Scientia Iranica E (2018) 25(6), 3765–3775

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

Lagrangian relaxation approach to minimizing makespan in hybrid flow shop scheduling problem with unrelated parallel machines

E. Asadi-Gangraj*

Department of Industrial Engineering, Babol Noshirvani University of Technology, Babol, Iran.

Received 9 April 2017; received in revised form 29 September 2017; accepted 25 December 2017

KEYWORDS
Hybrid flow shop;
Makespan;
Lagrangian relaxation;
Simplification of subproblems;
Dominance rules.

Abstract. This research addresses scheduling problem of n jobs in a Hybrid Flow Shop (HFS) with unrelated parallel machines in each stage. A monolithic Mixed Integer Linear Programming (MILP) model is presented to minimize the maximum completion time (makespan). As the research problem is shown to be strongly NP-hard, a Lagrangian Relaxation (LR) algorithm is developed to handle the HFS scheduling problem. We design two approaches, namely, simplification of subproblems and dominance rules, to solve the subproblems which are generated in each iteration. For evaluation purposes, numerical experiments with small- and large-size problems are randomly generated with up to 50 jobs and 4 stages. The experimental results show that the Lagrangian relaxation approaches outperform the MILP model with respect to CPU time. Furthermore, from the results, it can be concluded that the simplification of subproblems shows slightly better solutions than dominance rules do in solving the subproblems.

© 2018 Sharif University of Technology. All rights reserved.

1. Introduction

Nowadays, production scheduling is a complicated decision making process that has a considerable role in the competitive manufacturing and production environment. As a result, effective scheduling can lead to significant improvements in various performance measures like customer satisfaction, production costs, throughput, and bottleneck resources utilization [1].

This paper aims to minimize makespan in a hybrid flow shop environment with unrelated parallel machines. In this production system, machines are arranged into m stages in series, in which in a particular stage, i, there are S_i unrelated parallel machines. Job j has finite processing time and must be processed on only one machine in each stage. Preemption is not acceptable for the jobs and every machine must process only one job at a time.

The hybrid flow shop scheduling problem has received a great deal of attention due to its increasing applicability in many industries, such as automobile assembly plants [2] and packaging industry [3], during recent years [4]. Recently, different solution approaches like exact, heuristic, and metaheuristic are developed for the HFS problems. Moursli and Pochet [5] proposed a branch-and-bound algorithm for the hybrid flow shop scheduling problem with makespan minimization. They developed some heuristics and single-stage subproblem relaxation to compute the upper and lower bounds, respectively. Liu and Chang [6] considered a Lagrangian relaxation-based approach for hybrid flow shop scheduling problem with sequence-dependent setup time and cost and earliness-tardiness minimization.

Fattahi et al. [7] proposed a hierarchical branch-and-bound approach to solve the HFS problem with

*. *E-mail address: e.asadi@nit.ac.ir*

doi: 10.24200/sci.2017.20018

sequence-dependent setup time and one assembly stage. The selected objective function was to minimize the completion time of all products (makespan). Riane et al. [8] proposed two heuristic methods, based on dynamic programming and branch-and-bound method, to minimize the makespan in three-stage hybrid flow shop scheduling problem.

In the literature, heuristic methods and Branch-and-Bound are the two mostly employed approaches for the HFS scheduling problems [9]. Marichelvam et al. [10] considered a multistage hybrid flow shop scheduling problem to minimize the makespan. They proposed a new bat algorithm to solve the problem. Jolai et al. [11] investigated no-wait two-stage flexible flow shop scheduling to minimize makespan and maximum tardiness. They developed three metaheuristic approaches based on simulated annealing algorithm. Marichelvam et al. [12] proposed Cuckoo Search (CS) metaheuristic algorithm to solve the multistage hybrid flow shop scheduling problem for minimizing the makespan.

Due to the huge computational time in branch-and-bound approaches for large-size problems, industries commonly prefer to apply the heuristic methods [13]. On the other side, heuristic approaches cannot guarantee achieving the optimal or near-optimal solutions. Therefore, it is inevitable to develop appropriate approaches to solve the HFS scheduling problems in large size scale. Lagrangian relaxation approach has been introduced as a powerful tool to solve various optimization problems and it can be an efficient and effective approach to solving the HFS problems [14].

Liu and Luh [15] developed a novel "separable" integer programming formulation for permutation flow shop scheduling problem and proposed a Lagrangian relaxation method to minimize penalties on jobs tardiness and earliness of releasing the raw materials. Luh and Hoitomt [16] proposed Lagrangian relaxation algorithms for some scheduling problems, including single-operation and multiple-operation jobs with simple fork-join precedence constraints on identical parallel machines and job shop scheduling problem, respectively.

Chang et al. [17] developed a Lagrangian relaxation method and minimum cost linear network flow to solve the hybrid flow shop with earliness-tardiness cost. The numerical results indicated that their scheduling algorithm could achieve the near-optimal solution in a reasonable computational time. Irohara [18] proposed three Lagrangian relaxation methods for the HFS problem with limited buffer capacity to minimize the weighted earliness-tardiness cost. They relaxed machine capacity and precedence constraints to schedule all stages together.

