Minimizing total weighted tardiness and delivery costs with batch delivery capacity constraints

Mohammad Mahdavi Mazdeh[a], Amir Noroozi[b],

[a,b] Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran
[a] Tel: +98 21 73225002, Email: mazdeh@ iust.ac.ir
[b] Tel: +98 21 73225002, Email: amir_noroozie@iust.ac.ir

Abstract

In this paper, we consider coordinating the supply chain scheduling problem and batch delivery in a flow shop manufacturing system. The problem is to minimize the sum of weighted tardiness penalties and the delivery costs of the jobs, where the jobs are delivered in batches. A mixed integer programming model is proposed for the problem. In addition, a particle swarm optimization algorithm is used for solving the large-scale instances. To explore and to locate the algorithm in a better neighborhood, a Local Search is presented. To verify the developed model and evaluate the performance of algorithm against the exact solution, a commercial solver is used. Furthermore, the influence of the different parameters and assumptions of the proposed model are discussed.

Keywords:
Integrated Production-Distribution, Batch delivery, particle swarm optimization algorithm, Local Search.

Introduction

One of the important decisions both in production and distribution parties at operational level in a supply chain is scheduling. The traditional scheduling models were presented at the production stage and separately from distribution scheduling, while integrated scheduling of production and distribution may improve the performance of the whole supply chain [1, 2]. In addition, the cost of product distribution is accounted for about 30 percent of the cost of the product [3]. One of the most important activities that can influence on costs at this stage is batch delivery strategy. Batch delivery is a common characteristic of many real-life distribution centers in which orders are transported and ultimately delivered in containers such as transporters (vehicles), pallets, boxes or carts. Batch delivery leads to a significant decrease in the total transportation cost.

Emphasizing on the coordination and integration of production scheduling and batch delivery in distribution stage of a supply chain has become one of the vital strategies for the modern manufacturers to gain competitive advantages [4]. Chen [5] presented a survey of such existing models. Yin, et al. [6] addressed a batch delivery single machine scheduling in which jobs have an assignable common due window. Yin, et al. [7] considered the single machine problem with an assignable common due date and controllable processing times for minimizing a cost function based on earliness, weighted number of tardy jobs, job holding, due-date assignment, batch delivery, makespan, and resource consumption. Rasti-Barzoki and Hejazi [8] presented an Integer Programming model, a heuristic algorithm, and a branch and bound for solving single machine problem minimizing the weighted number of tardy jobs with the due date assignment, batch set up time and the capacity constrained deliveries to multiple customers in the single machine environment. In other works Rasti-Barzoki and Hejazi [9] studied an integrated due date assignment and production and batch delivery scheduling problem with controllable processing times. To minimize the sum of the weighted number of tardy jobs and the due date assignment, the resource allocation and the batch delivery costs, a pseudo-polynomial dynamic programming algorithm was presented. In the same integrated production and distribution scheduling problem, Assarzadegan and Rasti-Barzoki [10], to minimize costs associated with maximum tardiness, due date assignment and delivery in a single machine, presented a Mixed Integer Non-Linear Programming and a Mixed Integer Programming are used for the solution. Cheng and Wang [11] studied machine scheduling problems in which the jobs require set up time and belong to different job classes and they need to be delivered in batches to customers after processing. To minimize the weighted sum of the last arrival time of jobs to customers and the delivery (transportation) cost, they developed a dynamic programming algorithm to solve the problem optimally.

For minimizing the sum of weighted flow times and delivery costs in supply chain, Mazdeh, et al. [12] investigated structural properties of scheduling a set of jobs on a single machine for to one customer or to another machine and developed a branch and bound algorithm. They considered two variants of the problem: 1) there is no upper limit on the number of batches, and 2) there is an upper limit on the number of batches allowed. Mazdeh and Rostami [13] have been considered minimizing the maximum tardiness and the sum of delivery costs of a two-machine flow-shop problem in a batch delivery system. They presented a mixed integer linear programming model and a branch-and-bound algorithm. However, the flow shop scheduling problems are well known kinds of scheduling problems with many practical applications in manufacturing systems. As can be seen earlier, the most of the research on batch delivery addresses to single machines and to the best of our knowledge in batch delivery literature, there are developed models for only two batch processing machines flow shop. Therefore, in this paper, we developed the batch delivery problem to a flow shop with M machines.

