Journal of Industrial and Systems Engineering
Vol. 10, No. 1, pp 50- 60
Winter (January) 2017

A branch and bound algorithm to minimize the total weighted number of tardy jobs and delivery costs with late deliveries for a supply chain-scheduling problem

Morteza Rasti-Barzoki[1*], Seyed Reza Hejazi[1]

[1]*Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan Iran*

rasti@cc.iut.ac.ir, rehejazi@cc.iut.ac.ir

Abstract

In this paper, we study a supply chain scheduling problem that simultaneously considers production scheduling and product delivery. n Jobs have to be scheduled on a single machine and delivered to K customers for further processing in batches. The objective is to minimize the sum of the total weighted number of tardy jobs and the delivery costs. In this paper, we present a heuristic algorithm (HA) and a branch and bound (B&B) method for the restricted case, where the tardy jobs are delivered separately, and compare these procedures with an existing dynamic programming (DP) algorithm by computational tests. The results of computational tests show significant improvement of the B&B over the dynamic programming algorithm.

Keywords: Supply chain scheduling, batch delivery and tardy job, branch and bound

1- Introduction

Classical scheduling problems did not consider distribution costs so considering both the delivery costs and scheduling objectiveis an important issue that researchers have paid attention torecently. Lack of integration between production and distributionschedules yields substantial inefficiencies and, consequently, poor total system performance. To achieve optimal operational performance in a supply chain, it is critical to integrate these two problems (Chen, 2010).Chen (2010) presented an important review on the literature regarding integrated production and distribution scheduling models. In the current paper weaddress the minimizing the sum of the total weighted number of tardy jobs and the delivery costs for multi-customer and present a heuristic algorithm and a branch and bound method.

Moore's algorithm (Moore, 1968) solves the minimum number of tardy jobs on the single machinein $O(nlogn)$ time. The weighted version problem is hard (Karp, 1972) and sahni (1976) provided a dynamic programming and a fully polynomial time approximation scheme for solving it.Hallah and Bulfin (2003, 2007) considering zero ready time and non-zero ready time developedthe branch and bound methods for this problem.

---

*corresponding author
ISSN: 1735-8272,

A dynamic programming algorithm was proposed for the batching version of the problem in which jobs are processed in batches which require setup time by Hochbaum and Landy (1994) and later Brucker and Kovalyov (1996) improved it. Nevertheless, none of these studiesconsidered the delivery costs.

Hall and Potts (2003) considered the problem of scheduling the jobs under the batch availability assumption and for a single machine with several objectives including sum of flow times, maximum lateness and number of late jobs. Batch availability means that all the jobs forming a batch become available for dispatch only when the entire batch has been processed and presented several dynamic-programming algorithms for solving those. Mazdeh et al.(2007, 2008),considering multi-customer with zero or non-zero ready time,presenteda B&B method for minimizing the sum of the total flow time and delivery costs. Also Mazdeh et al. (2011) presented optimal properties and a B&B method for a scheduling and batchingproblem considering minimizing the sum of the total weighted flow time and delivery costs on the single machine. They compared their method with the dynamic programming presented by Ji et al. (2007). Steiner and Zhang (2007) surveyed the similar problem, i.e. scheduling and batch delivery to a customer, considering minimizing the sum of the total weighted number of tardy jobs and delivery costs on the single machine with batch setup time and presented a DP algorithm for the optimal solution. They (Steiner and Zhang, 2009) alsopresented a pseudo polynomialDP, we name it DP^r, for restricted case of multi-customer, where the tardy jobs are delivered separatelyat the end of schedule. In the current paper, we present a B&B method forthis problem and compare it with DP introduced by Steiner and Zhang (2009) using computational tests.Recently, Rasti-Barzoki et al. (2013)studied the minimizing of sum of the total weighted number of tardy jobs and the delivery costs for single customer.

In the current paper for the first time, we propose a heuristic algorithm and a Branch and Bound procedure for minimizing sum of the total weighted number of tardy jobs and delivery costs with late deliveries.

The rest of the current paper is organized as follows.Section 2 contains the problemdefinition. Sections 3 and 4 contain two mentioned algorithms including the HA and the B&Bstructure respectively. An example is provided in section 5. Section6analyses the computational tests and the last section contains our conclusions.