Sun and Nobel [19] considered a job shop scheduling problem with sequence-dependent setup time. They decomposed the problem into some single-machine subproblems via shifting bottleneck procedure and employed Lagrangian relaxation method to solve the subproblems. Tang et al. [14] proposed a Lagrangian relaxation approach for the hybrid flow shop with identical machines in each stage to minimize the sum of the weighted completion times. They proposed dynamic programming algorithm to solve the parallel identical machine subproblems.

Emami et al. [20] proposed Lagrangian relaxation algorithm for robust Order Acceptance and Scheduling (OAS) problem in an unrelated parallel machines environment to maximize the profit. They also applied a cutting plane method to update the Lagrangian multipliers in each iteration and introduced a heuristic method to generate the feasible solutions. Shishebori et al. [21] proposed a mixed integer programming model for a fuzzy robust multi-objective facility location network design problem. They developed a Lagrangian relaxation approach to solve the problem precisely.

During the last years, many searchers have studied and solved the HFS problems and significant efforts will be put to finding better solutions to this problem in future. This research proposes a decomposition approach based on Lagrangian relaxation approach to hybrid flow shop scheduling problem and two approaches, namely simplification of subproblems and dominance rules, are developed to solve the decomposed unrelated parallel machine subproblems.

The rest of the paper is organized as follows. Section 2 is devoted to describing an integer programming formulation of the HFS problem proposed by Nahavandi and Asadi Gangraj [22]. Section 3 presents a Lagrangian relaxation method for the HFS problem with makespan minimization. Computational results are presented in Section 4 based on some test problems. Finally, some concluding remarks and future research areas are given in Section 5.

2. Problem statement

2.1. Problem description

In an HFS, machines are arranged into m stages in series. On a particular stage i, there are S_i unrelated parallel machines and job j must be processed on machine k in each stage i with deterministic processing time p_{ijk}. The travel time between stages is negligible and there is not any constraint on buffer capacity between stages in the production line. Preemption is not allowed and the processing time involves both setup time and processing time of the job. One job must be processed only by one machine at any time in each stage and every machine can process only one job at a time. The selected objective function is to minimize the maximum completion time (makespan).

2.2. Mathematical model

This section is devoted to introducing the Mixed Integer Linear Programming (MILP) model of the HFS problem with unrelated parallel machines, which was proposed by Nahavandi and Asadi Gangraj [22]. They proposed a monolithic MILP model for the HFS problem, which required the following indices, parameters, and decision variables:

Indices:

j, l	Job index;
i, h	Stage index;
k	Machine index in each stage;
n	Number of jobs;
m	Number of stages;
R	Set of job indices necessary to define decision variables ($R = \{j, l \| j < l\}$).

Parameters:

S_i	Number of machines in stage i;
M	A very large number;
p_{ijk}	Processing time of job j in stage i on machine k.

Decision variables:

C_{ij}	Completion time of job j in stage i.

$$X_{ijk} = \begin{cases} 1 & \text{if job } j \text{ is assigned to machine } k \text{ in stage } i \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ilj} = \begin{cases} 1 & \text{if job } l \text{ is processed earlier than job } j \text{ in stage } i \\ 0 & \text{otherwise} \end{cases}$$

$$W_{ilj} = \begin{cases} 1 & \text{if job } l \text{ and job } j \text{ are processed on the same machine in stage } i \\ 0 & \text{otherwise} \end{cases}$$

By using the above notation, they formulated the problem as MILP, as follows:

$$\text{Min } Z = \max\{C_{mj}\} \qquad j = 1, 2, \cdots, n. \tag{1}$$

Subject to:

$$\sum_{k=1}^{S_i} X_{ijk} = 1 \qquad j = 1, 2, \cdots, n \qquad i = 1, 2, \cdots, m, \tag{2}$$

$$C_{1j} \geq \sum_{k=1}^{s_1} p_{1jk}.X_{1jk} \qquad j = 1, 2, \cdots, n, \tag{3}$$

$$C_{ij} \geq C_{i-1,j} + \sum_{k=1}^{S_i} p_{ijk}.X_{ijk} \qquad j = 1, 2, \cdots, n \qquad i = 2, \cdots, m, \tag{4}$$

$$C_{ij} + M \times (1 - Y_{ilj}) \geq C_{il} + p_{ijk}.X_{ijk} \qquad j, l = 1, 2, \cdots, n \quad \& \quad (j, l) \in R \qquad k = 1, 2, \cdots, S_i; \qquad i = 1, 2, \cdots, m, \tag{5}$$

$$C_{il} + M \times (1 - W_{ilj} + Y_{ilj}) \geq C_{ij} + p_{ilk}.X_{ilk} \qquad j, l = 1, 2, \cdots, n \quad \& \quad (j, l) \in R \qquad k = 1, 2, \cdots, S_i; \qquad i = 1, 2, \cdots, m, \tag{6}$$

$$W_{ijl} \geq X_{ijk} + X_{ilk} - 1 \qquad j, l = 1, 2, \cdots, n \quad \& \quad (j, l) \in R \qquad k = 1, 2, \cdots, S_i; \qquad i = 1, 2, \cdots, m, \tag{7}$$

$$C_{ij} \geq 0 \qquad j, l = 1, 2, \cdots, n, \qquad X_{ijk}, Y_{ilj}, W_{ilj} \in \{0, 1\} \qquad i = 1, 2, \cdots, m \qquad k = 1, 2, \cdots, S_i. \tag{8}$$

