Incorporating batch delivery in multi-factory supply chain scheduling; Heuristic and lower bounds

N. Karimi, H. Davoudpour

Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, 424 Hafez Avenue, Tehran 15916-34311, Iran

Abstract

In this paper an integrated delivery and production scheduling problem for serial multi-factory supply chain is addressed. Consider a supply chain scheduling problem in which number of jobs should be scheduled on series of factories and delivered to downstream factories for processing and finally delivered to the customer. The number of jobs in delivery batches is constrained by the batch size. The high delivery cost in manufacturing systems is the main motivation of integration of production and delivery scheduling which contains a more holistic view of the supply chain problem. So constituting batches of jobs may reduce the transportation cost but on the other side, it may have incremental effect on the total holding cost. The objective is to minimize the sum of the total holding cost and the total transportation costs. The problem is NP-hard. We formulate the problem as a Mixed Integer Programming (MIP) model. Moreover, in this paper, a heuristic and two lower bounds are presented. The effectiveness of these methods are investigated through the computational experiments.

Keywords: Multi-factory scheduling; Supply chain; Batch delivery; Mixed integer programming; Lower bound; Holding cost; Heuristic.

1. Introduction

Because of emerging various markets all over the world, globalization of these markets become inevitable for most of industries. Moreover increasing competitiveness revealed the importance of designing a supply chain. Supply chain is an improved organization which contains a number of companies, suppliers, customers, products and services. It makes a relationship between suppliers and customers through different stages, from producing raw material by suppliers to consumption of product by customers. These companies should work in a coordinated manner in order to vouch for a reliable flow of goods, services and information. So, many of the production firms are converting to global chains which contains several factories or manufacturing sites such as supplier, production shops and also outsourcing units [1]. This means that, in order to have competitive capabilities in international economics, most of firms are altering from single-factory production to multi-factory productions. So, sharing information and coordinating planning and scheduling in manufacturing facilities of the same supply chain leads to performance enhancement, higher reliability, lower inventory and etc.

As there are number of factories along the supply chain, the scheduling activities are more complex than the traditional single-factory scheduling problems [2]. Many of the researches and industrialists are interested in this issue in recent years. [3] stated that independent manufacturing firms are transforming to dependent manufacturing factories to become capable of competitive advantages in economic environment.

Multi-factory supply chains may have different structures based on the positioning of factories in the chain. They can be categorized to three classes: parallel, serial and network structures. In Parallel structure, multiple factories which are considered to be able to produce various types of product are positioned in a parallel structure. Whenever an order enters the system, it should be allocated to one of these plants. In the other words, the order allocation to an appropriate plant to meet the due dates is the most important decisions in these problems. [4] considered the multi-site scheduling problem and proposed a modeling method based on fuzzy concepts. [5] studied scheduling strategies in heterogeneous distributed systems. In this system half of the machines have two different speeds. Probabilistic, deterministic and combined policies are considered in this study. [6] presented a genetic based heuristic for an integrated process planning and scheduling for multi-plant supply chain. The considered supply chain problem consists of parallel single-machine plants. A modified genetic algorithm is proposed for distributed scheduling by [7]. [8] presented a method for evaluating the robustness of medium term distributed schedule. In this study distributed system is considered as a virtual jobshop. [9] presented a modified genetic algorithm for distributed scheduling in multi-factory and multi-product environment in which each product has its own production route. [10] considered the problem of [9] but in this study each product has its alternative production route and plants are positioned in different geographical places. [11] considered a distributed production system of multiple factories. [12] focused on the integration of the inventory replacement, job scheduling and lead time quotation decisions in MTO-MTS supply chain. Multi-factory problem, in which the production line is flowshop, is proposed by [13]. They proposed six mixed integer linear programming (MILP) models and also proposed two simple factory assignment rules together. 14 heuristics based on dispatching rules, effective constructive heuristics and variable neighborhood descent methods were proposed for this problem's solution. An improved genetic algorithm is proposed by [14] to solve the distributed and flexible job-shop scheduling problem. Integrated planning and scheduling for multisite, multiproduct and multipurpose batch plants was presented by [15]. They applied the augmented Lagrangian method to optimize the mentioned problem. [16] studied production network scheduling problem which consists of several factories geographically distributed. They considered each factory as a production agent with parallel machines which can has its own objective function. They proposed mixed integer linear programming model and a ε-constraint approach for obtaining Pareto optimal solutions and also presented two metaheuristic algorithm for the mentioned problem, genetic algorithm and imperialist competitive algorithm. [17] studied the system in which production is taken place in multiple factories which are geographically distributed and have different speed in job’s processing. They proposed a mixed integer programming approach for minimizing the total completion time of all factories. They proposed a genetic algorithm with new coding scheme and a heuristic approach for the mentioned problem. They compared the obtained solution with the proposed lower bound in their study.