In addition, the tardiness leads to the dissatisfaction of customers within the supply chain and imposes penalties on companies. The manufacturers try to dispatch products to customers with the lowest possible tardiness. On the other hand, manufactures like to decrease the sum of delivery costs by using a batch delivery system, although such a decision may lead to an increase in tardiness penalties. Therefore, the trade-off between these two factors, i.e. scheduling and delivery cost, is a crucial issue in supply chain management. However, the most of papers in batch delivery literature investigated the maximum tardiness. To address this gap, in this paper, we trade off the total weighted tardiness and delivery cost. Furthermore, in the considered problem we study a practical aspect of transportation, .i.e. the vehicles for transporting the batches has a limited capacity.

Problem statement

In the considered problem, there is one production line in the production stage with M machine. Subsequently, the completed orders are batches and shipped directly to the customer using vehicles, in the distribution stage. In transporting, the vehicle capacity is limited and considered as the maximum number of orders can be placed in the vehicle. Each order has a due date. If it is delivered later than the due date due to production scheduling, batching, and transportation, the company is obligated to pay a tardiness cost for the corresponding order. The goal is to minimize the cost resulting from transporting and tardiness cost. The following assumptions are considered.

- Problem parameters are deterministic
- The production system is a m machine production line
- There are several customers
- Orders are sent directly and in batches
- Machine breakdowns are not considered
- Preemption is not allowed

Index

N	The number of orders
M	The number of machines in flow shop
j	The index of order number
m	The index of machine number
i, t	The index of position in sequence
b	The index of batch number

Parameters list

w_j	Tardiness cost of the order j
δ	The transportation cost
$p_{j,m}$	The processing time of the order j
Cap	Vehicle loading capacity
d_j	The due date of the order j
Z	A big positive number

Decision variables list

$x_{j,t}$	1: if order j place in position t in sequence 0: otherwise
R_j	The ready time of order j
u_b	1: if an order place the batch b 0: otherwise
C_{tm}	The completion time of order in the position t on the machine m
$A_{j,b}$	1: if order j is allocated to batch b 0: otherwise
d'_b	The delivery time of order j
T_j	The tardiness of order j

Mathematical Model

$$\boldsymbol{Min} \sum_{j=1}^{J} w_j T_j + \sum_{b=1}^{N} \delta u_b \qquad (1)$$

Subject To :

$$\sum_{j=1}^{N} X_{j,t} = 1 \;; \qquad t = 1, \dots, N \qquad (2)$$

$$\sum_{t=1}^{N} X_{j,t} = 1 \;; \qquad j = 1, \dots, N \qquad (3)$$

$$C_{1m} = \sum_{i=1}^{m} \sum_{j=1}^{N} X_{j,1} P_{j,i} \;; \qquad m = 1, \dots, M \qquad (4)$$

$$C_{tm} \geq C_{tm-1} + \sum_{j=1}^{N} X_{j,t} P_{j,m} \;; \qquad m = 1, \dots, M \;, \; t = 2, \dots, N \qquad (5)$$

$$C_{tm} \geq C_{t-1m} + \sum_{j=1}^{N} X_{j,t} P_{j,m} \;; \qquad m = 1, \dots, M \;, \; t = 2, \dots, N \qquad (6)$$

$$R_j = \sum_{t=1}^{N} X_{j,t} C_{tM} \;; \qquad t = 1, \dots, N \,, j = 1, \dots, N \qquad (7)$$

