archives-ouvertes.fr

Tabu search algorithms to minimize the total tardiness in a flow shop production and outbound distribution scheduling problem

Quang Chieu Ta, Jean-Charles Billaut, Jean-Louis Bouquard

To cite this version:

Quang Chieu Ta, Jean-Charles Billaut, Jean-Louis Bouquard. Tabu search algorithms to minimize the total tardiness in a flow shop production and outbound distribution scheduling problem. International Conference on Industrial Engineering and Systems Management (IESM'2015), Oct 2015, Seville, Spain. hal-01245919

HAL Id: hal-01245919

https://hal-univ-tours.archives-ouvertes.fr/hal-01245919

Submitted on 7 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Heuristic algorithms to minimize the total tardiness in a flow shop production and outbound distribution scheduling problem

(presented at the 6^{th} IESM Conference, October 2015, Seville, Spain) © I^4e^2 2015

Quang Chieu Ta, Jean-Charles Billaut, Jean-Louis Bouquard
Université François Rabelais de Tours, CNRS,
LI EA 6300, OC ERL CNRS 6305,
64 avenue Jean Portalis, 37200 Tours, France
quang-chieu.ta@univ-tours.fr, jean-charles.billaut@univ-tours.fr, jean-louis.bouquard@univ-tours.fr

***Abstract*—In this paper, we consider a production and outbound distribution scheduling problem, coming from a real life problem in a chemotherapy production center. Only one vehicle with infinite capacity is available for delivery. The production workshop is an m-machine flow shop. To each job is associated a processing time per machine, a location site and a delivery due date. The travel times are known. The problem is to define a production schedule, batches of jobs, and delivery routes for each batch, so that the sum of tardiness is minimized. Heuristic algorithms are proposed and evaluated on random data sets.**

I. INTRODUCTION

We consider in this paper the permutation flow-shop scheduling problem and vehicle routing problem (VRP) integrated, also called 'production and outbound distribution scheduling problem' in the literature. The jobs have to be delivered to the customers after their production by using a single vehicle. This problem comes from a real life application in the domain of chemotherapy production ([17], [12]). In this production environment, the coordination of production and delivery at an operational level is very important for several reasons: the patients are waiting for their treatment, and avoiding stress and useless lost of time is important, and injectable products in syringue or pouch have to be delivered without lost of time. The production process is complex [3], but it can be easily approximated by a flow shop process with one stage for the sterilization, one stage for the production of the pouch or syringue, and one stage for the control. In the problem that we consider (and in the case of the hospital of Tours where around 150 preparations are daily performed), there is only one delivery man, so we consider that there is only one vehicle.

More precisely, we consider that there is a set $\mathcal{J} = \{J_1, ..., J_n\}$ of n jobs to schedule on a set $\mathcal{M} = \{M_1, ..., M_m\}$ of m machines organized in a flow shop environment. We denote by $p_{i,j}$ the processing time of J_j on machine M_i and d_j is the delivery due date of J_j. To each job J_j is associated a site j, where the job has to be delivered. The travel time matrix between sites is known and $t_{j1,j2}$ denotes the travel time between site $j1$ and site $j2$ ($\forall j1, j2 \in [0, n]$). It is assumed in the following that the production center is associated to site 0. Notice that in practice, if the delivery to the patients is done inside the hospital, there is not one site per job. The number of sites is limited, and several jobs can be delivered to the same site. However, if the delivery to the patients is done outside (home care services), there is potentially one site per patient with non negligible transportation times.

The problem is to define a schedule of the jobs on the machines, to define batches of jobs (one batch corresponds to one trip of the vehicle). For each batch, the vehicle routing problem consists in defining a route starting from the production site, visiting the customers associated to the jobs in the batch, and finishing at the production site. We define the variables $C_{i,j}$ to denote the completion time of job J_j on machine M_i ($\forall i \in [1, m], \forall j \in [1, n]$), D_j to denote the delivery completion time, T_j to denote the tardiness of J_j, defined by $T_j = \max(D_j - d_j, 0)$. The objective is to minimize the total tardiness of delivery denoted by $\sum T_j = \sum_{j=1}^{n} T_j$.

This problem is clearly an NP-hard problem [15].

