MATS–JSTL: A Multi-Agent Model Based on Tabu Search for Job Shop Problem with Time Lags

Madiha Harrabi[2(✉)] and Olfa Belkahla Driss[1,2]

[1] Stratégies d'Optimisation et Informatique IntelligentE, High Institute of Management, University of Tunis, 41, Street of Liberty Bouchoucha-City CP-2000-Bardo, Tunis, Tunisia
olfa.belkahla@isg.rnu.tn

[2] Higher Business School of Tunis, University of Manouba, Manouba, Tunisia
madiha.harrabi@gmail.com

Abstract. The Job Shop problem with Time Lags (JSTL) is an important extension of the classical job shop scheduling problem, in that additional constraints of minimum and maximum time lags existing between two successive operations of the same job are added. The objective of this work is to present a distributed approach based on cooperative behaviour and a tabu search metaheuristic to finding the scheduling giving a minimum makespan. The proposed model is composed of two classes of agents: a Supervisor Agent, responsible for generating the initial solution and containing the Tabu Search core, and Resource_Scheduler Agents, which are responsible for moving several operations and satisfaction of some constraints. Good performances of our model are shown through experimental comparisons on benchmarks of the literature.

Keywords: Job shop scheduling · Time lags · Tabu search · Multi-agent system

1 Introduction

Different works on scheduling were concerned with the analysis of single-machine systems, parallel-machines systems, and shop problems. Among all these problems, we focus in this paper to optimize the total duration of the scheduling often called makespan for the Job Shop problem with time lags. The job shop problem with time lags is defined by a set of n jobs which have to be sequenced on a set of m machines. The constraints of time lags is a generalization of precedence constraints which concern the minimum and maximum waiting time existing between two successive operations of the same job. In this paper, we propose a Multi-Agent model based on Tabu Search for the Job Shop problem with Time Lags (MATS–JSTL) in order to minimize the makespan. This paper is organized as follows. In Section 2, we present related works. In Section 3, we present the problem formulation. Then, we describe the tabu search method on which our model is based. In the following sections, the Multi-Agent system, and the adaptation of the tabu search parameters are provided. Then, we discuss experimental results of our model. Finally, section 8 contains our conclusions and remarks with some future directions.

M. Núñez et al. (Eds.): ICCCI 2015, Part I, LNAI 9329, pp. 39–46, 2015.
DOI: 10.1007/978-3-319-24069-5_4

2 Related Works

2.1 Previous Published Works on Scheduling Problems with Time-Lags

The time lags constraints were introduced for the first time by [14] in a problem with parallel machines. In the literature, there are two kinds of time lags constraints: we found time lags between successive operations of the same job and generic time lags between whatever pairs of operations introduced by [13] in the Job Shop problem. The first kind of time lags was introduced by [15] in the single-machine problems by including minimum and maximum times lags. [10] introduced both minimum and maximum time lags in the single machine problem. [6] proposed different methods for solving Flow Shop problem with minimum and maximum time lags. In [9] an exact procedure for a general RCPSP with minimum and maximum time lags is proposed.

2.2 Previous Published Works on Job Shop Problem with Time-Lags

The job shop problem with time lags constraints is very little studied in the literature and the number of research and solution methods is limited. [1] proposed a branch and bound method. In [3] an adaptation of the tabu search method was presented. [5] investigated a memetic algorithm. In [4], a method of priority rules had been proposed. A construction heuristics had been proposed by [6]. In [11] a CDS (Climbing Discrepancy Search) was proposed. [2] used a cooperative Multi-Agent system, composed of Interface Agent, Job Agent, and Resource Agent.

3 Problem Formulation

The Job Shop Scheduling problem with Time Lags is formulated as follows.

M	set of machines
O	set of operations to schedule
Z	completion time of the last operation
E_k	set of operations processed on machine k
t_i	starting time of operation i
p_i	processing time of operation i
s_i	next operation of operation i depending on job
$\mathrm{TL}_i^{\min}$	minimal time-lags between i and si
$\mathrm{TL}_i^{\max}$	maximal time-lags between i and si

$$a_{ij} = \begin{cases} 1 & \text{if operation } j \text{ is processed before } I \\ 0 & \text{otherwise} \end{cases}$$

H a positive large number

Minimize Z,
Subject to

$(H + p_j)\, a_{ij} + (t_j - t_i) \geq p_i \quad \forall (i, j) \in E_k, \forall k \in M$ (c1)

$(H + p_j)(1 - a_{ij}) + (t_i - t_j) \geq p_j \quad \forall (i, j) \in E_k, \forall k \in M$ (c2)

$t_i + p_i + TL_i^{min} \leq t_{si} \quad \forall i \in O$ (c3)