Expression (1) represents the makespan objective function, which is equal to maximum completion time. Constraint set (2) ensures that each job must be assigned to one machine at each stage. Constraint set (3) indicates that completion time of job j in the first stage is greater than or equal to its processing time in this stage. The relation between completion times in two consecutive stages for job j can be seen in Constraint set (4). Constraint sets (5) and (6) preclude the interference between the processing operations of any two jobs on a machine. Constraint set (7) determines the jobs which are processed on the same machine at stage i. Finally, Constraint set (8) shows the range of C_{ij} and force variables X_{ijk}, Y_{ilj}, and W_{ilj} to assume binary values of 0 or 1.

3. Lagrangian relaxation approach

Lagrangian relaxation approach is a mathematical programming technique to address the constrained optimization problems [23]. The Lagrangian relaxation approach relaxes the complicated problems through removing one or more constraints, especially complicated constraints, and including them in the objective function through multipliers or weights, called Lagrangian multipliers. Each Lagrangian multiplier introduces an additional term, which penalizes the generated solution that does not satisfy the relaxed constraint [24]. For this purpose, the following subsections describe different parts of the proposed LR approach to tackle the HFS problem with unrelated parallel machines at each stage.

3.1. Decomposition

According to the monolithic MILP model, presented in Section 2, only Constraint set (4) couples two successive stages. Through relaxing this complicated constraint set and including it in the objective function, the original problem can be decomposed into some subproblems, any of which is associated with one separate stage. By considering λ_{ij} as Lagrangian multipliers, the Lagrangian relaxation problem can be stated as follows:

$$LR: \ \min \ Z = \max\{C_{mj}\} + \sum_{j=1}^{n}\sum_{i=2}^{m}\lambda_{i-1,j} \left\{C_{i-1,j} - C_{ij} + \sum_{k=1}^{S_i} P_{ijk}X_{ijk}\right\}. \quad (9)$$

Subject to: Constraints (2), (3), and (5)-(8):

$$\lambda_{ij} \geq 0. \quad (10)$$

With respect to the non-negative Lagrangian multipliers (λ_{ij}), the Lagrangian dual problem can be stated as follows:

$$LD: \ \max_{\lambda_{ij}}\left[\max_{j}\{C_{mj}\} + \sum_{j=1}^{n}\sum_{i=2}^{m}\lambda_{i-1,j}(C_{i-1,j} - C_{ij}) + \sum_{j=1}^{n}\sum_{i=2}^{m}\sum_{k=1}^{S_i} P_{ijk}X_{ijk}\right]. \quad (11)$$

Subject to: Constraints (2), (3), (5)-(8):

$$\lambda_{ij} \geq 0. \quad (12)$$

Due to special structure of the objective function, it can be rewritten as:

$$\sum_{j=1}^{n}\lambda_{1j}C_{1j} + \sum_{i=2}^{m-1}\sum_{j=1}^{n}\left[(\lambda_{ij} - \lambda_{i-1,j})C_{ij} + \lambda_{i-1,j}\sum_{k=1}^{S_i} P_{ijk}X_{ijk}\right] + \sum_{j=1}^{n}\left[-\lambda_{m-1,j}\left(C_{mj} + \sum_{k=1}^{S_i} P_{mjk}X_{mjk}\right)\right] + \max_{j}\{C_{mj}\}. \quad (13)$$

By considering the objective function (Eq. (13)) and other constraint sets of the relaxed form (Eq. (12)), the Lagrangian dual problem is easily decomposed into some subproblems, each for one stage (first stage, stages $i = 2, \cdots, m-1$, and last stage). Hence, the subproblem for each stage is as follows:

$$SP_i(\lambda) = \sum_{j=1}^{n}\lambda_{ij}C_{ij} \qquad i = 1, \quad (14)$$

$$SP_i(\lambda) = \sum_{j=1}^{n}\left[(\lambda_{ij} - \lambda_{i-1,j})C_{ij} + \lambda_{i-1,j}\sum_{k=1}^{S_i} P_{ijk}X_{ijk}\right] \qquad i \neq 1, m, \quad (15)$$

$$SP_i(\lambda) = \sum_{j=1}^{n}\left[-\lambda_{i-1,j}\left(C_{ij} + \sum_{k=1}^{S_i} P_{ijk}X_{ijk}\right)\right] + \max_{j}\{C_{mj}\} \qquad i = m. \quad (16)$$

Subject to: Constraints (2), (3), (5)-(8).