In the serial structure there are a number of plants in the system which are positioned in the series. In the system with serial structure, while the material enters the system, the first plant starts processing. After finishing the process on this plant the semi-finished product would transfer to the downstream plant and the production process would start there. If these two plants are positioned in two different geographical places, transportation constraints should be considered too. Interrelation among factories in the serial multi-factory supply chain causes the high value of complexity. This means that the effect of material shortage in the upstream factories would be extended through the supply chain and cause delay in production in the downstream factories. On the other hand, stopping the production in the downstream factories because of inventory accumulation would cause decrease or stop in production of upstream factories. Thus, Production and transportation between upstream and downstream factories should be synchronized in order to decrease inventory cost and also to avoid risk of stock out for a factory [18]. [19] applied the constraint satisfaction approach for the integrated production and transportation scheduling case study. They considered multi-site manufacturing environment in which sites are positioned in the serial order. [20] investigated a problem which considers sequencing, lot-sizing and scheduling of several products manufactured through several firms in a serial-type supply chain. They implemented a time-varying lot-sizing policy for problem formulation and also solved it by a three-phase heuristic.

Network structure is a combination of serial and parallel structure. The same as parallel structure, there is number of identical plant in each stage and the allocation of orders (jobs) to factories should be considered too. [21] studied the multi-factory scheduling problem. In this study factories are structured as network. They presented a modified genetic algorithm to minimize the completion time.

As it is clear from review of the past studies most of them are focused on the parallel structure and few numbers of studies considered serial and network structures. So there is a big gap in this field and also evident opportunity for future researches.

In this study we investigate the serial type structure of the multi-factory supply chain. In such manufacturing systems semi-finished jobs are transported from an upstream to the downstream factory for further processing by transporters. This dispatching can be taken place for an individual product, or a batch of products. In the case of dispatching batch of products, the transportation cost will reduce but it may increase the total holding cost. Thus the goal of the problem is to determine simultaneously the optimal number of batches, the assignment of jobs to batches and the batch scheduling in a way that minimizes the problem's objective function. The objective of this paper is the sum of total transportation cost and total holding cost. The total transportation cost is an increasing function of the number of batch deliveries. As the coordination of the production and delivery scheduling simultaneously can improve the overall operational performance of the supply chain, it has being recently considered by researchers.

As the coordination of the production and delivery scheduling can improve the overall operational performance of the supply chains, it has being recently considered by researchers [22], [23] [24], [25]. The problem of scheduling and batch delivery to a customer with the aim of minimizing the sum of the total weighted flow time and delivery cost on a single machine is considered by [26]. [27],[28] have also minimized the sum of the total flow time and delivery cost considering multiple customers with zero and non-zero ready time. An integrated due date assignment and single machine production and batch delivery scheduling problem for make-to-order production system is addressed by [29]. In the research of [30], number of orders would be received by manufacturer from one customer while the orders need to be processed on one or two machines and be sent to the customer in batches. Their goal was minimizing the sum of the total weighted number of tardy jobs and the delivery costs.

So, in this study we investigate the serial multi-factory scheduling problem with batch delivery minimizing the sum of total holding cost and the total transportation costs. This problem focuses on the determining the assignment of jobs to batches and the start and finish time of processing of batches. We present the problem's mathematical model and two lower bounds and a heuristic. The remainder of this paper is as follows: description of the problem is presented in the next section. Section 3 details the mathematical formulation. Section 4 discusses the lower bounds. The heuristic is introduced in section 5. Section 6 presents computational results. Finally, in Section 7, we conclude the paper.

2. Problem definition

2.1. Assumption and notations

This study investigates the multi-factory scheduling problem in a supply chain in which factories are positioned in series. Jobs are transported to downstream factories and also delivered to the final customer by batches. The batch completion time is equal to the completion time of the last job in the batch. The minimization of the sum of total holding cost of jobs to the final customer and total transportation cost is the optimization criterion of the problem. It is assumed that:

- All jobs are available at zero time.
- Job processing in each factory cannot be interrupted.
- Factories are always available, with no breakdowns or scheduled/unscheduled maintenance.
- Infinite buffer exists between factories, before the first and after the last factory.
- Setup times are negligible.
- Jobs are available for processing in a factory immediately after arriving to the factory.
- Each factory can process at most one job at a time.
- A job cannot be processed in more than one factory at the same time.
- Number of jobs in each batch is at most equal to the batch (vehicle) capacity.
- Transportation times between factories are considered.
- Jobs are available for transferring between factories immediately after completion of processing the last job in the batch in the previous factory.
- There are sufficient numbers of vehicles for transportation.
- All data are known deterministically.