$t_{si} \leq t_i + p_i + TL_i^{max} \quad \forall i \in O$ (c4)

$Z \geq t_i + p_i \quad \forall i \in O$ (c5)

$t_i \geq 0 \quad \forall i \in O$ (c6)

$a_{ij} \in \{0; 1\} \quad \forall (i, j) \in E_k, \forall k \in M$ (c7)

Constraints (c1) and (c2) represent the machine disjunctions, which mean that two operations i and j use the same resource. These constraints express either that operation i precedes the operation j or the reverse. Constraint (c3) relates to the minimum time lags, and the constraint (c4) represents the maximum time lags. Constraint (c5) determines the completion time of the last operation, which represents the optimization criteria (makespan or Cmax). Constraints (c6) and (c7) give the domain of definition of variables a_{ij} and t_i.

4 Tabu Search

The Tabu Search (TS) algorithm was first proposed by [8] which has been successfully used for solving a large number of combinatorial optimization problems specially scheduling problems. Tabu search uses a local neighborhood search procedure to iteratively move from one potential solution S to an improved solution S' in the neighborhood of S, until some stopping criterion has been satisfied. A particularity of TS is that it explicitly employs the history of the search, and storing the selected move in a structure named tabu list for a certain number of iterations called Tabu list size.

5 The Multi-Agent Model

5.1 Global Dynamic

A multi-agent system is a computerized system composed of multiple interacting intelligent agents who cooperate, communicate, and coordinate with each other to reach common objectives. We propose a multi-agent model based on tabu search for job shop problem with time lags. It is composed of two classes of agents: Supervisor Agent and a set of m Resource_Scheduler Agents with m refers to the number of machines, see figure 1.

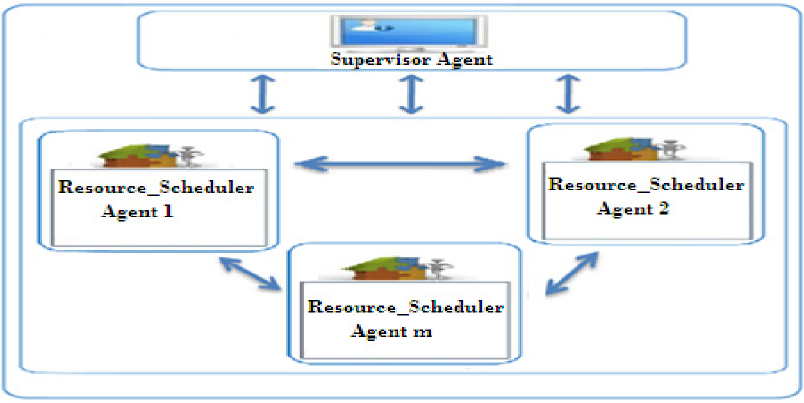

Fig. 1. The Multi-Agent model based on Tabu Search for Job Shop problem with Time Lags

5.2 Supervisor Agent Description

This agent contains the core of the tabu search algorithm. It aims to launch the program, build an initial solution, create Resource_Scheduler agents, and assign for each agent its workspace defined by all operations belonging to the same resource. It subsequently provides the necessary information for each Resource_Scheduler Agent. This agent is unsatisfied as long as the maximal number of iteration is not reached, and it aims to detect that the problem has been resolved. Static knowledge of this agent is composed of the initial solution S0 from which begins the optimization process, the used stopping criterion, the size of the Tabu List, the used diversification criterion. Its dynamic knowledge consists of: the current solution and its makespan, neighbor solutions and their makespans, the best found schedule and its makespan, the tabu list elements, the performed number of iterations, the number of iterations after the last improvement. The Supervisor agent is satisfied when the stopping criterion is reached; in this case it provides the found-solution to the user.

5.3 Resource_Scheduler Agent Description

The Resource_Scheduler Agents are created by the Supervisor Agent according to resource number. The decisions of scheduling are progressively negotiated between the different Resource_Scheduler Agents involved in the system to satisfy different constraints of precedence, disjunctive and time lags. Static knowledge of this agent is composed of the execution time of each job on all machines and time lags between operations. Dynamic knowledge as current solution and the job to move sent by the Supervisor Agent. Communication of Resource_Scheduler Agents handles the sending and receiving messages between each other containing the new locations of operations. This agent is satisfied when the constraints of succession and disjunction on his workspace operations are met, otherwise it is dissatisfied and he tries to solve the problem. The algorithm describes the moving operation procedure. cf. algo1.