For each subproblem, i is fixed and equal to stage number. According to the above, any particular subproblem can be stated as $P_m||\sum w_j c_j$, based on Pinedo triple notation, in which the weights of the jobs can be positive, zero, or negative. Since $P_m||\sum w_j c_j$ is NP-hard for $S_i \geq 2$ in the strong sense [25], it is not easy to solve with appropriate size by optimization software. Thus, we develop some proper constraints and appropriate dominance rules to reduce the complexity.

3.2. First approach: simplification of subproblems

As mentioned above, each subproblem is shown to be NP-hard; the *SPs* become harder to solve if the numbers of jobs and machines in each stage are increased. In this section, we develop some new constraints that can simplify solving the subproblems so that the sequence of the jobs at each machine can be determined easily. These constraints are inspired by Salmasi et al. [26]. For this purpose, we suppose that jobs j and l must be processed on machine k at stage i, simultaneously. There are two scenarios named $Seq1$ and $Seq2$, which can be seen in Figure 1. The only difference between these two sequences is in the positions of jobs j and l in the sequence. The completion time of the jobs before these jobs is the same, and is shown by C_A in both sequences in Figure 1.

As results, some new constraints are proposed for three groups of the subproblems. The first one is dedicated for the subproblem in the first stage, the second one is proposed for the subproblems in stages 2

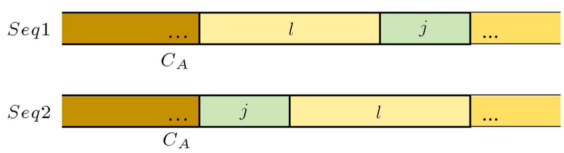


Figure 1. Two scenarios for jobs j and l.

through m-1, and finally, the last one is proposed for the subproblem in the last stage.

(a) **Simplification of $sp_1(\lambda)$**

According to the above, both jobs j and l are processed on machine k in the first stage. By assuming that the weighted completion time of $Seq2$ is greater than that of $Seq1$, Inequality (17) holds:

$$\lambda_{1l}(C_A+p_{1lk})+\lambda_{1j}(C_A+p_{1lk}+p_{1jk}) \leq \lambda_{1j}(C_A+p_{1jk})+\lambda_{1l}(C_A+p_{1jk}+p_{1lk}) \Rightarrow \frac{p_{1lk}}{\lambda_{1l}} \leq \frac{p_{1jk}}{\lambda_{1j}}. \tag{17}$$

Thus, constraint(s) must be incorporated into $sp_1(\lambda)$ with respect to inequality (17). By considering W_{1lj} and Y_{1lj}, we incorporate the constraint sets below into $sp_1(\lambda)$ to determine the sequence of jobs on each machine in the first stage:

$$2Y_{1lj}\left(\frac{p_{1jk}}{\lambda_{1j}}-\frac{p_{1lk}}{\lambda_{1l}}\right) \geq \left(\frac{p_{1jk}}{\lambda_{1j}}-\frac{p_{1lk}}{\lambda_{1l}}\right)W_{1lj} \quad j,l=1,2,\cdots,n \quad \& \quad (j,l)\in R, \tag{18}$$

$$Y_{1lj} \leq W_{1lj} \qquad j,l=1,2,\cdots,n \quad \& \quad (j,l)\in R. \tag{19}$$

If job l is processed before job j, the value in the parenthesis is positive in constraint set (18). Thus, if the value of W_{1lj} is equal to 1, then $Y_{1lj} \geq \frac{1}{2}$ and Y_{1lj} is also equal to 1. It shows that job l is processed before job j in the first stage. On the other side, if the value in the parenthesis is negative, then $Y_{1lj} \leq \frac{1}{2}$ and it leads Y_{1lj} to take a value equal to zero. With respect to Constraint (19), if both jobs l and j are processed on the same machine in the first stage ($W_{1lj}=1$), then Y_{1lj} can take value equal to 1;

(b) **Simplification of $sp_i(\lambda)$; $i=2,\cdots,m-1$**

Similar to the above manner and considering $Seq1$ and $Seq2$, we will have:

$$(\lambda_{il}-\lambda_{i-1,l})(C_A+p_{ilk})+\lambda_{i-1,l}p_{ilk} + (\lambda_{ij}-\lambda_{i-1,j})(C_A+p_{ilk}+p_{ijk}) + \lambda_{i-1,j}p_{ijk} \leq (\lambda_{ij}-\lambda_{i-1,j})(C_A+p_{ijk}) + \lambda_{i-1,j}p_{ijk} + (\lambda_{il}-\lambda_{i-1,l})(C_A+p_{ijk}+p_{ilk}) + \lambda_{i-1,l}p_{ilk} \Rightarrow \frac{p_{ijk}}{\lambda_{ij}-\lambda_{i-1,j}} \geq \frac{p_{ilk}}{\lambda_{il}-\lambda_{i-1,l}}. \tag{20}$$