2- Problem definition

Notation:

j	Job index
k	Customer index
n	Number of jobs
K	Number of customers
s_k	Batch setup time for jobs belong to customer k
p_j	Processing time of job j
d_j	Due date of job j
w_j	Weight of job j
U_j	One if job j be tardy and zero otherwise
θ_k	Delivery costs for sending batch to customer k
TB^k	Tardy batch for customer k
ω_c	Sets of close batches
ω_o	Open batch
ω_u	Set of un-surveyed jobs in partial scheduling
ω_e	Sets of TB^k
ω^k	Jobs in ω belonging to customer$k$, $\omega \in \{\omega_o, \omega_u, \omega_e\}$
$c(.)$	Customer (s) index of job (s).
B_k	Number of batches for customer k

Similar to Steiner and Zhang (2009), we assume that there are K customers and one manufacture in which each customer k orders $n_k$jobs to the manufacture and $n = \sum_{k=1}^{K} n_k$ is the number of jobs. No job can be preempted and each job has an important coefficient w_j. It is assumed that each job need one operation and manufacture does it by single machine, with p_j processing time. The due date of job j is d_j. Processed jobs are delivered in batches to each customer. A batch can contain jobs only for same customer and has a sequence-independent-batch setup time s_k. As a restricted case, it is assumed that all tardy jobs have to be delivered in a separate batch for each customer at the end of schedule as tardy batch (*TB*). There is a sufficient number of vehicles with unlimited capacity.The delivery cost for each customer is independent of batch size.Batching and sending several jobs in batches will reduce the transportation cost.The objective is minimizing the sum of the total weighted number of tardy jobs and delivery costs. Using the notation for integrated production and distribution schedulingproblems introduced by Chen (2010), we denote ourproblem as$1/s/V\ (\infty,\infty)\ ,direct/K/\sum_{j=1}^{n} w_j\, U_j + \sum_{k=1}^{K} \theta_k\, B_k$. This notation means that there is the single machine environment for processing jobs ("1") with batch setup time ("s") and sufficient vehicle with unlimited capacity ("$V\ (\infty,\infty)$") and directing delivery method ("$direct$") for sending the batches to ("K") customers.*Directing delivery method* means that orders are transmitted to each customer separately. $\sum_{j=1}^{n} w_j\, U_j$is the total weighed number of tardy jobs and $\sum_{k=1}^{K} \theta_k\, B_k$ is the total delivery costs. Without loss of generality, we re-index the jobs according to the EDD order, i.e. $d_1 \le d_2 \le \cdots \le d_n$.

It is obvious that the problems consideringdelivery costs are more complex than classical scheduling problems. Hall and Potts (2003) showed that the $1//V\ (\infty,\infty)\ ,direct/1/\sum_{j=1}^{n} w_j\, U_j + \theta B$ problem is ordinary NP-hard. Steiner and Zhang (2009) presented a pseudo-polynomial DP algorithm with$O(n^{K+1} min\{D, P+S, W+\theta\})$complexity in which $D = max(d_j)$, $P = \sum_{j=1}^{n} p_j$, $S = \sum_{k=1}^{K} s_k$, $\theta = \sum_{k=1}^{K} \theta_k$, $W = \sum_{j=1}^{n} w_j$for the above problem.

3- A heuristic algorithm

In this section, a heuristic algorithm for the mentioned problem is developed for overcoming the excessive computational time needed for solving the mathematical model. The primary advantage of a heuristic algorithm is its efficiency in running time. Our proposed algorithm is given below. By implementing the following algorithm, a sequence is obtained in which the on-time jobs will be sit in the first and (if necessary) a tardy batch for each customer will be sit in the last of the sequence.

1. Re-index the jobs according to the EDD order (if necessary)

2. Among the jobs $1, 2, \dots, n$ name the first tardy job index as n' by considering a related batch setup timebefore each job (for any more than two consecutivejobs with the same customer, consider only one related batch setup time for that jobs). Among the jobs $1, 2, \dots, n'$ remove the job with the largest processing time/weight ratio and send it to the final related*TB*.

3. Repeat thestep 2 until a sequence is obtained in which the on-time jobs sit first and some of the jobs form the *TB*s.

4. Consider the first on-time job as the start of an open batch, ω_o.If any on-time job does not exist, consider one batch for each customer.