Moving operation algorithm

```
: Start
: move_operation ← true;
: msg.receiver("job""op");
: compute_inactivity_intervals;
: insert_op ← satisfaction_verification(c3, c4 , c5 );
: if constraints_satisfaction ← false;
: search new location
: else
: insert_op ← true;
: msg.setContent("job""op", start_date, end_date);
: Update inactivity_intervals ;
: End
```

Algo. 1. Moving operation algorithm

6 Adaptation of Tabu Search Elements

6.1 Initial Solution

The initial solution is the starting embedding used for the algorithm to begin the search of better configurations in the search space. In this implementation the starting solution is randomly generated. Then it proceeds iteratively to visit a series of locally best configurations following a neighborhood function.

6.2 Neighborhood Function

In our approach, we choose to use a neighborhood structure based on insertion move. In fact, the move strategy is to move various operations in the inactivity intervals previously calculated by the Resource_Scheduler Agent. Respecting the precedence constraints and time lags constraints, the Resource_Scheduler Agents work together to find the new location for each operation that meets all these requirements. After moving all operations, each Resource_Scheduler Agent sends a message to the Supervisor Agent containing new locations of operations moved, so the Supervisor Agent generates the neighbor solution and launches the evaluation phase.

6.3 Evaluation of the Current Solution

The best non-tabu neighbor among the neighborhoods of the current solution will be selected for the next iteration, for this it would be necessary to evaluate all neighbors. The Supervisor Agent, after receiving all new locations of the operations of different jobs, from Resource_Scheduler Agents, it builds neighbor solution and calculates its makespan, chooses the best non-tabu neighbor and start the next iteration.

6.4 Tabu-List

A fundamental element of tabu search is the use of a flexible memory, short-term, which keeps track of some recent past movements. The size of the tabu list used is a static size “LT”. If the size of the tabu list exceeds the maximum allowed size, we remove from this list the oldest item (FIFO strategy: First In First Out: The first arrival element is the first to be deleted).

6.5 Diversification

When the number of iterations continuously increases without any improvement in the current solution, this means that the best solution found has not been replaced by one of these neighbors for some time, which is a sign that tabu search was probably trapped in a local optimum. In this situation a new schedule is built by the Supervisor Agent using a new insertion order of jobs in order to explore a new region of the search space. And it restarts again the resolution process. After that, the number of iterations diversification, i.e. the number of iterations after the last improvement, is reset and the research process continuous by considering the solution obtained by diversification phase as a new current solution until reaching the stopping criterion.

6.6 Stopping Criterion

In our model, we adopted as a stopping criterion, the maximum number of iterations. The tabu search process attempts to improve the solution after a maximum number of iterations. Then, the best found scheduling during the search and its makespan are returned.

7 Experimentation Results

Several experiments were conducted on a set of benchmarks for Job Shop problems with minimum and maximum time lags constraints existing in the literature, to test the effectiveness and the performance of our model using the JADE platform. These instances correspond to classical problems of Job Shop, and were constructed by [3]. Instances la01 to la05 (10 jobs, 5 machines) of [12] and instances ft06 (6 jobs, 6 machines) of [7] modified to introduce the constraints of time lags. The number of iterations is 1000.

Finally we choose to compare our distributed model MATS–JSTL with two centralized approaches, the approach of [4] which is based on a operation insertion heuristic OI, and the approach of [1] that is based on a job insertion heuristic JI. Then, we choose to compare our approach with the only distributed approach MAJSPTL of [2] which is a multi-agent approach. The results of comparison between several approaches are given in table 1 showing results of different instances. The obtained results show that our model MATS–JSTL provides good results for most existing benchmarks in the literature in terms of makespan. For all used instances, we find a

Table 1. Results for JI Heuristic/OI Heuristic/MAJSPTL/MATS–JSTL

	Makespan				CPU time (sec)
Instances	JI Heuristic	OI Heuristic	MAJSPTL	MATS–JSTL	MATS–JSTL
ft06_0_0	96	83	80	**80***	7
ft06_0_0.5	72	109	96	**69***	4
ft06_0_1	72	58	67	64	5
ft06_0_2	70	55	72	60	2
la01_0_0	1258	1504	1341	**1020***	1117
la01_0_0.5	1063	1474	1231	**980***	94
la01_0_1	928	1114	879	907	62
la01_0_2	967	948	980	**894***	37
la02_0_0	1082	1416	1264	1085	213
la02_0_0.5	1011	1207	1099	**973***	154
la02_0_1	935	1136	923	**694***	121
la02_0_2	928	895	978	**874***	473
la03_0_0	1081	1192	1023	1178	198
la03_0_0.5	930	1085	898	946	246
la03_0_1	886	931	904	894	168
la03_0_2	808	787	854	**761***	34
la04_0_0	1207	1346	1387	1393	218
la04_0_0.5	870	1156	921	1234	21
la04_0_1	1010	857	1053	915	78
la04_0_2	892	838	870	**827***	31
la05_0_0	1080	1224	1219	1168	178
la05_0_0.5	935	1208	1094	**962***	87
la05_0_1	814	964	952	**803***	61
la05_0_2	749	683	751	763	44