Thus, we introduce Constraint sets (21) and (22) to obtain the sequence of the jobs in stage 2 through $m-1$ as follows:

$$2Y_{ilj}\left(\frac{p_{ijk}}{\lambda_{ij}-\lambda_{i-1,j}}-\frac{p_{ilk}}{\lambda_{il}-\lambda_{i-1,l}}\right) \geq \left(\frac{p_{ijk}}{\lambda_{ij}-\lambda_{i-1,j}}-\frac{p_{ilk}}{\lambda_{il}-\lambda_{i-1,l}}\right)W_{ilj} \quad j,l=1,2,\cdots,n \quad \& \quad (j,l)\in R \quad i=2,\cdots,m-1, \quad k=1,2,\cdots,S_i, \tag{21}$$

$$Y_{ilj} \leq W_{ilj} \qquad j,l=1,2,\cdots,n \quad \& \quad (j,l)\in R \quad i=2,\cdots,m-1, \quad k=1,2,\cdots,S_i. \tag{22}$$

The explanation for the above constraints is similar to the one presented for Constraint sets (18) and (19).

(c) **Simplification of $sp_m(\lambda)$**

In the same manner, Inequality (23) holds:

$$-\lambda_{m-1,l}(C_A+p_{mlk}+p_{mlk}) - \lambda_{m-1,j}(C_A+p_{mlk}+p_{mlk}+p_{mjk}) + C_{\max} \leq -\lambda_{m-1,j}(C_A+p_{mjk}+p_{mjk}) - \lambda_{m-1,j}(C_A+p_{mjk}+p_{mjk}+p_{mlk}) + C_{\max} \Rightarrow \frac{p_{mjk}}{\lambda_{m-1,j}} \geq \frac{p_{mlk}}{\lambda_{m-1,l}}. \tag{23}$$

Therefore, the following constraint sets are incorporated into the model to determine the sequence of the jobs at the last stage:

$$2Y_{mlj}\left(\frac{p_{mjk}}{\lambda_{m-1,j}}-\frac{p_{mlk}}{\lambda_{m-1,l}}\right) \geq \left(\frac{p_{mjk}}{\lambda_{m-1,j}}-\frac{p_{mlk}}{\lambda_{m-1,l}}\right)W_{mlj} \quad j,l=1,2,\cdots,n \quad \& \quad (j,l)\in R, \tag{24}$$

$$Y_{mlj} \leq W_{mlj} \qquad j,l=1,2,\cdots,n \quad \& \quad (j,l)\in R. \tag{25}$$

3.3. Second approach: Dominance rules

In the second approach, we develop some dominance rules to easily solve the subproblems. For this purpose, we apply Lemma 1, which was developed by Tang et al. [14].

Lemma 1. Let ss_p, ss_0, and ss_n denote three subsets of the jobs on a particular machine at stage i, containing the jobs with positive, zero, and negative weights, respectively. Then, in the optimal solution, these subsets must be scheduled on a machine in the sequence of ss_p, ss_0, and ss_n, respectively; the jobs within ss_p and ss_n must be sequenced in WSPT order; the jobs in ss_0 can be sequenced randomly.

According to the objective function and structure of the constraints, we will develop some dominance rules for each subproblem as follows:

1. $sp_1(\lambda)$: According to the objective function of $sp_1(\lambda)$ (Eq. (14)), the subproblem is converted to parallel machine scheduling problem with weighted (with positive value) completion time minimization. Therefore, with respect to Lemma 1, if the jobs on each machine are sequenced based on WSPT order, we expect to minimize the objective function. Therefore, we consider WSPT rule to determine the sequence of jobs on each machine in the first stage, easily;
2. $sp_i(\lambda); i = 2, \cdots, m-1$: Similar to $sp_1(\lambda)$, the production system of the subproblem i, $i = 2, \cdots, m-1$, is parallel machine. The objective function contains two sections: The first one is $\sum_{j=1}^{n}(\lambda_{ij} - \lambda_{i-1,j})C_{ij}$ that minimizes the weighted completion time with the weight of $(\lambda_{ij} - \lambda_{i-1,j})$ (with positive, negative, and zero values). The second one is $\sum_{j=1}^{n}\lambda_{i-1,j}\sum_{k=1}^{S_i}p_{ijk}X_{ijk}$ and equal to the summation of the product of the Lagrangian multipliers and processing time of each job on the assigned machine in each stage. Also, $\sum_{j=1}^{n}\sum_{k=1}^{S_i}p_{ijk}X_{ijk}$ shows the summation of completion times of the last jobs on each machine; if the weights of these jobs ($\lambda_{i-1,j} \geq 0$) are added to the first section, the weight of the jobs increases. Thus, the first set of jobs with the weight of $\lambda_{ij} - \lambda_{i-1,j}$ cannot be processed as the last job on each machine and the second set of jobs with the weight of λ_{ij} is the last job on each machine. With respect to the above discussion, we can generate the following lemma.