5. Set $ko = c(\omega_o)$.

6. If no other on-time job exists
    Go to the step 9;
   Otherwise,
    Name the first un-surveyed on-time job as$q$ and the first jobof customer ko as $\acute{q}$
    If $q' = \emptyset$
      Close the open batch and go to the step 4;
    Otherwise
      Go to the step 8.

8. Add$\acute{q}$ to the open batch. Let N_O be the set of jobs in the open batch that becomes tardy. Do one of the following:

8.1. If $N_O = \emptyset$,

Add$\acute{q}$ to the open batch and go to the step 6.

8.2. If $\sum_{j\in N_O} w_j \geq \theta_{ko}$,

Close the open batch without $\acute{q}$ and consider q as a new open batch and go to the step 5.

8.3. If $\sum_{j\in N_O} w_j < \theta_{ko}$,

Remove the job with the largest $\frac{p_j}{w_j}$ from the set $N_O + \{\acute{q}\}$and add it to the related *TB*, i.e. TB^{ko}.

If the selected job is$\acute{q}$

Go tothe step 5.

Otherwise, If it belongs to N_O

UpdateN_O and repeat the current step, i.e. 8.3, until $N_O \neq \emptyset$ (As $N_O = \emptyset$, add $\acute{q}$ to the open batchand go to the step 5.

9. Close the open batch (if any) and calculate the objective value for the obtained solution.

4- Branch and Bound method

In this paper, a B&B method is provided for the above problem. The proposed B&B has n levels (excluding the zero level) where in each level the position of one job is determined. A batch is said to be closed if its last job is determined. Otherwise, it is said to be open. In general, a partial scheduling and related batching can be presented by $\omega_c\omega_o\omega_u\omega_e$ in which ω_c is the sets of close batches and ω_o is an open batch. $\omega_u$is the set of jobs whose positions have not been determined yet.ω_e is the set of separated batches for each customer as *TB*s. In general in each branching, we make a decision about the first job in ω_u, namely q, or about the first job in ω_u with similar customer as customer of ω_o, namely $\acute{q}$.For each node four branches are generated:in branches one and two we make a decision about $\acute{q}$ and in branch three and four we make a decision about q as follows:

Branch 1) $\acute{q}$ is added toω_o;

Branch 2) $\acute{q}$goes to the *TB* with related customer, i.e. $TB^k$where $k = c(\acute{q})$;

Branch 3) $\omega_o$is closed and q starts a new open batch;

Branch 4) q goes to the related *TB*, i.e.$TB^k$where $k = c(q)$.

In the following, we first present pruning rules and then discuss about upper bound and lower bound calculation.

4-1- Pruning rules

Regarding the mentioned properties in the section two and the B&B structure introduced in previous section, the following pruning rules can be used to decrease the search space:

P1) If a node isa leaf, i.e. we have reached a solution so fathom this node.

P2) If $LB \geq UB$, fathom the related node.

If a customer of an open batch does not have any other job in the un-surveyed jobs, the open batch must be close; therefore:

P3) In each node with $\omega_u^k = \emptyset$ where$k = c(\omega_o)$, close the open batch.

If there is not an open batch, in one branch the first job inthe un-surveyed jobs form a new open batch and in another branch the mentioned job goes to the related *TB*; therefore:

P4) Branches 1 and 2 are not considered whenever $\omega_o = \emptyset$.

If there exists an open batch and its customer has at least one job in the un-surveyed jobs and if the delivery cost of this customer exceeds the sum of the weights of jobs belonging to the open batch, and if the un-surveyed jobs with the same customer as the open batch customer, it is obvious that one batch for these jobs will be optimum. Therefore, we will have:

P5) If$\omega_o \neq \emptyset, \omega_u^k \neq \emptyset$and $\theta_k \geq \sum_{j\in\omega_o\omega_u^k} w_j$ where $k = c(\omega_o)$, in one branch add$\omega_o\omega_u^k$ to TB^k and if by adding $\omega_o\omega_u^k$ to ω_o, none of the jobs in the open batch become tardy add $\omega_u^k$to ω_o and close this batch.