- There is no limitation on the number of batches.

2.2. Complexity and NP-hardness

This section discusses the complexity of the problem. By considering single machine in each factory and neglecting the transportation time between factories, the problem can be simplified to a typical flow shop scheduling problem. As [31] proved that a flow shop scheduling problem is NP-hard, the mentioned problem is NP-hard too.

3. Lower bounds

As this is an NP-hard problem, finding optimal solutions for large size of problems are so time and effort consuming. But finding a limit for solution value helps us to judge the value of different method's solutions. The lower bound would provide us this limit for solution area of mentioned problem. On the other hand, it will be calculated very quickly. Thus, in this paper two simple but effective lower bounds are presented.

The objective of this paper is sum of transportation cost and total holding cost. As mentioned above, total flow time can reflects the total holding cost truly. Thus a lower bound for this problem can be defined by sum of lower bounds on the total flow time (LBF) and the lower bound on the transportation cost (LBD):

$$LB = LBF + LBD$$

Proposition 3.1. The lower bound on the total flow time is equal to total flow time obtained in the schedule formed considering each batch as a single-job one.

Proof. If the batch contains more than one job, completed jobs of a batch should remain in the factory until finishing processing of uncompleted jobs of the same batch. Thus, this would results in the increment in the value of total flow time. Thus this effect can be prevented by forming each batch by a single job.□

Thus from the preposition above we calculate the lower bound for the flow time of the problem which has only one job in each batch through the following propositions (Proposition 4.2 and Proposition 4.3).

Proposition 3.2. Inspiring from [32], lower bound in the total flow time of our presented problem, is as below:

E_{jf} : Earliest start time of job j in factory f

$E_{[j]f}$: The $j\,th$ smallest value of the E_{jf}

F_{jf} : Earliest finish time of job j in factory f

$F_{[j]f}$: The $j\,th$ smallest value of the F_{jf}

$$t_{jkl} : \begin{cases} \sum_{m=k}^{l} p_{jm} + \sum_{m=k}^{l} \tau^{m} & for\ 1 \le j \le n \text{ and } 1 \le k \le l \le F \\ 0 & \text{otherwise} \end{cases}$$

$t_{[j]kl}$: The $j\,th$ smallest value of the t_{jkl}

l_f : The earliest completion time of jobs in factory f

$$E_{[1]1} = 0$$

$$E_{[j]1} = \sum_{i=1}^{j-1} p_{[i]1} \qquad 1 \le j \le n$$

$$F_{[j]1} = E_{[j]1} + p_{[j]1} \qquad 1 \le j \le n$$
$$E_{[1]f} = F_{[1]f-1} + \tau^{f-1} \qquad 1 \le f \le F$$
$$F_{[1]f} = E_{[1]f} + p_{[1]f} \qquad 1 \le f \le F$$
$$E_{[j]f} = \max(F_{[j]f-1} + \tau^{f-1}, F_{[j-1]f}) \qquad 2 \le j \le n,\ 2 \le f \le F$$
$$l_f = \sum_{j=1}^{n} (E_{jf} + t_{jfF}) \qquad 1 \le f \le F$$
$$LBF_1 = \max_f \{l_f\}$$

Proof: Simply it can be understood that E_{jf} and F_{jf} are lower bounds for the start time and completion time of job j on factory f respectively. Thus the value of l_f is a lower bound for the flow time of every sequence of jobs.□

Proposition 3.3. The lower bound in the total flow time in the single-job batch scheduling problem can be calculated as follows:

$$LBF_2 = \sum_{i=1}^{n} \max_{1 \le f \le F} \{ \min_{j=1}^{n} \sum_{m=1}^{f-1} p_{jm} + D_f(i) + \min_{j=1}^{n} \sum_{m=f+1}^{F} p_{jm} + \sum_{m=1}^{f} \tau^m \}$$

where $D_f(i)$ is the sum of processing time of i jobs in factory f which are sorted in the ascending order of their processing time.

Proof. If we relax the constraint stating "each factory can process at most one job at a time", for all factories but one (factory f), then a relaxation of completion time can be obtained by setting for job j a release date $\sum_{m=1}^{f-1} p_{jm}$ and a delivery time $\sum_{m=f+1}^{F} p_{jm}$. The resulting relaxation is a single job processing factory scheduling problem with release dates (heads) and delivery times (tails) for jobs. A first relaxation of this problem can be achieved by considering release dates and delivery times of all jobs to $\min_{j=1}^{n} \sum_{m=1}^{f-1} p_{jm}$ and $\min_{j=1}^{n} \sum_{m=f+1}^{F} p_{jm}$, respectively.