Lemma 2. The following constraints between jobs j (last job in sequence) and l (other jobs) at stage $i = 2, \cdots, m-1$ hold:

$$2Y_{ilj}\left(\frac{p_{ijk}}{\lambda_{ij}} - \frac{p_{ilk}}{\lambda_{il} - \lambda_{i-1,l}}\right) \geq \left(\frac{p_{ijk}}{\lambda_{ij}} - \frac{p_{ilk}}{\lambda_{il} - \lambda_{i-1,l}}\right)W_{ilj}$$

$$j, l = 1, 2, \cdots, n \quad \& \quad (j,l) \in R$$

$$i = 2, \cdots, m-1, \qquad k = 1, 2, \cdots, S_i, \tag{26}$$

$$Y_{ilj} \leq W_{ilj} \qquad j, l = 1, 2, \cdots, n \quad \& \quad (j,l) \in R,$$

$$i = 2, \cdots, m-1. \tag{27}$$

Proof. According to the description of $sp_i(\lambda)$, $i = 2, \cdots, m-1$, the weights of the last job and other jobs in the sequence are λ_{ij} and $\lambda_{il} - \lambda_{i-1,l}$, respectively. Because of nonnegative Lagrangian multipliers and Lemma 1, the jobs can take positive weights. Thus, Inequality (28) holds:

$$\frac{p_{ilk}}{\lambda_{il} - \lambda_{i-1,l}} \leq \frac{p_{ijk}}{\lambda_{ij}}$$

$$j, l = 1, 2, \cdots, n \quad \& \quad (j,l) \in R$$

$$i = 2, \cdots, m-1, \qquad k = 1, 2, \cdots, S_i. \tag{28}$$

If job j is the last job in the sequence, Inequality (28) holds. Thus, if W_{ilj} takes the value equal to 1 in Constraint set (26) and the value in the parenthesis is positive, then $Y_{ilj} \geq \frac{1}{2}$ and Y_{ilj} is equal to 1. It shows that job j cannot be processed before job l ($j = 1, 2, \cdots, n$ & $j \neq l$) and it can impose job j as the last job in the sequence. On the other side, if the value in the parenthesis is negative, then $Y_{ilj} \leq \frac{1}{2}$ and it forces Y_{ilj} to take a value equal to zero. Furthermore, the description of Constraint set (27) is similar to the one presented for Constraint set (22).

3. $sp_m(\lambda)$: The objective function for subproblem m contains three terms. The first one is equal to $\sum_{j=1}^{n} -\lambda_{i-1,j}C_{ij}$ with the weight of $-\lambda_{i-1,j}$. The second term represents the summation of completion times of the last jobs on each machine and the last term shows the last job, which has the maximum completion time. Thus, we can assign the jobs in the last stage to three subsets as in Table 1.

According to the above, we can develop the following lemma to easily solve the subproblem m.

Lemma 3: The following constraint between jobs j (last job in sequence) and l (other jobs) in stage m holds:

$$2Y_{mlj}\left(\frac{p_{mlk}}{2\lambda_{m-1,l}} - \frac{p_{mjk}}{\lambda_{m-1,j}}\right) \geq \left(\frac{p_{mlk}}{2\lambda_{m-1,l}} - \frac{p_{mjk}}{\lambda_{m-1,j}}\right)W_{mlj},$$

$$j, l = 1, 2, \cdots, n \quad \& \quad (j,l) \in R$$

$$k = 1, 2, \cdots, S_m, \tag{29}$$

$$Y_{mlj} \leq W_{mlj} \qquad j, l = 1, 2, \cdots, n \quad \& \quad (j,l) \in R. \tag{30}$$

Table 1. Different subsets of jobs.

Subset	Weight	Position in the sequence
1	$-\lambda_{i-1,j}$	Cannot be the last job
2	$-2\lambda_{i-1,j}$	Last jobs
3	$1-2\lambda_{i-1,j}$	Last job (jobs) with the maximum completion time

Proof. According to the discussion about subproblem m, the weights of the last job (j) and other jobs (l) in the sequence are $-2\lambda_{m-1,j}$ and $-\lambda_{m-1,l}$, respectively. Therefore, based on WSPT rule, Inequality (31) holds:

$$\frac{p_{mlk}}{-\lambda_{m-1,l}} \leq \frac{p_{mjk}}{-2\lambda_{m-1,j}} \Rightarrow \frac{p_{mlk}}{\lambda_{m-1,l}} \geq \frac{p_{mjk}}{2\lambda_{m-1,j}}$$

$$j,l=1,2,\cdots,n \quad \& \quad (j,l)\in R$$

$$k=1,2,\cdots,S_m. \tag{31}$$

Similar to Lemma 2, if Inequality (31) holds, then the value in the parenthesis is positive and $Y_{mlj} \geq \frac{1}{2}$; thus, Y_{ilj} is equal to 1 and job l must be processed before job j in the last stage. On the other side, if the value in the parenthesis is negative, $Y_{mlj} \leq \frac{1}{2}$ and Constraint (29) forces Y_{ilj} to take value equal to zero.