According to the assumption, a tardy job does not sit between the on time jobs; therefore, the following rules establish:

P6) If$\omega_o \neq \emptyset$, $c(q) \neq c(\omega_o)$ and $d_q < \left(\sum_{k\in c(\omega_c\omega_o)} s_k + \sum_{j\in\omega_c\omega_o} p_j\right) + s_{c(q)} + p_q$ or $\omega_o \neq \emptyset$, $c(q) = c(\omega_o)$ and $d_q < \left(\sum_{k\in c(\omega_c\omega_o)} s_k + \sum_{j\in\omega_c\omega_o} p_j\right) + p_q$, branch 1, 2 and 3 are not created.

P7) If $\omega_o = \emptyset$ and $d_q < \left(\sum_{k\in c(\omega_c\omega_o)} s_k + \sum_{j\in\omega_c\omega_o} p_j\right) + p_q$, branches1, 2 and 3 are not created.

P8) If $d_q < \left(\sum_{k\in c(\omega_c\omega_o)} s_k + \sum_{j\in\omega_c\omega_o} p_j\right) + p_q$, branch 3 is not created.

P9) If $\omega_u^k \neq \emptyset$,where $k = c(\omega_o)$, and $d_{\acute{q}} < \left(\sum_{k\in c(\omega_c\omega_o)} s_k + \sum_{j\in\omega_c\omega_o} p_j\right) + p_{\acute{q}}$, branch 1 is not considered.

P10) If $\omega_o \neq \emptyset, \omega_u^k \neq \emptyset$, where $k = c(\omega_o)$, and $d_{\omega_o(1)} \leq \left(\sum_{k\in c(\omega_c\omega_o)} s_k + \sum_{j\in\omega_c\omega_o} p_j\right) + p_{\acute{q}}$ where $\omega_o(1)$ is the first job in ω_o, branch 1 is not considered.

If the customer of an open batch is same as the first un-surveyed job customer, the nodes that created by branches2 and 4 become same; therefore:

P11) If $c(q) = c(\omega_o)$, branch 2 is not created.

If one batch for each customer causes that none of the jobs become tardy, one batch for each customer is optimum; therefore:

P12) If $d_q \geq \left(\sum_{k\in c(\omega_c\omega_o)} s_k + \sum_{j\in\omega_c\omega_o} p_j\right) + \left(\sum_{k\in c(\omega_u)\backslash c(\omega_o)} s_k + \sum_{j\in\omega_u} p_j\right)andd_{\omega_o(1)} \geq \left(\sum_{k\in c(\omega_c\omega_o)} s_k + \sum_{j\in\omega_c\omega_o} p_j\right) + \sum_{j\in\omega_u^k} p_j$, where $k = c(\omega_o)$, one leaf is created as follows:

Add ω_u^k to ω_o and close ω_o;

For each$k \in c(\omega_u)$:

If $TB^k = \emptyset$, All jobs ofthe customer $k$create one batch

If $TB^k \neq \emptyset$and$\sum_{j\in\omega_u^k} w_j \geq \theta_k$, all jobs of the customer $k$create one batch

If $TB^k \neq \emptyset$and$\sum_{j\in\omega_u^k} w_j < \theta_k$, add all jobs of the customer $k$to TB^k

P13) If $d_q \geq \left(\sum_{k\in c(\omega_c\omega_o)} s_k + \sum_{j\in\omega_c\omega_o} p_j\right) + \left(\sum_{k\in c(\omega_u)} s_k + \sum_{j\in\omega_u} p_j\right)$one leaf is created;the open batch must be closed and for other jobs in ω_u one of the below state must be doing (similar to the previous):

For each $k \in c(\omega_u)$:

If $TB^k = \emptyset$, All jobs of the customer $k$create one batch

If $TB^k \neq \emptyset$and$\sum_{j\in\omega_u^k} w_j \geq \theta_k$, all jobs of the customer $k$create one batch

If $TB^k \neq \emptyset$and$\sum_{j\in\omega_u^k} w_j < \theta_k$, add all jobs of the customer $k$to TB^k

4-2- Upper bound

In this paper, the mentioned HA is considered as the upper bound for the mentioned problem.

4-3- Lower bound

- Minimum delivery costs:

The delivery costs for a partial scheduling is at least equal to the sum of the total delivery costs of theclose batches plus sum of the delivery costsfor each customer related to the rest of the jobs. If$TB^k = \emptyset$ where $k = c(\omega_o)$, the minimum delivery costs is$\sum_{k\in\omega_c} |\omega_c^k|\, \theta_k + \sum_{k\in\omega_o\omega_u\omega_e} \theta_k$ where $|\omega_c^k|$ is the number of close batch for customer k; otherwise it is equal to$\sum_{k\in\omega_c} |\omega_c^k|\, \theta_k + \sum_{k\in\omega_o\omega_u\omega_e} \theta_k + \sum_{k\in\omega_e} \theta_k$.