This yields for each factory *m* a lower bound equal to $\min_{j=1}^{n} \sum_{m=1}^{f-1} p_{jm} + \sum_{j=1}^{i} p_{jf} + \min_{j=1}^{n} \sum_{m=f+1}^{F} p_{jm} + \sum_{m=1}^{f} \tau^m$. Hence, a valid lower bound for completion time is $\max_{1 \le f \le F} \{ \min_{j=1}^{n} \sum_{m=1}^{f-1} p_{jm} + \sum_{j=1}^{i} p_{jf} + \min_{j=1}^{n} \sum_{m=f+1}^{F} p_{jm} + \sum_{m=1}^{f} \tau^m \}$. But since the sequence of jobs are not known, the term $\sum_{j=1}^{i} p_{jf}$ is not less than $D_f(i)$. So

$$\max_{1 \le f \le F} \{ \min_{j=1}^{n} \sum_{m=1}^{f-1} p_{jm} + D_f(i) + \min_{j=1}^{n} \sum_{m=f+1}^{F} p_{jm} + \sum_{m=1}^{f} \tau^m \} \le \max_{1 \le f \le F} \{ \min_{j=1}^{n} \sum_{m=1}^{f-1} p_{jm} + \sum_{j=1}^{i} p_{jf} + \min_{j=1}^{n} \sum_{m=f+1}^{F} p_{jm} + \sum_{m=1}^{f} \tau^m \}$$

Thus

$$C_i^F \ge \max_{1 \le f \le F} \{ \min_{j=1}^{n} \sum_{m=1}^{f-1} p_{jm} + D_f(i) + \min_{j=1}^{n} \sum_{m=f+1}^{F} p_{jm} + \sum_{m=1}^{f} \tau^m \}$$

As the flow time can be estimated by $TF = \sum_{i=1}^{n} C_i$, a lower bound for the total flow time of the problem is:

$$LBF_2 = \sum_{i=1}^{n} \max_{1 \le f \le F} \{ \min_{j=1}^{n} \sum_{m=1}^{f-1} p_{jm} + D_f(i) + \min_{j=1}^{n} \sum_{m=f+1}^{F} p_{jm} + \sum_{m=1}^{f} \tau^m \}. \square$$

Proposition 3.4. The lower bound in the delivery cost would be achieved by the schedule formed considering each batch as a full batch. Thus the number of batches is at least equal to:

$$LBD = \left\lceil \frac{n}{B} \right\rceil;$$

Proof. Considering each batch with number of jobs less than the batch size certainly causes a lower flow time cost but on the other hand would increase the cost of delivery. Thus filling the batch with the number of jobs equal to the batch size would avoid this cost (Let $\lceil x \rceil$ the smallest integer value equal to or greater than x).$\square$

4. Heuristic

A good strategy to expedite finding solutions and explore specific characteristic of the problem is utilizing heuristics, which needs limited computational time and effort. Heuristics can be used instead of exact methods for larger size of problem. Since our considered problem is in the NP-hard category, there is no insurance to find an optimal solution for every problem's instances via a heuristic but it would help us to find a near optimal solution in a reasonable time.

Our proposed heuristic, tackles with some decisions: (I) determination of the number of batches, (II) assignment of jobs to batches and (III) scheduling of batches. The procedure of this heuristic is as follows in which the local search procedure is adapted from [33]:

Step1. *Sorting*: according to ascending order of t_{jfe} for each e and f ($1 \le e \le f \le F$), a sequence of jobs is considered. Among these generated schedules, the one with minimum value of objective function is selected as the initial solution of the heuristic.

Step2. *Determination of the initial batches*: Assign predetermined number of sorted job to each batch (It means that each batch, except the last batch, contains B (equal to the vehicle capacity) jobs.). Consider it as the best solution (S_b).