3.4. Construction of the feasible solution

With respect to relaxing of the precedence constraints (Constraint set (4)), in general, the solution obtained by every subproblem is infeasible. To tackle this difficulty and achieve a feasible solution, a heuristic method is presented here to establish the precedence constraints. At the first step, the schedule obtained by the dual problem is considered as the initial schedule. Then, in each stage, the sequence of the jobs is generated by three parameters of completion time in the first stage, average process time in other stages, and a random number. At last, each job is assigned to the entire machines (available and unavailable) and the machine with the smallest completion time is selected. The steps of the proposed heuristic are presented in the following:

1. The schedules generated by the dual problem are considered as the initial schedule.
2. For each stage, the sequence is created in the non-decreasing order of:

$$C_{1j}e^{-\frac{\Omega}{\sum_{i=1}^{n}\bar{P}_{ij}}},$$

$$\bar{P}_{ij}=\frac{\sum_{k}^{S_i}P_{ijk}}{S_i},$$

$$\Omega\in U[0,1].$$

3. The jobs are assigned to all the available and unavailable machines in stage 1 through m and the machine with the lowest completion time is selected.
4. The end.

3.5. Updating the Lagrangian multipliers

A subgradient method [27] is adopted for updating λ_{ij} in each iteration of the LR algorithm. In this way, Eq. (32) is applied to update the Lagrangian multipliers in iteration t:

$$\lambda_{ij}^{t+1}=\lambda_{ij}^{t}+h^{t}G_{ij}^{t}, \tag{32}$$

in which, h^t and G_{ij}^t are the step size and subgradient, respectively. The step size is calculated by $h^t=\frac{1}{(a+bt)|G_{ij}^t|}$ (a and b are scalar) and the subgradinet is given by:

$$G_{ij}^{t}=C_{i-1,j}^{t}+C_{ij}^{t}+\sum_{k=1}^{S_i}P_{ijk}x_{ijk}^{t}.$$

Figure 2 illustrates the pseudocode of the proposed Lagrangian relaxation approach for the HFS problem with unrelated parallel machines, in which ε is a threshold value, and MaxIter and MaxCPU are the maximum number of iterations and maximum CPU time, respectively.

Initialization
Set $t=1$, Maxiter, MaxCPU, ε, λ_{ij}^{1} (initial vector of Lagrangian multipliers)
While $|\lambda_{ij}^{t+1}-\lambda_{ij}^{r}|\geq\varepsilon$ & $t\leq Maxiter(=500)$ & $CPUTime\leq MaxCPU(=3600$ sec.). Do
Solve $sp_1(\lambda)$
Solve $sp_2(\lambda)$; $i=2,\cdots,m-1$
Solve $sp_m(\lambda)$
Apply heuristic to convert the infeasible schedule into a feasible schedule
Update λ_{ij}^{t}
End while
Report optimal gap $\left(\frac{UB-LB}{LB}\right)$

Figure 2. Pseudo-code of the LR algorithm.

4. Numerical experiments

This section is devoted to evaluating the proposed approaches to solving the HFS problem with regard to some test problems. Due to NP-hardness of the HFS problem with unrelated parallel machines, the monolithic MILP model can obtain the optimal solution to small-size problems. In this section, first, numerical experiments are conducted to investigate the performance of the monolithic MILP model and the proposed LR approaches for small-size test problems. Furthermore, we compare the performances of the subproblem solving approach, simplification of subproblems, and dominance rules for large-size test problems.

The test problems in this experiment are divided into two categories: small- and large-size ones. For the small-size problems, the processing times, number of jobs, number of machines in each stage, and number of stages are uniformly distributed in the intervals $[5, 50]$, $[5, 10]$, $[2, 4]$, and $[2, 5]$, respectively. On the other side, the processing times, number of jobs, number of machines in each stage, and number of stages are uniformly distributed in the intervals $[5, 50]$, $[20, 50]$, $[2, 4]$, and $[2, 5]$ for large-size problems.

The monolithic MILP model and the LR algorithms are implemented in GAMS/CPLEX and tested on a computer with 2.8 GHz CPU and 2.0 GB of RAM. A time limit of 3600 s is imposed for running the monolithic MILP model.

4.1. Result analysis for small-size problems

This section compares the LR and monolithic MILP models with regard to small-size problems. For this purpose, 20 small-size test problems are generated and each test problem is run 10 times with different processing times. To compare the monolithic MILP model and LR approach, we apply two different criteria, namely CPU time (CPU) and Optimal Gap (OG), to assess the quality of the LR approaches. The OG is calculated as follows:

$$OG = \frac{UB - LB}{LB} \times 100. \tag{33}$$

The results of the experiment are shown in Table 2.

