A benders decomposition algorithm for multi-factory scheduling problem with batch delivery

N. Karimi and H. Davoudpour*

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

Received 12 October 2015; received in revised form 19 January 2016; accepted 4 April 2016

KEYWORDS
Multi-factory scheduling; Batch delivery; Benders decomposition; Mixed-integer programming

Abstract. The multi-factory supply chain problem is investigated to determine the production and transportation scheduling of jobs, which are allowed to be transported by batches. This is a mixed-integer optimization problem, which could be challenging to solve. The problem incorporates two parts: (1) assigning jobs to the appropriate batch, and (2) scheduling jobs of batches for production and transportation. Based on the problem structure and because of its NP-hardness characteristics, Benders decomposition is recognized as a suitable approach. This approach decomposes the problem into assignment master problem and scheduling sub-problem. This would facilitate the solution procedure. By comparing performance of the proposed algorithm with an exact approach, i.e. Branch and Bound, it is demonstrated that it is able to find near-optimal solution in low computational times in comparison with the Branch and Bound.

© 2017 Sharif University of Technology. All rights reserved.

1. Introduction

Global manufacturing systems play an important role in maintaining competitive position in modern markets. Many industries, such as the steel corporations, electric power generating industries, automotive companies, food, and chemical industries, can take advantages of these systems. Establishing factories in different positions can reduce the system’s cost significantly. Material should be transferred among factories and should be delivered to the customers via transportation systems. Therefore, transportation becomes a very significant factor in such problems. Thus, scheduling production and transportation in such integrated systems would cause a trade-off between these factors.

Thus, many manufacturing companies have been transformed into global chains covering multi-factory systems [1]. Designing the supply chain in such systems becomes very significant and crucial because it would affect performance, reliability, and costs of the system. Planning and scheduling activities have become much more complex than the conventional single-factory scheduling problems since they involve many companies or factories across the entire supply chain Moon and Seo [2]. In such systems, factories can be structured in parallel or in series. Each factory is able to produce the finished goods (performs the whole production process) in parallel structure, but it is only able to perform parts of goods processing in a series structure. Finished goods of each factory are the raw material of the next factory in this structure for Karatza [3], Moon et al. [4], Jia et al. [5], Chan et al. [6], Chan et al. [7], Chung et al. [8], and Sun et al. [9] are some examples of parallel structure of factories. However, studies on serial structure are limited to HMida and Lopez [10], Huang and Yao [11], and Karimi and Davoudpour [12]. Here, serial multi-factory structure is investigated to fill
the gap of studies in this field. Interrelatedness of factories would cause high complexity of the structure, because material shortage in the upstream factories would affect the whole supply chain and cause delay in production of the downstream factories. Similarly, inventory accumulation and, therefore, stopping the production in the downstream factories would cause decrease or stop in production of upstream factories. Though, the jobs transportation among factories plays an important role in the scheduling of the whole system. Transporting a single job or batch of jobs, which is subject to a vehicle capacity, would have different effects on the system’s cost. Although using batch would lead to lower transportation cost, it would increase the total completion time (cost) of jobs. For this reason, transportation cost is considered as a function of number of deliveries for transferring all jobs. There are some studies in the literature that consider joint production scheduling and batching for delivery. A coordination of production and delivery scheduling, which can improve performance of the supply chain, has recently been considered by researchers [13-16]. Mahdavi-Mazdeh et al. [17] investigated minimizing the sum of the total weighted flow time and delivery cost in a single machine with batch delivery to a customer. The same problem considering multiple customers with zero and non-zero ready times was studied by Mahdavi-Mazdeh et al. [18], Mahdavi-Mazdeh et al. [19], Rastiabarzoki and Hejazi [20] studied an integrated due date assignment, single-machine production, and batch delivery scheduling problem for make-to-order production system.

All the above-cited references assume that jobs are delivered from the machine to the customer(s) in the single-machine scheduling problem. But, the transportation and delivery of jobs in a shop scheduling or multi-factory scheduling are not considered, except in our previous work [12]. We studied the multi-factory scheduling problem in which jobs transportation in the system (among the factories) and their delivery at the end of the system (from the last factory to the customer) were considered to minimize the sum of tardiness and transportation costs. As the problem is NP-hard, a Branch and Bound (B&B) method was presented, which could find the solution only for small to medium sizes of the problem.