Step3. *Batching*:

for $L = 1$ (the first job in the first batch) to $L = n$ (the final job in the final batch) do:

Separate job L from its batch

If $L = 1$

Constitute a single-job batch

If its function value is better than the best solution

replace the best solution with it

```
                Perform local search
            Else continue with the best solution

        Else
            Check two following policies and generate two new solutions:
                1. If the size of the last batch is less than B, add it to the last batch(S_1)
                2.Constitute a single-job batch (S_2)
            Choose the better solution among S_1, S_2 according to the function value (S_c)
            If the function value of S_c is better than S_b replace it (S_b = S_c)
                Perform local search
            Else continue with the best solution
    End for
```

Figure1. The heuristic pseudo code

```
O = 0
While (O < n×(n−1))
    k = 0
    While k < 2
        If k = 0
            U_1 = rand(1,n)
            U_2 = rand(1,n)
            S'=swap(U_1,U_2)
        End
        If the function value of S' is better than S_b replace it
        If k = 0
            U_1 = rand(1,n)
            U_2 = rand(1,n)
            S'=inversion(U_1,U_2)
        End
        If the function value of S' is better than S_b replace it
            k = 0
        Else
            k = k + 1
        End
    End while
O= O+1
End while
```

Figure2. The Local search pseudo code

In order to clarify the performance of the heuristic, we detail its steps through a simple example below:

Consider problem with 5 jobs to be processed where the batch size is equal to 2 and the *sorting* of jobs according to their t_{jkl} results is: 2,1,3,5,4 (The cost values in each step are assumptive to better describe the procedure).

Initial batching of this problem is as bellows (each box show a batch):

2,1	3,5	4	Cost=15

We consider it as the best solution (S_b).

The *batching* procedure is as follows:

We separate the first job from the first batch and investigate the result:

2	1	3,5	4	Cost=14

As it has the better cost value than S_b, it would replace S_b.

In the next step two following conditions should be considered:

1:	2	1,3	5	4	Cost=13

2:	2	1	3	5	4	Cost=16

The first policy creates the better solution which is better than S_b. Thus, it would replace S_b.

Now the local search would be applied on the best solution; if two random number 2,4 are generated for this solution, swap operator of local search interchanges two jobs which are in the positions 2 and 4, it can be shown as:

2	1,3	5	4	Cost=13
2	5,3	1	4	Cost=18

As the solution from swap operator is not better than S_b, we would continue with the current best solution. If the inversion operator of local search on this solution generate two random numbers 2,5, the following solution would be created:

[2] [4,5] [3] [1] Cost=12

If the solution created through the local search is better than S_b, it would replace this solution.

The procedure would be continued until all jobs are considered.

5. Computational results

Since there is lack of benchmark instances in this field, the performance of the presented methods (lower bounds and heuristic) are investigated by comparison through random generated data. Lower bounds and the heuristic procedure were coded in Matlab software. A number of computational experiments were carried out on a PC with Intel Core 2 Duo and 2 GB of RAM memory.

Test data are generated based on various sizes of factors in terms of the number of jobs and number of factories. Processing times and transportation times of jobs are uniformly distributed in [1,99],[1,200] respectively. Delivery cost and holding cost of each batch are assumed to be equal to 10 and batch size equal to 2.

5.1. Performance measures

In order to evaluate the performance of the presented methods, firstly their validity is checked through a comparison with the solution of the MIP as an exact solution for small size of the problem. Then their effectiveness is investigated for larger size of the problem.
In order to compare the values which are obtained, the *GAP* performance measure is utilized which has a little different in definition for small and large size of the problem.

For small size of the problem:

$$GAP\_s = \frac{M\_s - MIP\_s}{MIP\_s}$$

where M_s is a solution obtained from each method (heuristic or lower bounds) and MIP_s is the solution obtained from mixed integer programming approach. It is obvious that MIP solution is the best solution for the considered instance. Thus the lower the value of the *GAP_s*, the more preferable the solution is.

5.2. Comparison for small-size problem

For performing comparison in this case, test data are generated for two levels of jobs: 3 and 6, and also two level of factories: 2 and 4. Combination of these values generates 4 instances which are shown in Table 1. In this table the *GAP_s* values of *LB1*, *LB2* and *H* (which are corresponding to $LBF_1 + LBD$, $LBF_2 + LBD$ and heuristic respectively) and heuristic are presented which show the solution's deviation from the exact solution. It is evident from these results that increasing the size of the instances, the outperformance of the *LB2* than *LB1* increases. The *GAP_s* values prove the good efficiency of the proposed heuristic.

Table 1. The comparison among MIP, lower bounds and heuristic for small size of problem

n	f	LB1	LB2	H
3	2	0.001	0.001	0
3	4	0.114	0.026	0

6	2	0.076	0.016	0
6	4	0.113	0.050	0

5.3. Comparison for large-size problem

In this section the efficiency of the proposed lower bounds and heuristic are investigated for larger size of the problem. For this reason the comparisons are performed with a well-known algorithm in the field of scheduling, Genetic Algorithm (GA).