$$\sum_{b=1}^{N} A_{j,b} = 1 \;; \qquad j = 1, \dots, J \qquad (8)$$

$$\sum_{j=1}^{J} A_{j,b} \leq Cap u_b \;; \qquad b = 1, \dots, J \qquad (9)$$

$$d'_b = \sum_{j=1}^{J} A_{j,b} R_j \;; \qquad j = 1, \dots, J \;, \; b = 1, \dots, N \qquad (10)$$

$$T_j \geq \textstyle\sum_{b=1}^{N} (d'_b - d_j) A_{j,b} ; \qquad j = 1, \dots, J \qquad (11)$$

$$T_j \geq 0; \qquad j = 1, \dots, J \qquad (12)$$

$$x_{j,t} \;, \; u_b \;, \; A_{j,b} \in \{0, 1\} \qquad j = 1, \dots, J \;, \; t = 1, \dots, N \;, \; b = 1, \dots, N \qquad (13)$$

$$R_j \,, T_j \,, d'_j \geq 0 \qquad j = 1, \dots, J \qquad (14)$$

In the model above, the objective function (1) minimizes total cost resulting from and transportation and tardiness costs. Constraint (2) is for assigning orders to a position in the production sequence. Constraint (3) ensures that each

position is assigned to an order. Constraint (4), (5) and (6) compute the completion time of order *j*. constraint (7) computes the ready time of the batch and constraint (8) assign orders to batches. In the batching, constraint (9) ensures that the total number of orders in the batches is smaller than the vehicles' capacity. In the batching process, a batch is ready for transport when all the orders are completed in a batch. It means that the delivery time of a batch equals to the maximum completion time of the orders in the batch. These calculations are considered in constraint (10) . After batching in distribution center, the formed batches should be directly shipped (transported) to the customers. The transportation time is considered illegible. Finally, constraints (11) and (12) calculate the tardiness of each order. Constraint (13) and (14) compute the decision variables of the problem.
Reference [13] stated that their problem is NP-hard. Due to that the model of this paper is more complex than their problem, our considered problem is NP-Hard. In following, we proposed a Particle Swarm Optimization (PSO) algorithm to solve the problem.

Particle Swarm Optimization Algorithm

In the developed PSO, each particle is a possible sequence of orders. We must note that the PSO algorithm is inherently continuous. It means that each particle represents continuous values. However, in scheduling problems, each solution is a permutation of the number of orders, i.e. a discrete value. In order to overcome this issue and the performance of PSO in discrete problems, researchers have made several propositions. These algorithms are known as discrete PSO algorithms. One of the studies that particularly investigate scheduling problems is presented by [14] and [15]. In this study, using the smallest position value (SPV), continuous values obtained by the algorithm are transformed into acceptable discrete values in the scheduling problem. This method is presented in **Figure 1**. In this figure, based on SPV, the smallest particle position is $X_{i5}^t = -1.20$. Therefore, the dimension number, i.e. 5, is selected as the order number that is used in the first particle position. The second smallest particle position is $X_{i2}^t = -0.99$. Thus, the dimension number, i.e. 2, is selected as the order number for the second particle position and so on.
After that the orders permutation is determined, two heuristic is used to form the batches.
Heuristic **H1**. The first order in the schedule is placed in batch 1 of the customer. For the second order, if the size of the order in the schedule is not larger than the remaining capacity of the batch, the order is placed in the batch. Else if the size is larger than the remaining capacity of the batch, this batch is closed. This process is implemented for next order in schedule. This process is repeated until all of orders are placed in batches.
Heuristic **H2**. The first order in the schedule is placed in batch 1. For second order, if the size of the order in the schedule is not larger than the remaining capacity of the batch, the order is placed in the batch. Else if the size is larger than the remaining capacity of the batch, the size of third order is checked and in the same way, until no order can be placed in the batch. This batch is closed and then, for un-batched orders, this process is repeated by a new batch. This process is repeated until all of orders are placed in batches.