The paper is organized as follows. Section II presents a survey of the literature in this domain. In Section III, a linear integer programming formulation of the problem is proposed. In Section IV we present the resolution methods. Three heuristic algorithms are proposed. In Section V some computational experiments are proposed.

II. LITERATURE SURVEY

There are few papers in the literature dealing with integrated production scheduling and vehicle routing problems at an operational level. These problems are also known under the denomination 'production and outbound distribution scheduling'. The survey paper of [5] introduces the problem and proposes a five-field notation $\alpha|\beta|\pi|\delta|\gamma$ to describe the problem. The notation of the problem that we consider is $Fm||V(1,\infty), routing|n|\sum T_j$, where the field $\alpha = Fm$ means that we consider an m-machine flow-shop scheduling problem, β is empty, the field π contains $V(1,\infty)$ meaning that there is only one vehicle with infinite capacity, and $routing$ meaning that orders going to different customers can be transported in the same shipment. In the field δ we have n to indicate that each job belongs to one customer. Finally, $\gamma = \sum T_j$ is the objective function, here the total tardiness of delivery.

Of course, a lot of papers in the literature deal with integrated production and distribution problem. The first paper is certainly the one of Hall and Potts [11], dealing with scheduling and delivery problems in the supply chain. Then, a lot of papers deal with the integration of the scheduling and the batching problem for delivery [14]. In these papers, the customers are supposed to be located in close proximity to each other, as if there was only one customer. Therefore, there is no vehicle routing problem associated to these (already) difficult problems. Notice that the problem denoted by $Fm \rightarrow D|v = 1, c = 1|C_{\max}$, where there is only one vehicle of capacity 1 is strongly NP-hard [14]. In other papers, such as in [19], the production system is considered as a single machine.

We focus here on few papers, where the production and the distribution problem present some similarities with our problem. Some recent references are also reported in the review paper presented in [7].

In [16], the authors consider a single machine problem together with routing decisions of a delivery vehicle (with limited capacity) which serves customers at different locations. The objective function is the minimization of the sum of jobs delivery times. The authors show that the problem is strongly NP-hard and consider a particular case (single-customer), and the general case with fixed number of customer sites for which they propose a dynamic programming algorithm.

In [8], the authors consider a fresh food production and distribution problem. The authors identify three stages: a stage of batch processing of raw materials into food products, a stage for packaging these products and a stage for their immediate distribution. The production environment is complex and sequence dependent setups costs are considered. For the distribution problem, tight time windows at customer location are considered. The authors propose a hierarchical approach, batching the customer orders with similar temperature and processing requirements and compatible delivery and vehicle departure times, and applying a heuristic approach to solve the distribution planning problem.

In [2], the authors consider an integrated production and inventory routing problem. They propose a mixed-integer programming model including a single production facility, a set of customers with time varying demand and a fleet of homogeneous vehicles. A hybrid methodology is proposed to solve the mixed-integer programming model.

In [22] the authors consider an integrated production and distribution planning problem, already studied in [1]. There is a production facility, modeled as a single machine, a single transporter and a fixed sequence of customers. A single product with limited lifespan is produced. Time windows are associated to the deliveries. The authors propose a branch-and-bound algorithm for the problem and extend the original problem to the case where the production start can be delayed and to the case where the production sequence and the routing sequence may be different. The authors propose heuristic algorithms for solving the problem. This model and the constraints considered are not similar to the problem defined in this paper.

III. Integer linear programming formulation

In this Section, we give a linear programming formulation of the problem. The resolution of this model with commercial solvers cannot lead to performing solutions if the problem size is medium. More generally, the complexity of this model prevents the use of exact algorithms for medium size instances.

The data are given by n, m, the processing times $p_{i,j}$, the delivery due dates d_j, and the matrix $t_{j1,j2}$. M is a big value, set to $\sum_i \sum_j p_{i,j}$. The objective function is:

$$\text{Minimize } \sum T_j = \sum_{j=1}^{n} T_j \tag{1}$$

The variables are:

- $y_{j1,j2}$, equal to 1 if job J_{j1} is scheduled before job J_{j2}, 0 otherwise,
- $x_{j1,j2}$, equal to 1 if job J_{j1} is transported before job J_{j2}, assuming that they are transported in the same route, 0 otherwise,
- $z_{j,k}$, equal to 1 if job J_j is transported during route k (at most n tours $k \in [1, n]$), 0 otherwise,
- $C_{i,j}$, ≥ 0, the completion time of job J_j on machine M_i,
- D_j, ≥ 0, the delivery time of job J_j,
- T_j, ≥ 0, the tardiness of job J_j,
- S_k and F_k, ≥ 0, the starting time and the finishing time of route number k.