A similar problem is considered here, which optimizes the sum of maximum completion cost of the jobs and transportation costs. In order to find the solution to this complex problem, Benders decomposition method is presented, which is a well-known technique for solving large-scale Mixed Integer Programming (MIP) problems Benders [21]. Its successful implementation in some planning and scheduling problems is presented in [22-24].

The rest of this article is organized as follows. In Section 2, the problem is described and the mathematical formulation is presented. Section 3 provides the Benders decomposition as a solution procedure to the problem. The experimental results are presented in Section 4. Conclusion and future directions are presented in Section 5.

2. Problem definition

A multi-factory production and transportation scheduling problem is addressed here, where factories are positioned in the series. There are n given jobs to be processed through F serial factories. The final products of each factory are considered as raw material of the next factory. They are transported and delivered to the next factory by means of transportation vehicle, which can transport a number of jobs as a batch at the same time. The batch transportation and delivery would reduce the transportation cost of the system, but they may increase the completion cost. The total transportation cost is an increasing function of the number of batches transported in the system. Though finding a batching scheme (the optimal number of batches and also the assignment of jobs to batches) and scheduling of batches in the whole system are the main aims of the problem, which help to minimize the sum of maximum completion cost and total transportation cost.

When processing of a job is finished in a factory, it should remain there until processing of the same batch’s uncompleted jobs is completed, and this will cause the increase in the maximum completion time of batches. The completion time of the last job in the batch is considered as the completion time of a batch. For this problem, it is assumed that: there are an infinite number of transportation vehicles with the same capacity and cost. All jobs are available at zero time. Jobs processing in each factory cannot be interrupted. Factories are always available with no breakdowns or scheduled/unscheduled maintenance. Infinite buffer exists around factories, before the first and after the last factories. Setup times are negligible. Jobs are available for processing in a factory immediately after arriving at 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. Completion time of a batch is the time when processing of the last job in the batch is completed. Transportation times between factories are considered. Jobs are available for transferring between factories immediately after completion of the processing of the whole batch included 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.
By considering single machine in each factory and neglecting the transportation time between factories and also considering a single job per transportation instead of batch of jobs, the problem can be simplified to a typical flowshop scheduling problem. As Rinooy Kan [25] proved that a flowshop scheduling problem was NP-hard, the mentioned problem is NP-hard too.

2.1. Mathematical formulation
Mathematical formulation of a serial multi-factory scheduling problem with batch delivery (transportation) is presented for more description of the problem and as a basis for the solution approach. The notations that are used in the model are introduced below:

Indices
- \( f \) Factory
- \( j \) Job
- \( h \) Batch

Parameters
- \( F \) Number of factories
- \( n \) Number of jobs to be processed
- \( B \) Capacity of each vehicle (maximum number of jobs in a batch)
- \( p_{ij} \) Processing time of job \( j \) in the factory \( f \)
- \( \tau_f \) The transportation time between factories \( f \) and \( f+1 \)
- \( \eta \) The cost of completion time
- \( \beta \) The cost of transportation

Decision variables
- \( \sigma_{jh} \) Equal to 1 if job \( j \) is positioned in batch \( h \), and 0 otherwise
- \( \Delta_h \) Equal to 1 if there is at least a job in batch \( h \), and 0 otherwise
- \( C_k^f \) Completion time of the batch \( h \) in factory \( f \)
- \( A_h \) The time that batch \( h \) arrive at customer
- \( C_{\text{max}} \) Maximum completion time of jobs in the system

The problem is formulated as a mixed integer programming model, which is as follows:

Minimize \( Z = \eta \times C_{\text{max}} + \beta \times \sum_{h=1}^{n} \Delta_h \) \hspace{1cm} (1)

\[ \sum_{h=1}^{n} \sigma_{jh} = 1 \quad j = 1, \ldots, n \] \hspace{1cm} (2)

\[ \Delta_h \geq \Delta_{h+1} \quad h = 1, \ldots, n-1 \] \hspace{1cm} (3)

\[ \sum_{j=1}^{n} \sigma_{jh} \leq B \times \Delta_h \quad h = 1, \ldots, n \] \hspace{1cm} (4)

\[ C_k^f \geq C_{k-1}^f + \sum_{j=1}^{n} (P_{jf} \times \sigma_{jh}) \quad h = 2, \ldots, n, \quad f = 1, \ldots, F \] \hspace{1cm} (5)