The *GAP_l* value can be redefined as:

$$GAP\_l = \frac{M\_s - GA}{GA\_s}$$

where GA_s is the solution obtained from GA approach. The tests are performed for different instances generated from different combination of $n = 20, 50, 80$ and $f = 3, 5, 10$.

Table 2. The comparison between lower bounds, heuristic with GA for large size of problem

n	f	Instance	LB1	LB2	Heuristic
20	3	1	0.123	0.098	0
		2	0.144	0.045	0
		3	0.120	0.072	0
			0.129	**0.071**	**0**
	5	1	0.166	0.075	0
		2	0.162	0.108	0
		3	0.140	0.079	0
			0.156	**0.087**	**0**
	10	1	0.130	0.136	0
		2	0.157	0.107	0
		3	0.170	0.097	0
			0.152	**0.113**	**0**
50	3	1	0.119	0.099	0
		2	0.143	0.084	0
		3	0.145	0.148	0
			0.135	**0.110**	**0**
	5	1	0.172	0.105	0.0008
		2	0.211	0.160	0.001
		3	0.164	0.110	0.0006
			0.182	**0.125**	**0.0008**
	10	1	0.208	0.176	0.002
		2	0.221	0.180	0.005
		3	0.204	0.162	0.003
			0.211	**0.172**	**0.003**
80	3	1	0.126	0.088	0.004
		2	0.138	0.113	0.008
		3	0.122	0.126	0.003
			0.128	**0.109**	**0.005**
	5	1	0.209	0.147	0.009
		2	0.186	0.145	0.012
		3	0.153	0.133	0.008
			0.182	**0.141**	**0.009**
	10	1	0.227	0.170	0.02
		2	0.212	0.194	0.007
		3	0.237	0.181	0.011
			0.225	**0.181**	**0.012**

The bold digits are the average *GAP_l* values of each combination. Results demonstrate that all methods have the similar trend relative to the variability of factors. Statistical validity of the results of this Table is verified through the analysis of the factors in the following subsections.

5.3.1. Statistical analysis of factors

In order to find the significant factors and the also important interaction among these factors, the Analysis of Variance (ANOVA) can be applied. A single factor ANOVA is performed to check whether presented methods (lower bounds and heuristic) have significant difference according to the

GAP_l performance measure. The results are presented in Table 3. As the *P-value* is very small, it can be concluded that each of these methods has significant effect on the problem's solutions.

Table3: ANOVA result for methods

Source	DF[a]	SS[b]	MS[c]	F-test	F0	P-Value
Method	2	0.3874	0.1937	197.78	3.102	0
Error	78	0.0763	0.0009			
Total	80	0.4638				

[a] DF: Degree of freedom, [b] SS: Sum of square, [c] MS: Mean square

Another analytical procedure is done for the effect of the variability of lower bounds and the number of jobs on the solution quality, using two ways ANOVA. Table 4 summarizes the results of this analysis.

Table4: ANOVA for methods versus number of job

Source	DF	SS	MS	F	F0	P
Method(A)	2	0.3874	0.1937	253.65	3.11	0
Job(B)	2	0.0154	0.0077	10.10	3.11	0
Interaction(A×B)	4	0.0059	0.0014	1.96	2.498	0.11
Error	72	0.0549	0.0007			
Total	80	0.4638				

It is clear from the results that the effect of methods and number of jobs are significant, but the interaction between these two factors is not influential.

The plot of relation between number of jobs and *GAP_l* values according to the methods is presented in figure3. From this figure, as the number of jobs increases, efficiency of each lower bound decreases.

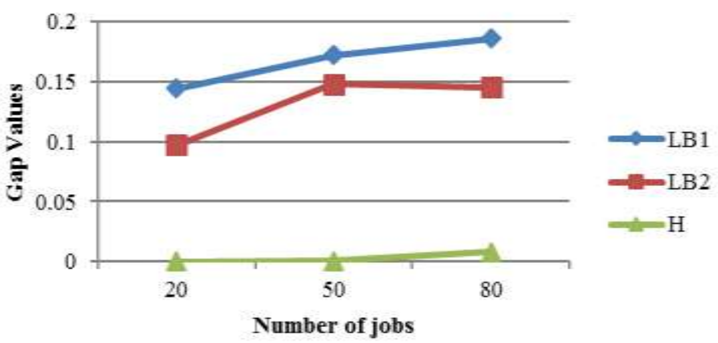

Figure3. The relation between number of jobs and Gap values of methods

The effect of magnitude of factories on the performance of the methods is investigated through the two ways ANOVA in Table5. As it can be seen from the table, lower bounds and number of factories have significant effect on the performance measure, but the effect of the interaction among them is not considerable.