For the scheduling problem, considering two arbitrary jobs J_{j1} and J_{j2} ($\forall j1 \in [1, n]$, $\forall j2 \in [1, n]$, $j1 \neq j2$), J_{j1} is before J_{j2} or J_{j2} is before J_{j1}.

$$y_{j1,j2} + y_{j2,j1} = 1 \tag{2}$$

On a given machine M_i ($\forall i \in [1, m]$), if J_{j1} is before J_{j2}, we have ($\forall j1 \in [1, n]$, $\forall j2 \in [1, n]$, $j1 \neq j2$):

$$C_{i,j2} \geq C_{i,j1} + p_{i,j2} - M(1 - y_{j1,j2}) \tag{3}$$

For any job J_j, the job completes on machine M_{i-1} before starting on M_i ($\forall i \in [1, m]$, $\forall j \in [1, n]$):

$$C_{i,j} \geq C_{i-1,j} + p_{i,j} \tag{4}$$

Any job J_j completes on the first machine after its duration ($\forall j \in [1, n]$):

$$C_{1,j} \geq p_{1,j} \tag{5}$$

The expression of the tardiness of J_j is ($\forall j \in [1, n]$):

$$T_j \geq D_j - d_j \tag{6}$$

One job J_j belongs necessarily to one tour k ($\forall j \in [1, n]$, $\forall k \in [1, n]$):

$$\sum_{k=1}^{n} z_{j,k} = 1 \tag{7}$$

There is necessarily one job (in $[0, n]$) before and after any job J_j in a tour ($j \in [1, n]$):

$$\sum_{j1=0}^{n} x_{j1,j} = 1 \tag{8}$$

$$\sum_{j2=0}^{n} x_{j,j2} = 1 \tag{9}$$

In a tour, J_{j1} is before J_{j2} or J_{j2} is before J_{j1} ($\forall j1 \in [1, n]$, $\forall j2 \in [1, n]$, $j1 \neq j2$) or there is no relation between them:

$$x_{j1,j2} + x_{j2,j1} \leq 1 \tag{10}$$

If job J_{j1} and job J_{j2} are in the same tour ($\forall j1 \in [1, n]$, $\forall j2 \in [1, n]$, $j1 \neq j2$, $\forall k \in [1, n]$), one variable $x_{j1,j2}$ or $x_{j2,j1}$ is equal to 1:

$$x_{j1,j2} + x_{j2,j1} \geq z_{j1,k} + z_{j2,k} - 1 \tag{11}$$

Route k can only start after the end of previous routes ($\forall k1 \in [1, n-1]$, $\forall k2 \in [k1, n]$):

$$S_{k2} \geq F_{k1} \tag{12}$$

Route k can only start after the completion of all the jobs transported ($\forall j \in [1, n]$, $\forall k \in [2, n]$):

$$S_k \geq C_{m,j} - M(1 - z_{j,k}) \tag{13}$$

The delivery of a job cannot be before the starting time of the route plus the transportation time from the production site to the customer ($\forall j \in [1, n]$, $\forall k \in [2, n]$):

$$D_j \geq S_k + t_{0,j} - M(1 - z_{j,k}) \tag{14}$$

The finishing time of route k is after the vehicle returns to the production site ($\forall j \in [1, n]$, $\forall k \in [2, n]$):

$$F_k \geq D_j + t_{j,0} - M(1 - z_{j,k}) \tag{15}$$

The delivery time of J_{j2} is after the delivery time of J_{j1} if J_{j1} is before J_{j2} in the same route ($\forall j1 \in [1, n]$, $\forall j2 \in [1, n]$, $j1 \neq j2$):

$$D_{j2} \geq D_{j1} + t_{j1,j2} - M(1 - x_{j1,j2}) \tag{16}$$

Some cuts can be added in the model in order to improve its resolution. For example, if J_{j1} is scheduled before J_{j2}, then the tour of J_{j1} is not after the tour of J_{j2} ($\forall j1 \in [1, n]$, $j2 \in [1, n]$, $j1 \neq j2$):

$$\sum_{k=1}^{n} k \times z_{j1,k} \leq \sum_{k=1}^{n} k \times z_{j2,k} + M(1 - y_{j1,j2}) \tag{17}$$