\[ C_k^f \geq A_k^f + \sum_{j=1}^{n} (P_{jf} \times \sigma_{jh}) \quad h = 1, \ldots, n, \quad f = 1, \ldots, F \] \hspace{1cm} (6)

\[ A_{h+1}^f \geq C_h^f + \tau_f \quad h = 1, \ldots, n, \quad f = 1, \ldots, F \] \hspace{1cm} (7)

\[ C_{\text{max}} \geq C_h^f \quad h = 1, \ldots, n, \quad f = 1, \ldots, F \] \hspace{1cm} (8)

\[ \sigma_{jh} = \begin{cases} 1 & \text{if job } j \text{ is assigned to batch } h \\ 0 & \text{otherwise} \end{cases} \] \hspace{1cm} (9)

\[ \Delta_h = \begin{cases} 1 & \text{if it is not a null batch} \\ 0 & \text{otherwise} \end{cases} \] \hspace{1cm} (10)

\[ C_h^f, A_h^f \geq 0 \quad h = 1, \ldots, n, \quad f = 1, \ldots, F \] \hspace{1cm} (11)

The objective function minimizes the maximum completion cost and total transportation cost. The maximum completion cost is shown in Eq. (1) by \( \eta \times C_{\text{max}} \). As it is clear, \( \sum_{h=1}^{n} \Delta_h \) denotes the number of batches transferred between factories and also delivered to the customers. As each job should be assigned to only one batch, Constraint (2) is correct. Constraint (3) indicates that a job cannot be assigned to a batch if the previous batch is empty. The transportation vehicle has the limited capacity (\( B \)); thus, the number of jobs contained in each batch is limited. This limitation is insured by Constraint (4). This constraint also associates the number of jobs in the batch to the state of the batch. It denotes that if a batch is null (\( \Delta_h = 0 \)), no job should be assigned to it (\( \sigma_{jh} = 0 \)). Constraints (5) and (6) prevent beginning of the \( h \)th batch in a factory unless it is delivered from the previous factory and processing of the previous batch is completed at this factory. Constraint (7) guarantees that arrival time of a batch at a factory is at least the sum of completion time of that batch at the previous factory and its transportation time to the current factory. Constraint (8) determines the value of \( C_{\text{max}} \). In order to assign jobs to batches, the required binary variables are defined by Constraint (9). Constraint (10) defines \( \Delta_h \) to denote the batch condition; it means that if
there is at least one job in the batch $h$, $\Delta_h$ is equal to 1. In order to schedule batches, two non-negative continuous variables $A_k^h$ and $C_k^f$ are defined, which are the arrival time and completion time of batch $h$ to factory $f$, respectively; these variables are defined using Constraint (11).

3. Benders decomposition

Benders decomposition is a well-known approach for handling complicating variables, which increase computational difficulty of the problem. In fact, it is appropriate for modeling of the problem, which can be partitioned into smaller ones. In this approach, the overall formulation of the problem should be decomposed into smaller problems: master problem and sub-problem(s); and, then, they should be solved iteratively. At each iteration, solution of the master problem is used in the sub-problem as input data and solution of the sub-problem is inserted into the master problem as a new constraint, which is named ‘Benders’ Cut. When the sub-problem cannot find a feasible solution, the feasibility cut is added to the master problem. It directs the master solution to feasible region. Otherwise, the optimality cut is added to the master problem to enhance the quality of the solution by guiding the search to the optimal region. At the end of each iteration, convergence criteria of the method are checked.

3.1. Basics of Benders decomposition

Suppose the following original problem, which contains two sets of decision variables $x$ and $y$:

$$ \min \ bx + dy, $$  

$$ Ax + Ey \leq B, $$  

$$ 0 \leq x \leq x^{up}, $$  

$$ 0 \leq y \leq y^{up}, $$

where $x$ is the set of complicating variables, which make the solution procedure of the model complex. Thus, separating these variables from the model would help the model to be solved in a simpler way. Master problem consists of only complicating variables and sub-problems contain other variables.

By considering $\bar{x}$ as the fixed value for the complicating variables $x$, the sub-problem of the original problem can be defined as follows:

$$ \min \ b\bar{x} + dy, $$  

$$ Ey \leq B - A\bar{x}, $$  

$$ 0 \leq y \leq y^{up}. $$

The dual form of the sub-problem can be written as follows:

$$ \max \ (B - \bar{A}x)\lambda, $$  

$$ E^T\lambda \leq d, $$  

$$ \lambda \geq 0. $$

The advantage of using this dual model is that its solution space is not dependent on $x$ and it is only related to the objective function. In other word, different values of $x$ would not affect the solution space.