Order number	**1**	**2**	**3**	**4**	**5**	**6**
X_{ij}^t	1.80	-0.99	3.01	-0.72	-1.20	**2.15**
The permutation of the SPV rule	5	2	4	1	6	**3**

Figure 1- The SPV method

Initial population

The initial solution is usually generated randomly, as in the proposed algorithm. Equations (15) and (16) are used to create the initial random solutions and the initial positions and velocities.

$$X_{ij}^0 = X_{\min} + (X_{\max} - X_{\min}) \times r_1 \quad (15)$$

$$V_{ij}^0 = V_{\min} + (V_{\max} - V_{\min}) \times r_2 \quad (16)$$

Where, r_1 and r_2 are random value with a uniform distribution in range (0, 1).
A significant factor on the final result's quality of a search procedure is the initial solution. It has already been recognized and emphasized by many researchers in the recent years. Hence, in order to achieve a satisfactory level of the final result's quality for such a hard combinatorial problem, precise considerations should be given to the intelligent selection of their initial procedures. In this paper, four types of initial procedures have been proposed which is based on two best known rules of scheduling literature and two simple suggestion heuristic.
In the problem, the order completion time of an order is equal to the ready time of the order for batching and transporting. Therefore, a smaller completion time may be lead to a smaller ready time, which influences directly on delivery time of the batch that the order belong it and eventually, delivery time of the orders. On the other hand, in the literature, the shortest processing time (SPT) is proposed to minimize the completion time. In addition to, the due date of the orders influences directly on scheduling of the orders and eventually, tardiness of the orders. On the other hand, in the literature, the earliest due date (EDD) rule is proposed to minimize the orders lateness (a function of due date). The rules determine a schedule of orders. After that, for generating a complete solution we use the heuristic to form the batches the orders according to the schedule.
In this paper, five initial procedures are used here; SPT-H1, SPT-H2, EDD-H1 and EDD-H2, and also random. In each iteration, the particles are updated according to the best positions of each particle (*Pbest*) and the position of the best particles is the global optimum (*Gbest*). The main part of PSO is updating the velocity and position of the particles using the following equations:

$$V_{ij}^{t+1} = \chi\left(wV_{ij}^t + C_1 r_{1j}^t \left(Gbest - X_{ij}^t\right) + C_2 r_{2j}^t \left(Pbest - X_{ij}^t\right)\right) \quad (17)$$

$$X_{ij}^{t+1} = X_{ij}^t + V_{ij}^{t+1} \quad (18)$$

In equation (17), the velocity vector of each particle is updated according to its velocity at the previous stage (V_{ij}^{t}). Moreover, r_{1j}^{t} and r_{2j}^{t} are two random numbers with a uniform distribution in range (0, 1), which are generated independently. Values C_1 and C_2 are learning coefficients (or acceleration coefficients) and control the effect of *Pbest* and *Gbest* on the search process. Besides, *w* indicates the inertia weight coefficient which controls the impact of the previous velocities on the current velocity. Considering a contraction coefficient prevent of the particle's velocity from leaving the velocity range and control the convergence velocity. In this regard, χ is the contraction coefficient and insures the convergence of the algorithm. For more details, you can refer to [16]. After updating the particles' velocity, their positions (X_{ij}^{t}) are updated using Equation (18).

Local Search

Local search performs a quick exploration around a solution to locate algorithm in a better neighborhood of the current solution. In this work, when the *Gbest* is updated, the local search is applied on the *Gbest*. If the new *Gbest* results in a better objective function, the current particle is replaced by the new *Gbest*. The pseudo code of local search is demonstrated in **Figure 2**.

```
i=1;
x=Gbest;
do{
    v=the value of i-th position is randomly regenerated;
    if ( f(v) < f(x)) then
    {
    x=v;
    i=g;
    }
      i++;
}while(i<g+1);
```

Figure 2- pseudo code of local search

The pseudo code of the PSO with heuristics and local search (PSO-H-LS) is presented in the **Figure 3**.

```
START:
    Set Parameters;
    Generate Initial Population using heuristics and random procedure;
    Do{
      Form batches by FF
      Evaluate the Objective Function
      Update Pbest;
      Update Gbest;
      Apply Local Search;
      Update velocity;
      Update position;
    }while(stopping criterion is not met);
END;
```

Figure 3- The pseudo code of the PSO-H-LS algorithm

Computational results

In order to evaluate the performance of the proposed algorithms, different sizes of test problems (small and large) are needed. Considering previous works four small dataset and five large dataset is created.