This model contains $3n^2 + n$ binary variables and $4n + mn$ continuous variables and $n^3 + 9n^2 + nm + 4n + \frac{1}{2}n(n-1)$ constraints. This model contains a lot of constraints with the 'big M' parameter and therefore the linear relaxation of this model yields to poor lower bounds. So the solver cannot cut a lot and at then end, the model cannot be solved to optimality for medium size instances in a reasonable computation time. Only decomposition methods (column generation, Benders decomposition, ...) can be used for having optimal solutions in a reasonable time and only for medium size instances. For the real-life problem that we consider, instances contain arround 150 jobs, so the use of exact approaches has not been investigated further.

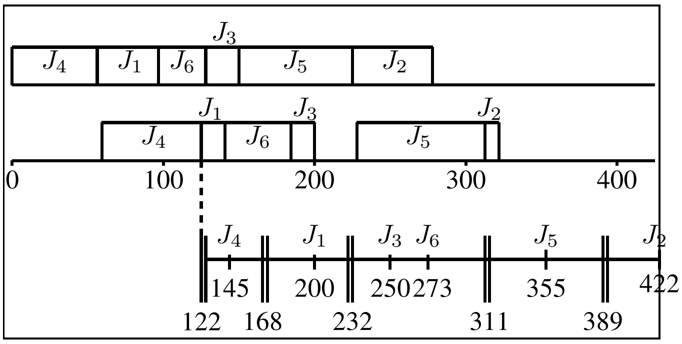

Fig. 1. Gantt representation of the solution

Example

We illustrate the problem with the following instance. For example, let consider the 6-job 2-machine instance described in Table I.

TABLE I. INSTANCE WITH 6 JOBS AND 2 MACHINES

j	1	2	3	4	5	6
$p_{1,j}$	40	49	22	58	75	29
$p_{2,j}$	17	8	12	64	85	47
d_j	203	422	241	68	359	293

$(t_{j1,j2}) =$

	0	1	2	3	4	5	6
0	0	32	33	18	23	34	38
1	32	0	42	43	53	66	50
2	33	42	0	21	53	57	7
3	18	43	21	0	32	36	23
4	23	53	53	32	0	15	56
5	34	66	57	36	15	0	58
6	38	50	7	23	56	58	0

The optimal solution is given by the sequence $(J_4, J_1, J_6, J_3, J_5, J_2)$ for the flow shop scheduling problem. Then, each single job constitutes a batch of delivery, except jobs J_6 and J_3 which are transported in the same batch, J_3 first and then J_6. The value of the objective function is equal to 86. The values of the variables $C_{1,j}$, $C_{2,j}$, D_j and T_j are given in Table II. The composition, the starting time and finishing time of each route are given in this Table as well. Figure 1 gives a Gantt chart representation of the optimal solution.

TABLE II. RESULT FOR THE INSTANCE WITH 10 JOBS AND 2 MACHINES

j	1	2	3	4	5	6
$C_{1,j}$	58	98	127	149	224	273
$C_{2,j}$	122	139	186	198	309	317
D_j	200	422	250	145	355	273
T_j	0	0	9	77	0	0

k	1	2	3	4	5
jobs	J_4	J_1	J_6, J_3	J_5	J_2
S_k	122	168	232	311	389
F_k	168	232	311	389	455

IV. Resolution methods

Several heuristic algorithms are proposed in this section.

A. Greedy algorithm

The first algorithm proposed for finding an initial solution is the following greedy algorithm. Starting from a sorting of the jobs in EDD order, batches of equal size are defined. This solution can be coded in a $2n$ size vector containing for each batch the number of jobs in the batch and the list of jobs. The vector finishes with some 0 if necessary.