The unbounded solution (extreme ray) of the dual problem shows the infeasibility of the sub-problem. Thus, the feasibility cut is needed to be added to the master problem. Otherwise, the optimality cut is generated to lead the search space to better solution and to close the optimality gap. The master problem is composed of all constraints of the problem, which are just related to the complicating variable. It also consists of cuts, which are added at each iteration:

$$ \min \ \alpha $$  

$$ \lambda^p_i (b - ax) + dy^p_i \leq \alpha \quad i = 1, \ldots, \text{iter} - 1, $$  

$$ \lambda^i_i (b - ax) + dy^i_i \geq 0 \quad i = 1, \ldots, \text{iter} - 1, $$  

$$ 0 \leq x \leq x^{up}, $$

where $\lambda^p$ and $\lambda^i$ are the extreme points and the extreme rays obtained from the dual sub-problem at iteration $i$. Constraint (23) shows the optimality cut of the problem, where $\alpha$ is an auxiliary continuous variable, which approximates the original objective function using the solution achieved from the sub-problem. Constraint (24) is the feasibility cut of the problem, which guides the search to feasible region. The $\text{iter}$ in Constraints (23) and (24) is related to the number of iterations (note that in the first iteration, the master problem is solved without any benders cuts). In fact, these constraints cause the relation between the sub-problem and master problem.

As mentioned earlier, convergence criteria should be investigated at the end of each iteration. One of these criteria is the predetermined gap between upper and lower bounds (optimality gap). The optimal value of the master problem provides us with the lower bound for the original problem (lower bound: $\alpha$). The upper bound of the problem is estimated by sum of master problem and sub-problem objective function (note that the objective function value of the sub-problem is equal to the objective function value of its dual problem (upper bound: $\lambda^i (b - ax^*) + dy^i$)). In addition, the index $i$ is used to show the value of variable in iteration $i$. The other criterion is maximum number of iterations of the method. When the method meets one of the criteria, it is converged and should stop.
3.2. Benders reformulation

Since our problem contains binary variables, which are considered as complicating variables, Benders decomposition is a suitable method for solving it. Thus, by fixing their values, complexity of the problem would decrease considerably. One can observe that once the jobs assignment has been fixed, one can solve the scheduling problem in a simpler way. Thus, the role of master problem in this context is recognizing the good assignments of jobs to batches and determining the required number of batches. The sub-problem schedules batches production and delivery in the whole system. To decompose the considered model to be solvable by Benders decomposition, a master and a sub-problem would be created.

Constraints only related to the complicating variables are assigned to the master problem and the others are assigned to the sub-problem. The primal Benders Sub-Problem (SP) is as follows:

SP:

\[
\text{Minimize } Z_t = \eta \times C_{\text{max}} + \beta \times \sum_{h=1}^{n} \Delta_h, \quad (26)
\]

\[
C^f_h \geq C^f_{h-1} + \sum_{j=1}^{n} (P_{jf} \times \sigma^f_{j,k}) \\
h = 2, \ldots, n, \ f = 1, \ldots, F, \quad (27)
\]

\[
C^f_h \geq A^f_h + \sum_{j=1}^{n} (P_{jf} \times \sigma^f_{j,k}) \\
h = 1, \ldots, n, \ f = 1, \ldots, F, \quad (28)
\]

\[
A^{f+1}_h \geq C^f_h + \tau^f \\
h = 1, \ldots, n, \ f = 1, \ldots, F - 1, \quad (29)
\]

\[
C_{\text{max}} \geq C^f_h \\
h = 1, \ldots, n, \ f = 1, \ldots, F, \quad (30)
\]

\[
C^f_h, A^f_h \geq 0 \\
h = 1, \ldots, n, \ f = 1, \ldots, F. \quad (31)
\]

\[
\Delta_h \text{ and } \sigma^f_{j,k} \text{ are the fixed values of } \Delta_h \text{ and } \sigma^f_{j,k}, \text{ achieved from master problem (for the first iteration of the method, these values are set arbitrarily). Defining the dual variables } s^f_{h,i}, u^f_{h,i}, v^f_{h,i}, w^f_{h,i} \text{ associated with Constraints (27)-(30) the dual model of the SP (DSP) can be described as follows:}
\]

