai, B. Benmedjdoub, M. Boudhar, A branch and bound and parallel genetic algorithm for the job shop scheduling problem with blocking, Int. J. Oper. Res.14 (3) (2012) 343365.
[12] J. Grabowski, J. Pempera, Sequencing of jobs in some production system, European Journal of Operation Research 125 (2000) 535550.
[13] D. Pacciarelli, M. Pranzo, Production scheduling in a steelmaking-continuous casting plant, Computer and Chemical Engineering 28 (2004) 28232835.
[14] J. Romero, L. Puigjaner, T. Holczinger, F. Friedler, Scheduling intermediate storage multipurpose batch plants using the S-graph, AIChE J. 50 (2004) 403417.
[15] L. Chen, N. Bostel, P. Dejax, J. Cai, L. Xi, A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal, European Journal of Operation Research 18 (2007) 4058.
[16] A. DAriano, D. Pacciarelli, M. Pranzo, A branch and bound algorithm for scheduling trains in a railway network, European Journal of Operation Research 183 (2007) 643657.
[17] J. Hurink ,S. Knust, Tabu search algorithms for job-shop problems with a single transport robot, European Journal of Operational Research 2005;162(1):99111.
[18] J. Blazewicz,H. Eiselt ,G. Finke,G. Laporte,J. Weglarz, Scheduling tasks and vehicles in a flexible manufacturing system, International Journal of Flexible Manufacturing Systems 1991;4(1):516.
[19] U. Bilge,G. Ulusoy, A time window approach to simultaneous scheduling of machines and material handling system in an FMS. Operations Research 1995;43(6):105870.
[20] G. Ulusoy, F. Sivrikaya-erifolu, U. Bilge, A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles, Computers & Operations Research 1997;24(4):33551.
[21] G. Alvarenga,G. Mateus, G. De Tomi, A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows, Computers & Operations Research 2007;34(6):156184.
[22] M. Soylu,N. O zdemirel, S. Kayaligil, A self-organizing neural network approach for the single AGV routing problem, European Journal of Operational Research 2000;121(1):12437.
[23] J. Hurink,S. Knust, A tabu search algorithm for scheduling a single robot in a job shop environment, Discrete Applied Mathematics 2002;119(12):181203.
[24] T. Abdelmaguid,A. Nassef, B. Kamal, M. Hassan, A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles, International Journal of Production Research 2004;42(2):26781.
[25] P. Lacomme,M. Larabi, N. Tchernev, A disjunctive graph for the job shop with several robots, In: MISTA conference; 2007. p. 28592.
[26] L. Deroussi, M. Gourgand, N. Tchernev, A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles, International Journal of Production Research 2008;46(8):214364.
[27] H. Gong, L. Tang, Two-machine flow shop scheduling with intermediate transportation under job physical space consideration, Computers & Operations Research 2011;8(9):126774.
[28] M. Mateo, R. Companys, J. Bautista, Resolution of graphs with bounded cycle time for the cyclic hoist scheduling problem, In: 8th International Workshop on Project Management and Scheduling. Cite seer; 2002. p. 25760.
[29] A. Caumond, P. Lacomme, A. Moukrim, N. Tchernev, An MILP for scheduling problems in an FMS with one vehicle, European Journal of Operational Research 2009;199(3):70622.
[30] Y. Crama, J. Van de Klundert, Cyclic scheduling of identical parts in a robotic cell. Operations Research 1997;45(6):95265.
[31] N. Brauner, G. Finke, V. Lehoux-Lebacque, C. Potts, J. Whitehead, Scheduling of coupled tasks and one machine no wait robotic cells, Computers & Operations Research 2009;36(2):3017.
[32] AI. Corra, A. Langevin, L-M. Rousseau, Scheduling and routing of automated guided vehicles: a hybrid approach. Computers & Operations Research 2007; 34(6):1688707, [part special Issue: Odysseus 2003 second international workshop on freight transportation logistics. doi:10.1016/j.cor.2005.07.004].
[33] E. Stafford, F. Tseng,J. Gupta, Comparative evaluation of MILP flow shop models, Journal of Operations Research Society 2005; 56;88101.
[34] A. Manne, On the job shop scheduling problem. Operations Research 1960;8,21922