- Minimum sum of the weighted number of tardy jobs:

According to the mentioned B&B structure, in each node $\omega_c\omega_o$ are not delayed. The minimum sum of the weighted number of tardy jobs for ω_u can be calculated by adding up the weights of the $\acute{n}$ jobs that have the least weights among the jobs belonging to ω_u, where $\acute{n}$ is the minimum number of tardy job.$\acute{n}$can be obtained from this set and can be calculated using Moore's algorithmby considering completion time of ω_o as machine availability time. We show this value by$\sum_{j=1}^{\acute{n}} w_{[j]}^u$ in which $w_{[j]}^u$ is the j^{th} weight when we sort the weights of the jobs in ω_u in ascending order. $\omega_e$is the sets of *TB*s so the sum of the weighted number of tardy jobs for ω_e is therefore $\sum_{j\in\omega_e} w_j$.

Hence, the lower bound (*LB*) can be obtained in the following way:

$$LB = \begin{cases} \sum_{k\in\omega_c} |\omega_c^k|\, \theta_k + \sum_{k\in\omega_o\omega_u\omega_e} \theta_k + \sum_{j=1}^{\acute{n}} w_{[j]}^u + \sum_{j\in\omega_e} w_j \; TB^k = \emptyset \\ \sum_{k\in\omega_c} |\omega_c^k|\, \theta_k + \sum_{k\in\omega_o\omega_u\omega_e} \theta_k + \sum_{k\in\omega_e} \theta_k + \sum_{j=1}^{\acute{n}} w_{[j]}^u + \sum_{j\in\omega_e} w_j \; TB^k \neq \emptyset \end{cases}$$

in which $k = c(\omega_o)$.

5- Numerical example

To illustrate the proposed B&B method, we apply it to a simple numerical example. The basic input data with $\theta = (4, 2)$ and $s = (1, 2)$ are shown in table 1.

Table 1. Numerical example data

J	*1*	*2*	*3*	*4*	*5*
k	2	1	1	2	2
p_j	2	2	3	4	1
d_j	5	7	10	11	12
w_j	5	2	1	5	3

The result of the B&B method for this example is shown in the Figure 1by the first best method for searching the tree. For representation of the close batch the sign "//" and for the open batch the sign "_" is used. The bold numbers at the end of sequence are the jobs that go to the end as *TB*. Applying the mentioned HA in section 3, the $/1/45/23/$solution obtained for the upper bound, therefore the upper bound is$UB = 2 + 2 + 4 + 2 + 1 = 11$. Thus, the initial upper bound is optimal.