DSP:

\[
\text{max } Z_d = \sum_{h=2}^{NB} \sum_{f=1}^{F} \sum_{j=1}^{n} s^f_{h,j} \times P_{jf} \times \sigma^f_{j,k} \times \sum_{h=1}^{NB} \sum_{f=1}^{F} \sum_{j=1}^{n} u^f_{h,j} \\
\times P_{jf} \times \sigma^f_{j,k} + \sum_{h=1}^{NB} \sum_{f=1}^{F} v^f_{h} \times \tau^f, \quad (32)
\]

\[
-s^f_{h+1} + u^f_{h,i} - v^f_{h,i} \leq 0 \quad h = 1, \quad (33)
\]

\[
1 \leq f \leq F,
\]

\[
s^f_{h} - s^f_{h+1} + u^f_{h,i} - v^f_{h,i} \leq 0 \quad h = 2, \ldots, NB, \quad (34)
\]

\[
1 \leq f \leq F,
\]

\[
s^f_{h} + u^f_{h,i} - v^f_{h,i} \leq 0 \quad h = NB, \quad (35)
\]

\[
1 \leq f \leq F,
\]

\[
-w^f_{h} \leq 0 \quad h \leq NB, \quad f = 1, \quad (36)
\]

\[
v^{f-1}_{h} - w^{f-1}_{h} \leq 0 \quad h \leq NB, \quad f = F. \quad (37)
\]

\[
-w^f_{h} + v^{f-1}_{h} \leq 0 \quad h \leq NB, \quad 1 < f < F. \quad (38)
\]

\[
w^f_{h} \leq \eta \quad 1 \leq h \leq NB, \quad 1 \leq f \leq F. \quad (39)
\]

\[
s^f_{h}, u^f_{h,i}, v^f_{h,i}, w^f_{h,i} \geq 0 \quad 1 \leq h \leq NB, \quad 1 \leq f \leq F. \quad (40)
\]

where \(NB\) is the number of non-empty batches; in other words, by number of batches with \(\Delta_h = 1\), which is obtained from the MP solution, Eq. (32) is objective function of the maximization dual problem subject to Constraints (33)-(40). The Benders cut is deduced from the solution of the DSP at the end of each iteration and it is added to the Master Problem (MP):

MP:

\[
\text{Minimize } Z_d = \alpha, \quad (41)
\]

\[
\sum_{h=1}^{n} \sigma_{j,k} = 1, \quad j = 1, \ldots, n, \quad (42)
\]

\[
\Delta_h \geq \Delta_{h+1} \quad h = 1, \ldots, n - 1, \quad (43)
\]

\[
\sum_{j=1}^{n} \sigma_{j,k} \leq B \times \Delta_h \quad h = 1, \ldots, n, \quad (44)
\]
\[ \sum_{h=2}^{NB} F \sum_{j=1}^{n} s_h^j \times P_{jf} \times \sigma_{jh} + \sum_{h=1}^{NB} \sum_{f=1}^{F} v_f^h \times P_{jf} \times \sigma_{jh} + \sum_{h=1}^{NB} \sum_{f=1}^{F} v_f^h \times \tau_f \leq \alpha, \]  
where 
\[ \sigma_{jh} = \begin{cases} 
1 & \text{if job } j \text{ is assigned to batch } h \\
0 & \text{otherwise}
\end{cases} \]

\[ \Delta_h = \begin{cases} 
1 & \text{if } h = 1, \ldots, n, \text{ is not a null batch} \\
0 & \text{otherwise}
\end{cases} \]

\[ s_h^j, u_h^j, \text{ and } w_f^h \text{ are the solutions of DSP to } s_h^j, \\
\text{ and } u_h^j, \text{ and } w_f^h \text{ are considered to have fixed values in MP (note that since the SP would never generate the infeasible solution to our problem, feasibility cut is not required to be added to the MP). After solving MP, the lower and upper bounds of the problem are calculated to investigate whether the method should be terminated or continued.} \]

4. Experimental results

This section contains the computational experiments for evaluation of the proposed Benders decomposition method for the mentioned problem. The algorithm is coded in commercial software GAMS and executed on a PC with Intel Core 2 Duo and 2 GB of RAM memory.

To the best knowledge of the authors, this paper is a novel research in the scheduling field; therefore, there is lack of benchmark or competent study on this problem for evaluating the method. Thus random datasets for different sizes of the problem are generated for assessing and checking efficiency of the method against an adapted B&B presented in [12]. Features of the generated test problems are described in the following: then, the computational experiments are presented.