Table5: ANOVA for method versus number of factory

Source	DF	SS	MS	F	F0	P
Method(A)	2	0.3874	0.1937	343.33	3.11	0
Factory (C)	2	0.0244	0.0122	21.67	3.11	0
Interaction(A×C)	4	0.0113	0.0028	5.02	2.498	0.001
Error	72	0.0406	0.0005			
Total	80	0.4638				

Figure 4 depicts this analysis of relation between variability of number of factories and lower bounds.

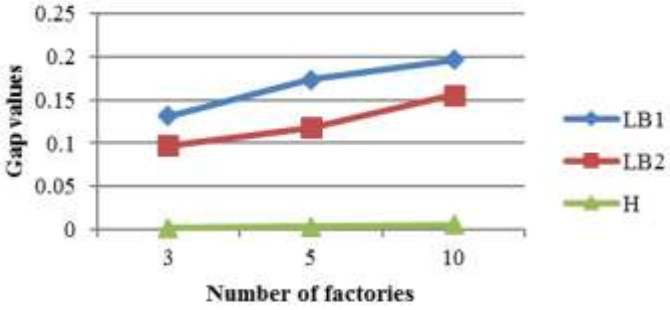

Figure4. The relation between number of factories and Gap values of methods

6. Conclusion

In this paper, we studied a scheduling model that simultaneously considers the production scheduling and product delivery. The problem of minimizing the sum of total holding cost and batch delivery costs in a supply chain was modeled by a mixed integer programming. In short, the proposed formulation simultaneously determines the number of batches, the jobs assignment to each batch, the start and finish time of batch's processing and the delivery time of jobs to customer. A heuristic and two lower bounds were presented for this problem. Although, the MIP cannot provide the solution for real size problem, its solution was used as a benchmark for a comparison with the heuristic and lower bounds in small size of the problem. The results showed the good efficiency of this methods specially the heuristic. Then a well-known metaheuristic in the field of scheduling (GA) is applied for investigating the performance of methods for larger size of the problem.

For further research, finding batching strategy for this problem is highly requested. The application of different approaches which are able to find optimal or near optimal solutions may be promising alternatives. Hence considering other objectives for this problem and also further assumptions such as limited buffer capacity, set up times and job ready times are also of interest.

References

1. Moon C, Seo Y, Yun Y, Gen M (2006) Adaptive genetic algorithm for advanced planning in manufacturing supply chain. J Intell Manuf 17:509–522. doi: 10.1007/s10845-005-0010-0

2. Moon C, Seo Y (2005) Evolutionary algorithm for advanced process planning and scheduling in a multi-plant. Comput Ind Eng 48:311–325. doi: 10.1016/j.cie.2005.01.016

3. Alvarez E (2007) Multi-plant production scheduling in SMEs. Robot Comput Integr Manuf 23:608–613. doi: 10.1016/j.rcim.2007.02.006

4. Sauer J, Suelmann G, Appelrath H (1998) Multi-site Scheduling with Fuzzy Concepts. Int J Approx Reason 19:145–160.

5. Karatza HD (2001) Job scheduling in heterogeneous distributed systems. J Syst Softw 56:203–212. doi: 10.1016/S0164-1212(00)00098-4

6. Moon C, Kim J, Hur S (2002) Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain. Comput Ind Eng 43:331–349. doi: 10.1016/S0360-8352(02)00078-5

7. Jia HZ, Nee AYC, Fuh JYH, Zhang YF (2003) A modified genetic algorithm for distributed scheduling problems. J Intell Manuf 14:351–362. doi: 10.1023/A:1024653810491

8. Archimède B, Charbonnaud P, Mercier N (2003) Robustness evaluation of multisite distributed schedule with perturbed virtual jobshops. Prod Plan Control 14:55–67. doi: 10.1080/0953728021000080988

9. Chan FTS, Chung SH, Chan PL. (2005) An adaptive genetic algorithm with dominated genes for distributed scheduling problems. Expert Syst Appl 29:364–371. doi: 10.1016/j.eswa.2005.04.009

10. Chan FTS, Chung SH, Chan PLY (2006) Application of genetic algorithms with dominant genes in a distributed scheduling problem in flexible manufacturing systems. Int J Prod Res 44:523–543. doi: 10.1080/00207540500319229

11. Jia HZ, Fuh JYH, Nee AYC, Zhang YF (2007) Integration of genetic algorithm and Gantt chart for job shop scheduling in distributed manufacturing systems. Comput Ind Eng 53:313–320. doi: 10.1016/j.cie.2007.06.024