The EDD order is $(J_4, J_1, J_3, J_6, J_5, J_2)$. If the number of batches is equal to 3, batch 1 will contain jobs (J_4, J_1), batch 2 will contain jobs (J_3, J_6), and batch 3 will contain jobs (J_5, J_2). Such a solution is represented by the following vector:

$$V = (\boxed{2}, 4, 1, \boxed{2}, 3, 6, \boxed{2}, 5, 2, \boxed{0}, 0, 0)$$

The evaluation of such a vector is described by Algorithm 1. The scheduling problem is solved by using NEH algorithm [18] and CDS algorithm [4], assuming that the machines are available at dates R_i ($i \in [1, m]$) and the sequence with minimum makespan is kept. The objective function here is the makespan minimisation because once the batch is defined, the best solution is obtained when the vehicle starts as early as possible. The optimization of the total tardiness of delivery for this batch is taken into account in the next step. For the routing of the jobs, two heuristic algorithms are also applied. In the first one, the nearest neighbor is chosen, in the second, the EDD order is considered for the delivery, assuming that the vehicle is only available at time t. Again, the best routing sequence is kept. Machine release dates and vehicle availability are updated for the next iteration (next batch).

Algorithm 1 Vector evaluation

Input: vector V
Initialise dates $R_i = 0$, $\forall i \in [1, m]$
Initialise date $t = 0$
for each batch k **do**
– Compute a schedule by using NEH algorithm assuming that the machines are available at dates R_i.
– Compute a schedule by using CDS algorithm assuming that the machines are available at dates R_i.
– Keep the schedule with minimum makespan and update dates R_i.
– Compute a routing for the vehicle using Nearest neighbor heuristic, assuming that the vehicle is available at time $\max(t, R_m)$.
– Compute a routing for the vehicle using EDD heuristic, assuming that the vehicle is available at time $\max(t, R_m)$.
– Keep the best route (total tardiness minimisation) and update t.
end for
Return (total tardiness)

For the example under consideration, the vector has an evaluation of 131. The production schedule is given by sequence $(J_4, J_1, J_3, J_6, J_5, J_2)$, the routing is given by $J_4 \prec J_1$, then $J_3 \prec J_6$ and finally $J_2 \prec J_5$.

The greedy algorithm that we propose is described in Algorithm 2.

Algorithm 2 Greedy algorithm GR

S = the jobs sorted in EDD order
$UB = \infty$
for b in 1 to $n/2$ **do**
– Build a vector V with b batches, i.e. each batch contains $\lceil n/b \rceil$ jobs (except the last one that contains $n - (b - 1)\lceil n/b \rceil$ jobs).
– Evaluate V with Algorithm 1 and update UB if it leads to a better solution
end for
Return (UB)

One difficulty in this method is the intensive use of NEH algorithm. This algorithm is known for being very efficient for solving the $Fm||C_{\max}$ problem, but for large-scale problems, its running time is very long. Its complexity is in $O(n^3 m)$, even if it can be reduced to $O(n^2 m)$ [20], whereas the complexity of CDS is in $O(nm^2 + mn\log(n))$. Finally, the whole complexity of Algorithm 2 is in $O(n^3 m + n^2 m^2)$, which is not negligible for instances with important values of n and m, as we will see in Section V.

B. Tabu search algorithm

Tabu search (TS) has been initially proposed by Glover [9], [10]. TS is a metaheuristic local search algorithm that begins with an initial solution and successively moves to the best solution in the neighborhood of the current solution. The algorithm maintains a list of forbidden solutions, to prevent the algorithm from visiting solutions already examined (these solutions are called *tabu*). The elements of our TS algorithm are described below.

A solution is coded by the vector V already presented, and is evaluated by Algorithm 1. The initial solution is the solution given by the Greedy Algorithm 2.

Then, several neighborhood operators are applied to this vector V:

- SWAP($V, j1, j2$) operator allows to swap two jobs J_{j1} and J_{j2}, belonging to two different batches,
- EBSR($V, j1, j2$) for "Extract and Backward Shift Reinsertion", extracts a job J_{j2} belonging to a batch $b2$, and re-insert this job before job J_{j1}, belong to a batch $b1$, before batch $b2$,
- EFSR($V, j1, j2$) for "Extract and Forward Shift Reinsertion", extracts a job J_{j1} belonging to a batch $b1$, and re-insert this job after job J_{j2}, belong to a batch $b2$, after batch $b1$.