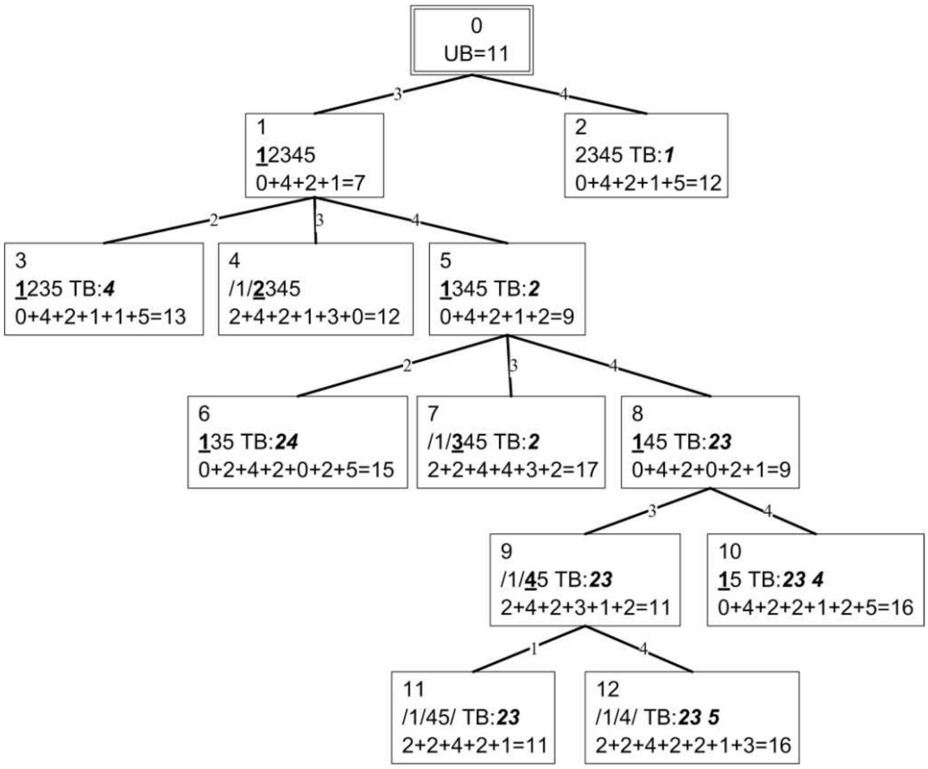

Fig 1. B&B tree for an example

6- Computational results

There are no known benchmark test problem instances for evaluating the HA and the B&B and comparing their performance with the dynamic programming algorithm (DP) of Steiner and Zhang (2009). In this paper, the instances are generated as follows. n is set to $5, 10, \dots, 25$. The numbers of customers for each job number are shown in Tables2 and 3. The processing times, the batch setup times, the due dates and the job weightswere randomly generated integers from the uniform distribution defined on[1 100], $[0\ 0.4\bar{p}]$,$[0\ (\sum s_k j_k + \sum p_j)]$, in which j_k is the number of jobs belongs to the customer k in each instance, and[1 100] respectively. Based on the batch delivery costs values, we generatedtwo classesof the problems, namely A and B, for each given the number of job. For class A and class B the intervals that the delivery costswere generated randomly are $[0\ \bar{w}]$ and $[0\ 20\bar{w}]$ respectively. For any combinations of n and K, 20 instances are constructed randomly. Hence we generate and solve totally 600 (2*15*20) instances. The mentioned B&B with first-best method for searching the tree were coded using Matlab, and were run on a computer with a 2.81GHz CPU and a 512M RAM.

For solving each problem, a time constraint equal to1000 seconds is considered and if the problem cannot get a solution regarding this constraint, then we do not use the procedure for that problem. The computational results for the classes A and B are shown in the Tables 2 and 3 respectively. Stars in these tables indicate that the corresponding method could not solve instances in mentioned time constraint.

In each class, the DP has solved 6.67% of all generated problems in time constraint that we consider while our B&B has solved 91.33% and 94.33% of them in the class A and the class B respectively. These results show that our B&B is more efficient than the DP of Steiner and Zhang (2009).

The results show that as the number of customers for each job number increases the number of instances solved optimally by the HA and the B&B increased. The results also show that, as the number of customers for each job number increases the run time of the B&B and the average relative error of the HA and the maximum of it decreased. These results show that for the same number of jobs, the problems with a larger number of customers are easier. When the number of job is same as the customer number, i.e. each customer hasonly one job, the run time of the B&B is minimum. In this situation, the total delivery costs are constant and the objective is minimizing the total weighted number of tardy jobs.Based on the average run time of the B&Bwe cannot getan accurate result about the hardness of the classes A and B respect to each other.

The HA has solved 31.67% and 25% of the instances in theclass A andthe class B optimally respectively. The maximum average relative error of the HA is 146.3%. These results show that we need a better heuristic algorithm. The average run time has increased with increasingthejob number and the maximum average run time is 0.034 seconds.

Table 2. Computational results for class A

n	K	No. of optimum instances			Avg. of CPU time (s)			$\frac{HA-Opt.}{Opt.}(\%)$	
		HA	B&B	DP	HA	B&B	DP	Avg.	max
5	5	19	20	20	0.002	0.005	4.897	0.37	7.38
10	5	5	20	*	0.007	0.077	*	21.60	89.30
	10	16	20	*	0.007	0.013	*	0.72	6.44
15	5	0	20	*	0.006	1.128	*	40.49	104.68
	10	5	20	*	0.011	0.252	*	18.52	73.46
	15	16	20	*	0.012	0.027	*	0.61	8.75
20	5	0	18	*	0.010	*	*	*	139.18
	10	0	20	*	0.016	7.280	*	36.78	86.22
	15	3	20	*	0.017	0.335	*	12.71	40.74
	20	16	20	*	0.022	0.063	*	0.24	1.78
25	5	0	2	*	0.013	*	*	*	146.30
	10	0	14	*	0.016	*	*	*	72.46
	15	1	20	*	0.024	39.146	*	21.39	61.71
	20	2	20	*	0.028	4.501	*	11.39	57.11
	25	12	20	*	0.031	0.139	*	0.58	4.58