We generate 10 random data sets of different problem sizes with number of jobs ranging from 5 to 200 and number of factories ranging from 2 to 8. All the processing times and transportation times are generated randomly using uniform distribution with parameters \([1, 99]\) and \([50, 200]\), respectively. Capacity of each transportation vehicle has a uniform distribution with parameters \([2, 20]\). Coefficient of transportation is considered in cases of equal, greater, and smaller than the coefficient of makespan using the values 10 and 1000. The optimality gap and maximum number of iterations of the Benders decomposition as stopping criteria of the method are set to 0.01 and 40, respectively. In order to show the size alteration through different instances, Table 1 is presented, which contains number of constraints and variables that are involved for solving Benders decomposition.

Seeking to evaluate the performance of our approach, different experiments are considered in terms of lower and upper bounds, objective function value, computing times, and number of Benders cuts. First, for illustrating the progression of upper- and lower-bound values through iterations, Figure 1 is presented. This figure shows these iterative values for the problem instance with 100 jobs and 5 factories.

It is clear from the figure that deviation of the lower and upper bounds narrows during iterations until the algorithm meets the convergence criteria (the allowed gap between lower and upper bounds) through 19 iterations and, then, it stops.

<table>
<thead>
<tr>
<th>[n]</th>
<th>[F]</th>
<th>Constraints</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>76</td>
<td>88</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>177</td>
<td>112</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>288</td>
<td>170</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>334</td>
<td>142</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>478</td>
<td>185</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>541</td>
<td>172</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>763</td>
<td>269</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>2851</td>
<td>368</td>
</tr>
<tr>
<td>50</td>
<td>5</td>
<td>3538</td>
<td>748</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>10701</td>
<td>693</td>
</tr>
<tr>
<td>100</td>
<td>5</td>
<td>12003</td>
<td>1374</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>23563</td>
<td>1027</td>
</tr>
<tr>
<td>150</td>
<td>5</td>
<td>25738</td>
<td>1691</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
<td>41401</td>
<td>1383</td>
</tr>
<tr>
<td>200</td>
<td>5</td>
<td>42913</td>
<td>1939</td>
</tr>
</tbody>
</table>

Figure 1. Iterative results for lower and upper bounds.
For investigating computing times and quality of the solution, the method is compared with an adapted B&B algorithm [12] for this problem, which is capable of finding exact solution. Note that the execution times of both methods are limited to 6000 (s) and if the method cannot find the solution for an instance in that time limit, (-) is put in the relative cell of the table.

The columns O.F and R.T represent objective function values and run times of each method and the column cuts shows the number of cuts generated by Benders method to find the solution. Vars and Cons are representatives of numbers of variables and constraints generated by the Benders decomposition method and the Gap value is the relative deviation of the Benders decompositions objective function value from the objective function value of the exact solution of B&B.

Tables 2-4 display the computational results of the mentioned comparison. It is clear that, since B&B achieves the exact solution, the quality of its solution is better than the solution of Benders decomposition but its procedure is too time-consuming in finding the solution. Though, the performance of Benders decomposition can be investigated by means of its

<table>
<thead>
<tr>
<th>n × F</th>
<th>Branch and bound</th>
<th>Benders decomposition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O.F</td>
<td>R.T</td>
</tr>
<tr>
<td>5× 2</td>
<td>5940</td>
<td>0.267</td>
</tr>
<tr>
<td>5× 5</td>
<td>12010</td>
<td>0.421</td>
</tr>
<tr>
<td>5× 8</td>
<td>18200</td>
<td>0.438</td>
</tr>
<tr>
<td>10× 2</td>
<td>9410</td>
<td>0.38</td>
</tr>
<tr>
<td>10× 5</td>
<td>15330</td>
<td>2.658</td>
</tr>
<tr>
<td>10× 8</td>
<td>20400</td>
<td>6.636</td>
</tr>
<tr>
<td>15× 2</td>
<td>11700</td>
<td>1.07</td>
</tr>
<tr>
<td>15× 5</td>
<td>17730</td>
<td>174.65</td>
</tr>
<tr>
<td>15× 8</td>
<td>22060</td>
<td>275.48</td>
</tr>
<tr>
<td>20× 2</td>
<td>14120</td>
<td>2.133</td>
</tr>
<tr>
<td>20× 5</td>
<td>20100</td>
<td>2409.59</td>
</tr>
<tr>
<td>20× 8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>50× 2</td>
<td>29080</td>
<td>10.75</td>
</tr>
<tr>
<td>50× 5</td>
<td>35060</td>
<td>4466</td>
</tr>
<tr>
<td>50× 8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>100× 2</td>
<td>51720</td>
<td>81.688</td>
</tr>
<tr>
<td>100× 5</td>
<td>60330</td>
<td>258.59</td>
</tr>
<tr>
<td>100× 8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>150× 2</td>
<td>78130</td>
<td>322.76</td>
</tr>
<tr>
<td>150× 5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>150× 8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>200× 2</td>
<td>102570</td>
<td>784.1</td>
</tr>
<tr>
<td>200× 5</td>
<td>109640</td>
<td>2510.747</td>
</tr>
<tr>
<td>200× 8</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 3. Comparison of Benders decomposition performance with branch and bound for \( \eta < \beta \) (\( \eta = 10 \) and \( \beta = 1000 \)).