These basic neighborhood operators are applied for all couples of positions $(k1, k2)$ with $k1 < k2$ (job J_{j1} is on position $k1$ and job J_{j2} is on position $k2$), and it is clear that J_{j1} and J_{j2} do not belong to the same batch ($k2$ starts with the position of the first job in the next batch).

One element of the Tabu list contains four items: $(j1, j2, b1, b2)$, i.e. the jobs index and their batch numbers.

The Tabu search algorithm is briefly described in Algorithm 3. UB denotes the current value of the best neighbor, BNV indicates the Best Neighbor Vector. The stopping criterion is a limit of computation time.

Algorithm 3 Tabu Search algorithm TS

Input: V = the solution returned by the Greedy algorithm 2
while stopping criterion not met **do**
  – $UB = \infty$
  for all pairs $(k1, k2)$, $k1 < k2$ **do**
    – test SWAP$(V, j1, j2)$ if not Tabu and update BNV and the Tabu list if necessary
    – test EBSR$(V, j1, j2)$ if not Tabu and update BNV and the Tabu list if necessary
    – test EFSR$(V, j1, j2)$ if not Tabu and update BNV and the Tabu list if necessary
  end for
  – Update the current solution $V \leftarrow BNV$, update UB
end while
Return (UB)

C. Combined heuristic

A combined heuristic CH between the Greedy Algorithm GR and the Tabu Search TS is also proposed. This algorithm applies the Tabu Search Algorithm to the vector generated at each iteration ("**for** b in 1 to $n/2$ **do**") of GR, and returns the best found solution. This method is a sort of multi-start Tabu Search.

This method is potentially better than GR and TS, except for the computation time. And because the computation time will be limited, we will see that it can lead, for some big instances, to worse solutions than TS and than GR. So for this method, a second version called $CH2$ has been tested where the neighborhood is limited ($k2$ cannot be greater than $k1 + \delta$).

V. Computational experiments

We present in this section the generation of data, and we discuss the results.

A. Generation data

Data sets have been randomly generated. Notice that there is no benchmark instance for the m-machine flowshop and vehicle routing problem integrated, although benchmarck instances do exist for the m-machine flowshop problem with total tardiness minimization ([21] where several heuristic algorithms are extensively tested). Processing times $p_{i,j}$ have been generated in interval [1,100]. Due dates d_j have been generated in $[50, 50n]$. The geographical coordinates of site j are generated in $[0, \alpha\frac{100}{\sqrt{2}}]$ (see **Fig. 2**). The travel time $t_{i,j}$ is the classical euclidian distance:

$$t_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

If α is equal to 1, the maximum distance between two sites is equal to 100, i.e. traveling times and processing times are in the same order of magnitude. If α is less than 1, the travel times are smaller than the processing times, and it is the contrary if α is greater than 1.

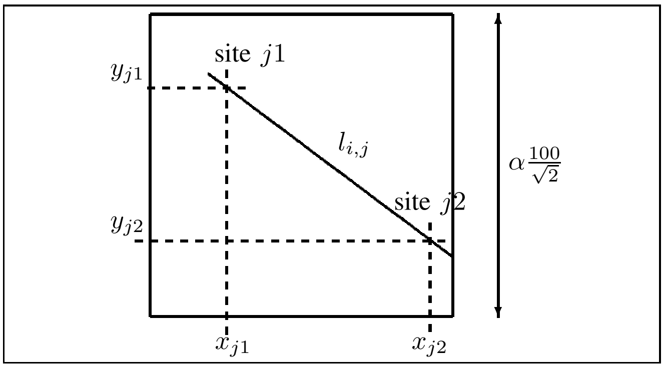

Fig. 2. Illustration of calculation of $t_{j1,j2}$

Thirty instances are generated for each combination of n and m, with $n \in \{20, 50, 100, 150, 200\}$ and $m \in \{5, 10\}$, leading to 300 instances per value of α.

We define CLASS 0 the instances where $\alpha = 0.75$, CLASS 1 the instances where $\alpha = 1.00$ and CLASS 2 the instances where $\alpha = 1.25$. In this paper, we only report the results obtained with CLASS 1.