For determining the number of machines in the flow shop and number of orders in large size of problems, 5 combinations from four levels for number of machines i.e. $M = \{5, 10, 15, 20\}$ and five levels for number of orders i.e. $N = \{15, 30, 50, 75, 100\}$ are considered i.e. $N \times M = \{15 \times 5, 30 \times 10, 50 \times 10, 75 \times 15, 100 \times 20\}$. Moreover, for small problems, 4 combinations from $M = \{2, 3, 4\}$ and $N = \{4, 6, 8, 10\}$ are considered i.e. $M \times N = \{4 \times 2, 6 \times 3, 8 \times 3, 10 \times 4\}$. In this paper, the data were generated from a uniform discrete distribution

- The processing time on [1,3].
- The tardiness cost on [3,9].
- The transportation cost [5,10].

The vehicle capacity is considered to be 5 for all the test problems. Equations (19) to (21) compute the due date of the orders. Equation (19) computes the average of processing time of accepted order. Equation (20) calculates a primary duration time for processing orders in a batch and delivering the batch to a customer.

$$\bar{P} = \frac{\sum_{k=1}^{K}\sum_{j=1}^{J} P_{jk}}{(\sum_{k=1}^{K} n_k)} \quad (19)$$

$$DD_k = \lceil (n_k \times \bar{P}) \rceil \quad (20)$$

After the calculations above, the due dates of each customer are generated from following distribution.

$$d_k \sim [Floor\ of\ L \times DD_k\ , Round\ of\ H \times DD_k] \quad (21)$$

where L and H are lower and upper limits and are set to be 0.8 and 1.9, respectively.

Furthermore, for each combination of large problems, four sample problems are created (24 in total) and for more reliability, each problem is executed ten times. Therefore, we have 240 executions for each algorithm to solve the model. For the proposed algorithms, the stop criteria are as follows:

1. Reach a specified number of generations.
2. No change in the TNP in the certain number of repetitions.

All the algorithms were implemented using C# programming language (visual studio 2013) on a computer with a 2.6GHz CPU and a 256Mb RAM.

In this section, we investigate verifying the developed model, evaluating the performance of algorithms and the influence of the heuristics and local search on the performance of the PSO (PSO-H: the heuristics for initial population; PSO-LS: applying the local search; PSO-H-LS: the heuristics for initial population and applying the local search).

In order to verify the developed model and evaluate the performance of algorithms against the exact solution, the commercial solver LINGO 11 is used to solve the small instances of the problems and the outputs are presented in

13th International Conference of Industrial Engineering**Mazandaran University of Science and Technology**Mizban International Hotel, 22nd and 23rd February 2017

Table 1. The first column represents the data sets characteristics. The *Obj* and *Time* columns show the objective function and CPU time (millisecond) of the algorithm. As can be seen in Table , the LINGO could find the optimal solution for 2 first data sets and, due to the large number of decision variables and complexity of the problem, it could not reach a solution for other instances, after seven hours computational time. For five data sets, although the LINGO find the optimal solution, however, the algorithms can find the near of optimal solution in a time less than the LINGO. Furthermore, for 8×3 and 10×4 data sets, the algorithms find the optimal or near of optimal solutions and better than the LINGO in a logical time.