Table 3. Computational results for class B

n	K	No. of optimum instances			Avg. of CPU time (s)			$\frac{HA-Opt.}{Opt.}$(%)	
		HA	B&B	DP	HA	B&B	DP	Avg.	max
5	5	20	20	20	0.004	0.006	3.513	0.00	0.00
10	5	0	20	*	0.002	0.066	*	37.60	70.38
	10	13	20	*	0.005	0.022	*	0.11	0.82
15	5	0	20	*	0.007	1.063	*	45.32	73.66
	10	1	20	*	0.012	0.266	*	19.74	48.08
	15	14	20	*	0.013	0.038	*	0.08	0.59
20	5	0	18	*	0.013	*	*	*	82.17
	10	0	20	*	0.015	9.453	*	28.12	49.00
	15	0	20	*	0.018	0.912	*	13.81	29.79
	20	12	20	*	0.022	0.083	*	0.06	0.57
25	5	0	9	*	0.011	*	*	*	57.13
	10	0	17	*	0.016	*	*	*	65.44
	15	0	19	*	0.024	*	*	*	36.45
	20	0	20	*	0.030	4.552	*	10.26	18.62
	25	15	20	*	0.034	0.086	*	0.03	0.32

7- Conclusion

In this paper, we presented a heuristic algorithm and a new branch and bound method for minimizing the sum of the total weighed number of tardy jobs and batch delivery costs in a two-level supply chain for restricted case, in which the tardy jobs are delivered in a separate late batch to every customer. Our method compared with an existing dynamic programming algorithm by computational tests. Computational testsare used to study the efficiency of the B&B algorithm and the results showed that the B&B is more efficient than the DP of Steiner and Zhang (2009) by far.

Using meta-heuristics or other methods for solving the general case of the mentioned problem can be studied in the future because the restricted case may produce no-optimal solution for the general case. In addition other production-scheduling objectives such as total weighted of tardiness can be added to the objective function considered in this paper or instead of it (for similar new work one can refer to Zegordi et al. (2010) and Hamidinia et al. (2012)). Also one can integrates the same problem with order acceptance issue in both tactical and technical levels in supply chain (for example see Ramayah et al. (2007), Renna (2009),Renna (2012), Kalantari et al. (2011), Yin et al. (2010), Reisi-Nafchi and Moslehi (2015)). Also with due date assignment one can refer to Steiner and Zhang (2011) or Rasti-Barzoki and Hejazi (2013). For resource allocation one can refer to Rasti-Barzoki and Hejazi (2015).

References

Brucker P. and Kovalyov M.Y., 1996. Single Machine Batch Scheduling to Minimize The Weighted Number of Late Jobs, Mathematical Methods of Operation Research 43, 1-8.

Chen Z-L., 2010. Integrated Production and Outbound Distribution Scheduling: Review and Extensions. Operations Research 58 (1), 130-148.

Gens G.V. and Levner E.V., 1979. Discrete optimization problems and efficient approximate algorithms, Engineering Cybernetics 17 (6), 1-11.

Gens G.V. and Levner E.V., 1981. Fast Approximation Algorithm for Job Sequencing With Deadlines. Discrete Applied Mathematics 3 (4), 313-318.

Hall N.G., Potts C.N., 2003. Supply Chain Scheduling: Batching And Delivery. Operations Research 51 (4), 566-584.

Hallah R.M., Bulfin R.L., 2003. Minimizing the Weighted Number of Tardy Jobs on a Single Machine, European Journal of Operational Research 145, 45-56.

Hallah R.M., Bulfin R.L., 2007. Minimizing the Weighted Number of Tardy Jobs on a single machine with release dates, European Journal of Operational Research 176, 727-744.

Hamidinia A., Khakabi mamaghani S., Mahdavi Mazdeh M., Jafari M., 2012, A genetic algorithm for minimizing total tardiness/earliness of weighted jobs in a batched delivery system, Computers & Industrial Engineering 62, 29–38

Hochbaum D.S., Landy D., 1994. Scheduling with batching: minimizing the weighted number of tardy jobs. Operations Research Letters, 16:79-86.