\[ \begin{array}{cccccccccc}
\text{n \times F} & \text{Branch and bound} & \text{Benders decomposition} \\
 & \text{O.F} & \text{R.T} & & \text{O.F} & \text{R.T} & \text{Cuts} & \text{Vars} & \text{Cons} & \text{GAP} \\
5 \times 2 & 9060 & 0.337 & & 9300 & 0.788 & 3 & 64 & 40 & 0.02649 \\
5 \times 5 & 16270 & 0.31 & & 17150 & 0.651 & 3 & 103 & 58 & 0.05408 \\
5 \times 8 & 22379 & 0.584 & & 23590 & 0.753 & 3 & 142 & 76 & 0.05411 \\
10 \times 2 & 14700 & 0.315 & & 15270 & 0.994 & 4 & 177 & 76 & 0.03877 \\
10 \times 5 & 22170 & 2.67 & & 24180 & 0.669 & 3 & 243 & 100 & 0.09066 \\
10 \times 8 & 28070 & 8.91 & & 30090 & 1.256 & 4 & 339 & 148 & 0.07196 \\
15 \times 2 & 19620 & 0.645 & & 20120 & 1.497 & 4 & 340 & 111 & 0.02548 \\
15 \times 5 & 26990 & 61.25 & & 28060 & 1.45 & 4 & 448 & 154 & 0.03964 \\
15 \times 8 & 33020 & 40.294 & & 36390 & 9.992 & 4 & 562 & 202 & 0.10205 \\
20 \times 2 & 24090 & 2.32 & & 24600 & 2.086 & 4 & 541 & 136 & 0.02117 \\
20 \times 5 & 31030 & 28.015 & & 32960 & 1.87 & 4 & 688 & 196 & 0.06219 \\
20 \times 8 & 38260 & 167.87 & & 41670 & 2.783 & 5 & 859 & 274 & 0.08912 \\
50 \times 2 & 54730 & 15.315 & & 55410 & 3.035 & 3 & 2851 & 330 & 0.00749 \\
50 \times 5 & 61400 & 107.25 & & 64640 & 7.019 & 7 & 3253 & 506 & 0.05276 \\
50 \times 8 & - & - & & 70020 & 156.7 & 10 & 3619 & 654 & - \\
100 \times 2 & 101220 & 144.87 & & 101710 & 10.07 & 2 & 10881 & 804 & 0.00484 \\
100 \times 5 & 110620 & 24.94 & & 113650 & 49.782 & 9 & 11638 & 1019 & 0.02739 \\
100 \times 8 & - & - & & 124370 & 54.688 & 9 & 12930 & 1788 & - \\
150 \times 2 & 152380 & 164.28 & & 152470 & 69.55 & 4 & 23551 & 981 & 0.00039 \\
150 \times 5 & 179370 & 1274.52 & & 162920 & 363.41 & 15 & 24853 & 1574 & 0.02240 \\
150 \times 8 & - & - & & 171690 & 620 & 20 & 26587 & 2458 & - \\
200 \times 2 & 201670 & 961.331 & & 209610 & 83.32 & 2 & 41509 & 1394 & 0.03937 \\
200 \times 5 & 209200 & 758.34 & & 218670 & 487.63 & 6 & 43364 & 2429 & 0.01526 \\
200 \times 8 & - & - & & 225180 & 1078.98 & 14 & 44149 & 2533 & - \\
\end{array} \]

run time and gap. By increasing the size of problem instances, the advantage of Benders decomposition, in terms of computational time, intensifies relative to the B&B. For smaller sizes of the instances, it expended larger computational time than B&B. As B&B is an exact solution approach, it is not capable of finding solution for large problem instances in a reasonable time. It is obvious that for larger sizes of the problem, Benders decomposition searches the solution space in bigger number of iterations and, thus, bigger number of cuts are generated for guiding the search procedure.