Table 1- Comparison of algorithms in small instances

N	LINGO		PSO		PSO-H		PSO-LS		PSO-H-LS	
	Obj	*Time**	*Obj*	*Time**	*Obj*	*Time**	*Obj*	*Time**	*Obj*	*Time**
$\mathbf{4 \times 2}$	17		17	< 100	17	< 100	17	< 100	17	< 100
$\mathbf{6 \times 3}$	29	467000	31	< 100	31	< 100	30	< 100	30	< 100
$\mathbf{8 \times 3}$	41	$> 25200000(7^h)$	38	< 1000	38	< 1000	36	< 1000	36	< 1000
$\mathbf{10 \times 4}$	68	$> 25200000(7^h)$	62	< 2000	58	< 2000	58	< 2000	57	< 2000

*** The time unit is millisecond.**

We investigate the influence of the heuristics, for generating the initial solutions, on the performance of the suggested PSO. To this purpose, we run the large instance 50×10 using the heuristics as some of the individuals in the initial population and do not using the heuristics. **Figure 4** shows the convergence behavior of the two runs against the number of iterations. The figure demonstrated that the proposed heuristics has a considerable effect on the convergence behavior of the algorithm and decreases the number of iterations for achieving to an optimal or near to optimal solution.

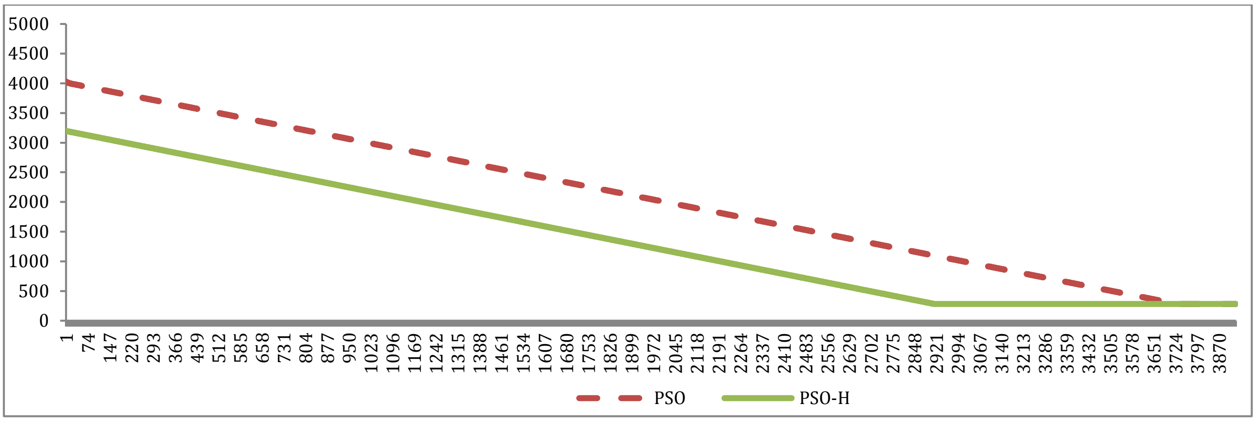

Figure 4- convergence behavior of the algorithm with and without using the heuristics in initial population

We investigate the efficiency of Local search on the performance of the suggested PSO. Because the scale of objective functions in each instance and run is different, the relative percent deviation (RPD) is calculated for the problems. RPD is a measure that is mostly used in the literature and it is defined as follows.

$$RPD = \frac{Min_{sol} - Alg_{sol}}{Min_{sol}} \qquad (22)$$

Where, Alg_{sol} is the solution of the algorithm and Min_{sol} is the minimum value of the solutions. In this measure, the lowest RPD is selected as the best algorithm.

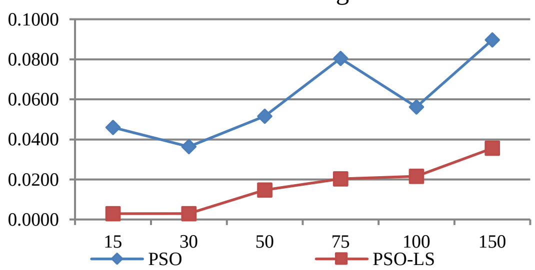

Figure 5- Interaction between algorithm performance (RPD) and size of problems

As can be seen in **Figure 5**, PSO-H-LS considerably outperform the PSO.