12. Kaminsky P, Kaya O (2008) Inventory positioning, scheduling and lead-time quotation in supply chains. Int J Prod Econ 114:276–293. doi: 10.1016/j.ijpe.2008.02.006

13. Naderi B, Ruiz R (2010) The distributed permutation flowshop scheduling problem. Comput Oper Res 37:754–768. doi: 10.1016/j.cor.2009.06.019

14. De Giovanni L, Pezzella F (2010) An Improved Genetic Algorithm for the Distributed and Flexible Job-shop Scheduling problem. Eur J Oper Res 200:395–408. doi: 10.1016/j.ejor.2009.01.008

15. Shah NK, Ierapetritou MG (2012) Integrated production planning and scheduling optimization of multisite, multiproduct process industry. Comput Chem Eng 37:214–226. doi: 10.1016/j.compchemeng.2011.08.007

16. Behnamian J, Ghomi SMTF (2012) International Journal of Computer Integrated Manufacturing Incorporating transportation time in multi-agent production network scheduling. 37–41.

17. Behnamian J, Fatemi Ghomi SMT (2013) The heterogeneous multi-factory production network scheduling with adaptive communication policy and parallel machine. Inf Sci (Ny) 219:181–196. doi: 10.1016/j.ins.2012.07.020

18. Simchi-levi D, Kaminsky P, Simchi-levi E (2000) Designing and managing the supply chain, 3rd ed. New York:Irwin McGraw-Hill

19. H’Mida F, Lopez P (2012) Multi-site scheduling under production and transportation constraints. Int J Comput Integr Manuf 1–15. doi: 10.1080/0951192X.2012.688141

20. Huang J-Y, Yao M-J (2013) On the optimal lot-sizing and scheduling problem in serial-type supply chain system using a time-varying lot-sizing policy. Int J Prod Res 51:735–750. doi: 10.1080/00207543.2012.662604

21. Chung SH, Lau HCW, Choy KL, et al. (2010) Application of genetic approach for advanced planning in multi-factory environment. Int J Prod Econ 127:300–308. doi: 10.1016/j.ijpe.2009.08.019

22. Potts CN (1980) Technical Notes Analysis of a Heuristic for One Machine. Oper Res 28:1436–1441.

23. Herrmann JW, Lee C-Y (1993) On scheduling to minimize earliness-tardiness and batch delivery costs with a common due date. Eur J Oper Res 70:272–288.

24. Cheng TCE, Gordon VS, Kovalyov MY (1996) Single machine scheduling with batch deliveries. Eur J Oper Res 94:277–283. doi: 10.1016/0377-2217(96)00127-0

25. Hall NG, Potts CN (2003) Supply Chain Scheduling: Batching and Delivery. Oper Res 51:566–584.

26. Mahdavi-Mazdeh M, Shashaani S, Ashouri A, Hindi KS (2011) Single-machine batch scheduling minimizing weighted flow times and delivery costs. Appl Math Model 35:563–570. doi: 10.1016/j.apm.2010.07.023

27. Mahdavi-Mazdeh M, Sarhadi M, Hindi KS (2007) A branch-and-bound algorithm for single-machine scheduling with batch delivery minimizing flow times and delivery costs. Eur J Oper Res 183:74–86. doi: 10.1016/j.ejor.2006.09.087

28. Mahdavi-Mazdeh M, Sarhadi M, Hindi KS (2008) A branch-and-bound algorithm for single-machine scheduling with batch delivery and job release times. Comput Oper Res 35:1099–1111. doi: 10.1016/j.cor.2006.07.006

29. Rasti-barzoki M, Hejazi SR (2013) Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries for multiple customers in supply chains. Eur J Oper Res 228:345–357. doi: 10.1016/j.ejor.2013.01.002

30. Rasti-barzoki M, Hejazi SR, Mahdavi-Mazdeh M (2013) A branch and bound algorithm to minimize the total weighed number of tardy jobs and delivery costs. Appl Math Model 37:4924–4937. doi: 10.1016/j.apm.2012.10.001

31. Rinnooy K, A. H. G (1976) Machine scheduling problems: Classification, complexity and computations. Martinus Nijhoff, Hague, Neth

32. Chung C, Flynn J, Kirca O (2002) A branch and bound algorithm to minimize the total flow time for m -machine permutation flowshop problems. Intern J Prod Econ 79:185–196.

33. Bank M, Ghomi SMTF, Jolai F, Behnamian J (2012) Two-machine flow shop total tardiness scheduling problem with deteriorating jobs. Appl Math Model 36:5418–5426. doi: 10.1016/j.apm.2011.12.010