B. Results

We present in this section the computational results. In Table III, columns m and n indicate the size of the instances, column 'BestH' indicates the average best value, then for each heuristic algorithm H, one column indicates the average objective function value$\sum T_j(H)$, the average computation time (in seconds), the number of times the method gives the best solution (#best) and the average deviation to the best solution Δ.

$$\Delta_H = \frac{\sum T_j(H) - Best \sum T_j}{\sum T_j(H)}$$

GR indicates the greedy algorithm, TS refer to the Tabu Search algorithm with a Tabu list of 10 elements, and CH to the combined heuristic. The computation time has been limited to 300 seconds for all algorithms.

The results show the dominance of the Tabu Search. The Combined Heuristic CH is efficient for the small instances, with up to 50 jobs, but for larger instances, the $CH2$ with limited neighborhood is better.

VI. Conclusion

We approach a problem where a m-machine permutation flow shop scheduling problem and a vehicle routing problem are integrated, and the objective is to minimize the total delivery tardiness. We present an MILP formulation of the problem, a greedy algorithm and Tabu Search based heuristics with an indirect coding for a solution. Some computational experiments are conducted and the first results show that the Tabu Search greatly improves the initial solution given by GR.

In the future, it could be interesting to propose lower bounds for this problem. The scheduling problem and the

TABLE III. RESULTS FOR CLASS 1 INSTANCES (1)

m	n	BestH	$\sum T_j(GR)$	CPU	Δ_{GR}	#best	$\sum T_j(TS)$	CPU	Δ_{TS}	#best
			GR				TS			
5	20	4953,9	9170,2	0	0,455	0	5356,7	0	0,061	20
5	50	15432,2	29346,2	0	0,471	0	18628,6	1,9	0,142	13
5	100	38543	83060,1	0	0,524	0	39760	44,7	0,024	25
5	150	65062,4	141633,5	0	0,525	0	69323,7	261,7	0,049	24
5	200	110566,8	210017,8	0	0,465	0	119708,3	304,6	0,063	20
10	20	12051,2	16950,9	0	0,284	0	12637,8	0,1	0,042	16
10	50	34846,1	54884,2	0	0,361	0	38166,6	4,4	0,079	14
10	100	83454,6	130753,6	0	0,351	0	90157,4	79,8	0,059	20
10	150	139087,8	232448,8	0	0,392	0	145136,1	302,3	0,032	24
10	200	258808,3	354125	0,1	0,268	0	269366,4	322,6	0,037	11

TABLE IV. RESULTS FOR CLASS 1 INSTANCES (2)

m	n	BestH	$\sum T_j(CH)$	CPU	Δ_{CH}	#best	$\sum T_j(CH2)$	CPU	Δ_{CH2}	#best
			CH				$CH2$			
5	20	4953,9	6006,5	1,3	0,2	10	7025,2	0,2	0,3	0
5	50	15432,2	17348,1	191,8	0,092	17	21281,067	10,845	0,259	0
5	100	38543	147849,9	300	0,729	0	59370,4	267,624	0,31	5
5	150	65062,4	373769,4	300	0,826	0	100582,4	300,232	0,313	6
5	200	110566,8	655918,3	300	0,832	0	148460,933	300,477	0,215	10
10	20	12051,2	12969,2	2,7	0,063	14	14545,5	0,329	0,164	0
10	50	34846,1	37792,4	299,9	0,069	16	46418,367	16,947	0,242	0
10	100	83454,6	193939,3	300	0,562	0	105381,467	293,261	0,185	10
10	150	139087,8	440612,5	300	0,682	0	182707,767	300,316	0,217	6
10	200	258808,3	743212,1	300,1	0,653	0	282373,7	300,7	0,076	19

vehicle routing problem being already difficult, finding good lower bounds seems to be very challenging. The resolution of the problem to optimality seems also to be a challenging problem. For this research direction, a model with less 'big-M' constraints can certainly be proposed, and decomposition methods seem to be research directions to investigate for such a difficult problem ([13]). Some other metaheuristic methods can be developed. A Tabu Search algorithm with a direct encoding can be proposed, as well as a genetic algorithm and a simulated annealing algorithm, known for its efficency for the two-machine scheduling problem. Then, the combination of mathematical programming and local search (*matheuristic* in the literature or hybrid optimization, see [6]) can be used, in order to improve the efficiency of the resolution methods. Hybrid methods seem very efficient for such difficult problems.

ACKNOWLEDGMENTS

This work was supported by the financial support of the Vietnamese government and by the ANR ATHENA project, grant ANR-13-BS02-0006 of the French Agence Nationale de la Recherche.