Ji M., He Y., Cheng T.C.E., 2007, Batch delivery scheduling with batch delivery cost on a single machine, European Journal of Operational Research 176 745–755.

Kalantari, M., Rabbani, M. and Ebadian, M. (2011) 'A decision support system for order acceptance/rejection in hybrid MTS/MTO production systems', Applied MathematicalModelling, Vol. 35, No. 3, pp.1363–1377.

Karp, R. M., 1972. Reducibility among Combinatorial Problems. R. E. Miller, J. W. Thatcher, Eds. Complexity of Computer Computations. Plenum Press, New York, 85-103.

Mazdeh M.M., Sarhadi M., Ashouri A., Hindi K.S., 2011. Single-Machine Batch Scheduling Minimizing Weighted Flow Times and Delivery costs. Applied Mathematical Modelling 35, 563-570.

Mazdeh M.M., Sarhadi M., Hindi K.S., 2007. A Branch-And-Bound Algorithm For Single-Machine Scheduling With Batch Delivery Minimizing Flow Times And Delivery costs, European Journal of Operational Research 183, 74-86.

Mazdeh M.M., Sarhadi M., Hindi K.S., 2008. A Branch-And-Bound Algorithm For Single-Machine Scheduling With Batch Delivery And Job Release Times, Computers and Operations Research 35, 1099-1111.

Moore J. M., 1968. An n job, one machine sequencing algorithm for minimizing the number of late jobs, Management Science 15, 102-109.

Rasti-Barzoki, M., & Hejazi, S. R. (2013). Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries for multiple customers in supply chains. European Journal of Operational Research, 228, 345-357.

Rasti-Barzoki, M., & Hejazi, S. R. (2015). Pseudo-polynomial dynamic programming for an integrated due date assignment, resource allocation, production, and distribution scheduling model in supply chain scheduling. Applied Mathematical Modelling.

Rasti-Barzoki, M., Hejazi, S. R., &Mazdeh, M. M. (2013). A Branch and Bound Algorithm to Minimize the Total Weighed Number of Tardy Jobs and Delivery Costs. Applied Mathematical Modelling, 37, 4924-4937.

Ramayah, T., Roy, M.H., Li, K.B., Jantan, M., Zbib, I. and Ahmed, Z.U. (2007) 'Type of procurement and operational performance: comparing e-procurement and offline purchasing'. International Journal of Services and Operations Management, Vol. 3, No. 3, pp.279–296.

Reisi–Nafchi, M., & Moslehi, G. (2015). Integrating two–agent scheduling and order acceptance problems to maximise total revenue by bounding each agent penalty function. International Journal of Services and Operations Management, 20, 358-384.

Renna, P. (2009) 'A multi-agent system architecture for business-to-business applications'. International Journal of Services and Operations Management, Vol. 5, No. 3, pp.375–401.

Renna, P. (2012) ‘Simulation-based tool to analyse the effect oforder acceptance policy in a make-to-order manufacturing system’, International Journal of Services and Operations Management, Vol. 11, No. 1, pp.70–86.

Sahni S.K., 1976. Algorithms for Scheduling Independent Tasks. Journal of the ACM 23 (1), 116-127.

Steiner G., Zhang R., 2007. Minimizing the Weighted Number of Late Jobs with Batch Setup Times and Delivery costs on a Single Machine , Multiprocessor Scheduling: Theory and Applications, Book edited by Eugene Levner, Itech Education and Publishing, Vienna, Austria.

Steiner G., Zhang R., 2009. Approximation algorithms for minimizing the total weighted number of late jobs with late deliveries in two-level supply chains, journal of scheduling 12, 565–574.

Yin, Y., Cheng, S.R., Cheng, T.C.E., Wu, W.H.and Wu, C.C. (2010) ‘Two-agent single-machinescheduling with release times and deadlines’. International Journal of Shipping and Transport Logistics, Vol. 5, No. 1, pp.75–94.

ZegordiS.H. Kamal AbadiI.N., Beheshti Nia, M.A.,2010. A novel genetic algorithm for solving production and transportation schedulingin a two-stage supply chain, Computers & Industrial Engineering 58, 373-381.