makespan value better than Artigues's method [1] (JI heuristic) in 62% of instances, better than Caumand's method [4] (OI heuristic) in 75% of instances, and better than Ben Yahia and Belkahla's method [2] (MAJSPTL) in 75%. CPU times (sec) of the solutions found by our model given in table 1, are reasonable.

8 Conclusion and Future Researches

In this paper, we have proposed a multi-agent model based on Tabu Search meta-heuristic. It consists of Supervisor Agent and Resource_Scheduler Agents in interaction, trying to find the best scheduling. Experiments made on already existing benchmarks, show the effectiveness of our model in some cases in terms of makespan.

The results were encouraging; it would be interesting to develop other aspects considered in our future researches. We extend our model by introducing a heuristic in the construction of initial solution and diversification phase. We can study another extension of the Job Shop problem with time lags introducing constraints time lags between generalized pair of operations that is very little studied in the literature. We can also propose hybridization of metaheuristics via the Multi-Agent dynamics, such as simulated annealing, ant colonies or the genetic algorithm to better explore the search space to improve the solution.

References

1. Artigues, C., Huguet, M., Lopez, P.: Generalized disjunctive constraint propagation for solving the job shop problem with time lags. Engineering Applications of Artificial Intelligence **24**, 220–231 (2011)
2. Ben Yahia, A., Belkahla Driss, O.: A multi-agent approach for the job shop problem with time lags. In: International Conference on Artificial Intelligence, ICAI, Tunisie (2013)
3. Caumond, A., Gourgand, M., Lacomme, P., Tchernev, N.: Métaheuristiques pour le problème de jobshop avec time lags : Jm|li,s j(i)|Cmax. In: 5ème Confèrence Francophone de Modélisation et SIMulation (MOSIM 2004). Modélisation et simulation pour l'analyse et l'optimisation des systèmes industriels et logistiques, Nantes, France, pp. 939–946 (2004)
4. Caumond, A., Lacomme, P., Tchernev, N.: Feasible schedule generation with extension of the Giffler and Thompson's heuristic for the job shop problem with time lags. In: International Conference of Industrial Engineering and Systems Management, pp. 489–499 (2005)
5. Caumond, A., Lacomme, P., Tchernev, N.: A memetic algorithm for the job-shop with time-lags. Computers Operations Research **35**, 2331–2356 (2008)
6. Deppner, F.: Ordonnancement d'atelier avec contraintes temporelles entre opérations. PhD thesis, Institut National Polytechnique de Lorraine (2004)
7. Fisher, H., Thompson, G.L.: Probabilistic learing combination of local jobshop scheduling rules. In: Industrial Scheduling, pp. 225–251. Prentice Hall (1963)
8. Glover, F.: Future paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research **5**, 533–549 (1986)
9. Heilman, R.: A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. European Journal of Operational Research **144**(2), 348–365 (2003)
10. Hurink, J., Keuchel, J.: Local search algorithms for a single machine scheduling problem with positive and negative time-lags. Discrete Applied Mathematics **112**, 179–197 (2001)
11. Karoui, W., Huguet, M.-J., Lopez, P., Haouari, M.: Méthode de recherche à divergence limitée pour les problèmes d'ordonnancement avec contraintes de délais [Limited discrepancy search for scheduling problems with time-lags]. In: 8ème ENIM IFAC Conférence Internationale de Modélisation et Simulation (MOSIM 2010), Hammamet, Tunisie, May 10–12, 2010
12. Lawrence, S.: Supplement to Resource Constrained Project Scheduling : An experimental investigation of Heuristic Scheduling Techniques. Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA (1984)
13. Lacomme, P., Huguet, M.J., Tchernev, N.: Dedicated constraint propagation for Job-Shop problem with generic time-lags. In: 16th IEEE Conference on Emerging Technologies and Factory Automation IEEE Catalog Number: CFP11ETF-USB, Toulouse, France (2011). ISBN 978-1- 4577-0016-3
14. Mitten, L.G.: Sequencing n jobs on two mahines with arbitrary timlags. Management Science 5(3) (1958)
15. Wikum, E.D., Llewellyn, D.C., Nemhauser, G.L.: One-machine generalized precedence constrained scheduling problems. Operations Research Letters **16**, 87–99 (1994)