REFERENCES

[1] Armstrong, R., Gao, S., and Lei, L. A zero-inventory production and distribution problem with a fixed customer sequence. Annals of Operations Research, 159(1), 395414, 2008.

[2] J.F. Bard, N. Nananukul, A branch-and-price algorithm for an integrated production and inventory routing problem. Computers and Operations Research 37:2202-2217, 2010.

[3] J-C. Billaut, T. Drevon, J-F. Tournamille, A complete view of the scheduling problem of chemotherapy production with expensive and perishable raw materials, 20th Conference of the International Federation of Operational Research Societies (IFORS2014), Barcelona, July 2014.

[4] Campbell, H. G., Dudek, R. A., and Smith, M. L., A heuristic algorithm for the n job, m machine sequencing problem. Management Science, 16(10):B630B637, 1970.

[5] Chen Z-L., Integrated Production and Outbound Distribution Scheduling: Review and Extensions Operations Research, 58(1), 130148, 2010.

[6] F. Della Croce and M. Ghirardi and R. Tadei, *Recovering Beam Search: enhancing the beam search approach for combinatorial optimization problems.* Journal of Heuristics, 10:89–104, 2004.

[7] B. Fahimnia, R. Z. Farahani, R. Marian, L. Luong, A review and critique on integrated productiondistribution planning models and techniques Journal of manufacturing systems, 32(1):1-19, 2013.

[8] P. Farahani, M. Grunow, H-O. Gnther, Integrated production and distribution planning for perishable food products. Flexible Services and Manufacturing Journal, 24:2851, 2012.

[9] F. Glover, *Tabu Search - Part I.* ORSA Journal on Computing, 1(3):1990–206, 1989.

[10] F. Glover, *Tabu Search - Part II.* ORSA Journal on Computing, 2(1):4–32, 1990.

[11] Hall, N. G., and Potts, C. N., Supply chain scheduling: batching and delivery. Operations Research, 51(4), 566584, 2003.

[12] Y. Kergosien, J-F. Tournamille, B. Laurence, J-C. Billaut, Planning and Tracking Chemotherapy Production for Cancer Treatment: a Performing and Integrated Solution, International Journal of Medical Informatics, vol. 80 (9), 655-662, 2011.

[13] Y. Kergosien, M. Gendreau, J-C. Billaut, A Benders decomposition based heuristic for a combined transportation and scheduling problem in chemotherapy production, Operational Research Applied to Health Services (ORAHS13), Istanbul, July 2013.

[14] Lee, C. Y., and Chen, Z. L., Machine scheduling with transportation considerations. Journal of Scheduling, 4, 324, 2001.

[15] J. K. Lenstra and A. H. G. Rinnooy Kan, *Complexity of vehicle routing and scheduling problems.* Networks, 11,:221–227, 1981.

[16] Li, C. L., Vairaktarakis, G., Lee, C. Y., Machine scheduling with deliveries to multiple customer locations. European Journal of Operational Research, 164(1), 39-51, 2005.

[17] A. Mazier, J-C. Billaut, J-F. Tournamille, Scheduling preparation of doses for a chemotherapy service, Annals of Operations Research, vol. 178 (1), pp. 145-154, 2010.

[18] Nawaz, M., Enscore, Jr, E. E., and Ham, I., A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. OMEGA, The International Journal of Management Science, 11(1):9195, 1983.

[19] G. Steiner and R. Zhang, Approximation algorithms for minimizing the total weighted number of late jobs with late deliveries in two-level supply chains. Journal of Scheduling, 12: 565574, 2012.

[20] E. Taillard, Some efficient heuristic methods for the sequencing problem. European Journal of Operational Research, 47:65-74, 1990.

[21] E. Vallada, R. Ruiz, and G. Minella, *Minimising total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and*

metaheuristics. Computers and Operations Research, 35(4):1350–1373, 2008.

[22] C. Virguz, S. Knust, Integrated production and sitribution scheduling with lifespan constraints. Annals of Operations Research, 213:293-318, 2014.

[23] C. C. Wu and W. C. Lee and T. Chen, *Heuristic algorithms for solving the maximum lateness scheduling problem with learning considerations.* Computers & Industrial Engineering, 52:124–132, 2007.