From the objective function and gap values for the Benders decomposition, it is understood that it can find near-optimal solution in very smaller time than the B&B.

5. Conclusion

In this paper, we have presented the Benders decomposition to solve a multi-factory scheduling with batch delivery among factories and also to the final customer. The problem entails minimizing the costs associated with maximum completion time and transportation in
the system and at the end of the system. The main contribution of this paper is Benders reformulation of the problem, which facilitates the solution approach by decomposing the hard problem to two simpler problems. The objective function of the master problem can also be considered as the lower bound of the original problem.

Numerical experiments were conducted to evaluate the efficiency of the proposed method tackling a large-scale real-world problem. The comparison of this method with the exact solution approach, adapted B&B method, presented in [12] is performed. The experimental results confirm the superior performance of our presented method in terms of the run time to the B&B algorithm, especially for a larger number of problem instances. It is clear that for larger sizes, the B&B cannot find the solution in reasonable time, but Benders decomposition is capable of finding near-optimal solution in the considered time.

For future research, we are seeking for ways to accelerate the Benders decomposition algorithm, such as developing a method to generate a set of cuts at each iteration, for this problem. In addition, some more research can be done with different assumptions of this problem, like unlimited number of vehicles, unlimited buffers, etc.

Table 4. Comparison of Benders decomposition performance with branch and bound for \( \eta > \beta \) (\( \eta = 1000 \) and \( \beta = 10 \)).

<table>
<thead>
<tr>
<th>( n \times F )</th>
<th>Branch and bound</th>
<th>Benders decomposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n \times 2 )</td>
<td>( n \times 5 )</td>
<td>( n \times 8 )</td>
</tr>
<tr>
<td>( 5 \times 2 )</td>
<td>O.F.</td>
<td>R.T.</td>
</tr>
<tr>
<td>590040</td>
<td>0.439</td>
<td>598650</td>
</tr>
<tr>
<td>1199050</td>
<td>0.483</td>
<td>1254050</td>
</tr>
<tr>
<td>1815050</td>
<td>0.721</td>
<td>1820050</td>
</tr>
<tr>
<td>10 \times 2</td>
<td>933060</td>
<td>0.432</td>
</tr>
<tr>
<td>10 \times 5</td>
<td>1527300</td>
<td>2.786</td>
</tr>
<tr>
<td>10 \times 8</td>
<td>2030100</td>
<td>7.907</td>
</tr>
<tr>
<td>15 \times 2</td>
<td>1162060</td>
<td>1.48</td>
</tr>
<tr>
<td>15 \times 5</td>
<td>1781300</td>
<td>76.78</td>
</tr>
<tr>
<td>15 \times 8</td>
<td>2281150</td>
<td>303.21</td>
</tr>
<tr>
<td>20 \times 2</td>
<td>1401110</td>
<td>1.791</td>
</tr>
<tr>
<td>20 \times 5</td>
<td>1992180</td>
<td>1317.36</td>
</tr>
<tr>
<td>20 \times 8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>50 \times 2</td>
<td>2972260</td>
<td>12.29</td>
</tr>
<tr>
<td>50 \times 5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>75 \times 8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>100 \times 2</td>
<td>5121510</td>
<td>1277.4</td>
</tr>
<tr>
<td>100 \times 5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>100 \times 8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>150 \times 2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>150 \times 5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>150 \times 8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>200 \times 2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>200 \times 5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>200 \times 8</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
References


Biographies

Neda Karimi received her PhD degree in Industrial Engineering in 2016. Her research is mainly focused on scheduling, supply chain, and mathematical modeling.

Hamid Davoudpour has been a Faculty Member at Amirkabir University, Tehran, Iran, since 1978.
He is currently Associate Professor in the Faculty of Industrial and Systems Engineering and Management.

His research interests include production planning and scheduling, and locations and allocation problems.

He is Chief Editor of International Journal of Modern Science and Technology. He has published 12 books and more than 150 papers in international journals and presented many others at national and international conferences.