Conclusion

In this paper, we have studied supply the integration of chain scheduling a scheduling problem with batch delivery problem. In this supply chain, a pool of orders is received by company. The orders have different size and the capacity of vehicles is limited. We considered a delivery cost for each batch of jobs. The main contribution and conclusion of the present study are summarized as follows:

- A new Mixed Integer Programming is proposed.
- A PSO algorithm was developed for large-scale of the problem.
- A Local Search was presented to explore and to locate the algorithm in a better neighborhood.
- Random instances are generated to evaluate the performance of the algorithms.
- Using a commercial solver, the developed model was verified and the performance of algorithm against the exact solution was evaluated.

References

[1] A. Condotta, S. Knust, D. Meier, and N. V. Shakhlevich, "Tabu search and lower bounds for a combined production–transportation problem," *Computers & Operations Research*, vol. 40, pp. 886-900, 2013.

[2] M. Hajiaghaei-Keshteli and M. Aminnayeri, "Solving the integrated scheduling of production and rail

transportation problem by Keshtel algorithm," *Applied Soft Computing*, vol. 25, pp. 184-203, 2014.

[3] M. Diaby, "Successive linear approximation procedure for generalized fixed-charge transportation problems," *Journal of the Operational Research Society*, pp. 991-1001, 1991.

[4] C.-S. Su, J. C.-H. Pan, and T.-S. Hsu, "A new heuristic algorithm for the machine scheduling problem with job delivery coordination," *Theoretical Computer Science*, vol. 410, pp. 2581-2591, 2009.

[5] Z.-L. Chen, "Integrated production and outbound distribution scheduling: review and extensions," *Operations Research*, vol. 58, pp. 130-148, 2010.

[6] Y. Yin, T. Cheng, C.-J. Hsu, and C.-C. Wu, "Single-machine batch delivery scheduling with an assignable common due window," *Omega*, vol. 41, pp. 216-225, 2013.

[7] Y. Yin, T. Cheng, C.-C. Wu, and S.-R. Cheng, "Single-machine common due-date scheduling with batch delivery costs and resource-dependent processing times," *International Journal of Production Research*, vol. 51, pp. 5083-5099, 2013.

[8] M. Rasti-Barzoki and S. R. Hejazi, "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*, vol. 228, pp. 345-357, 2013.

[9] M. Rasti-Barzoki and S. R. Hejazi, "Pseudo-polynomial dynamic programming for an integrated due date assignment, resource allocation, production, and distribution scheduling model in supply chain scheduling," *Applied Mathematical Modelling*, vol. 39, pp. 3280-3289, 2015.

[10] P. Assarzadegan and M. Rasti-Barzoki, "Minimizing sum of the due date assignment costs, maximum tardiness and distribution costs in a supply chain scheduling problem," *Applied Soft Computing*, vol. 47, pp. 343-356, 2016.

[11] T. E. Cheng and X. Wang, "Machine scheduling with job class setup and delivery considerations," *Computers & Operations Research*, vol. 37, pp. 1123-1128, 2010.

[12] M. M. Mazdeh, S. Shashaani, A. Ashouri, and K. S. Hindi, "Single-machine batch scheduling minimizing weighted flow times and delivery costs," *Applied Mathematical Modelling*, vol. 35, pp. 563-570, 2011.

[13] M. M. Mazdeh and M. Rostami, "A branch-and-bound algorithm for two-machine flow-shop scheduling problems with batch delivery costs," *International Journal of Systems Science: Operations & Logistics*, vol. 1, pp. 94-104, 2014.

[14] M. F. Tasgetiren, M. Sevkli, Y.-C. Liang, and G. Gencyilmaz, "Particle swarm optimization algorithm for single machine total weighted tardiness problem," in *Evolutionary Computation, 2004. CEC2004. Congress on*, 2004, pp. 1412-1419.

[15] M. F. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem," *European Journal of Operational Research*, vol. 177, pp. 1930-1947, 2007.

[16] F. Van Den Bergh, "An analysis of particle swarm optimizers," University of Pretoria, 2006.