Based on Table 2, the LR approach improved with simplification of subproblems significantly outperforms dominance rules in all cases. According to OG, the average result of dominance rules is 1.49%, while that of simplification of subproblems is 1.77%. Because of applying the heuristic approach in calculating the upper bound, the solution with OG of less than 3% is

Table 2. Testing results of LR approaches and monolithic MILP model for small-size problems.

Test problem	Problem structure	MILP		Dominance rules		Simplification of subproblems	
		CPU	OG	CPU	OG	CPU	OG
1	10J3S(232)	3600	11.5	65.4	2.76	88.7	2.38
2	10J3S(432)	3600	14.7	72.6	2.84	98.5	2.41
3	10J2S(23)	621	0	44.8	2.28	61.1	1.98
4	10J2S(43)	425	0	46.6	2.18	59	2.02
5	8J4S(2322)	119	0	55.7	2.06	66.4	1.88
6	8J4S(2324)	120	0	56.9	2.15	70.6	1.96
7	8J3S(232)	54	0	37.4	2.11	56.4	1.86
8	8J3S(432)	26	0	39.2	2.08	61.7	1.85
9	8J2S(23)	9	0	23.6	1.81	36	1.69
10	8J2S(43)	10	0	28.1	1.94	41.4	1.81
11	6J4S(2322)	3.3	0	30.9	1.73	43.8	1.32
12	6J4S(2324)	4.7	0	36.3	1.69	49.1	1.44
13	6J3S(232)	3.1	0	23.1	1.48	30.5	1.12
14	6J3S(432)	2.4	0	27.5	1.51	41.6	1.18
15	6J2S(43)	0.97	0	8.3	1.28	15.6	0.99
16	6J2S(23)	1.26	0	11.8	1.33	23.5	1.02
17	5J4S(2322)	1.26	0	16.6	1.25	25.4	0.78
18	5J4S(2324)	0.98	0	20	1.19	29.1	0.84
19	5J3S(232)	0.70	0	3.7	0.80	8.4	0.53
20	5J3S(432)	0.38	0	8.3	0.98	12.3	0.69

very close to optimal [10]. In terms of CPU time, the LR with dominance rules can achieve the final solution in 32.8 s, but the simplification of subproblems takes about 45.9 s in average to solve the test problems.

4.2. Analysis of results for large-size problems

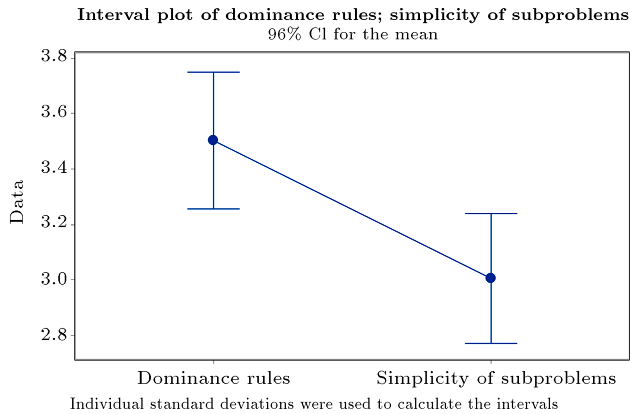
The parameters in this section are applied to generate the large-size problem to compare the performances of the proposed approaches for HFS with unrelated parallel machines. The number of jobs is extended to 20 and 50 and other parameters are same as those in the last section.

Due to NP-hardness of HFS with unrelated parallel machines, the monolithic MILP model cannot achieve the optimal solution in reasonable time. Thus, the test problems in this section only compare the performances of the proposed LR approaches with each other. The experimental results are summarized in Table 3.

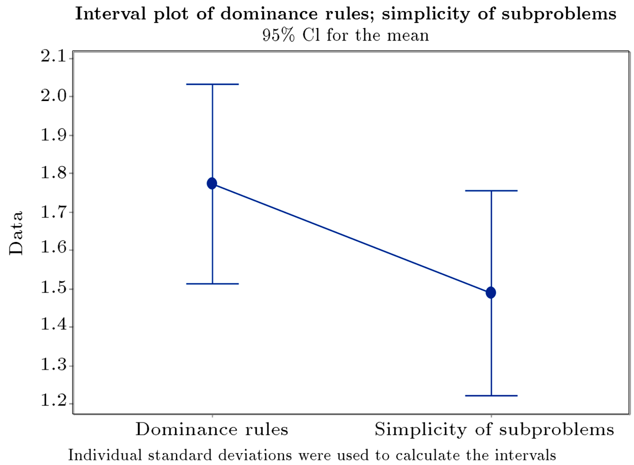
The problems in Table 3 are presented to compare the performances of the proposed LR approaches for large size. According to testing results, the average OGs of dominance rules and simplification of subproblems equal 3.51% and 3.00%, respectively. Also, the average CPU time is 220.09 s for dominance rules and 248.15 s for simplification of subproblems. According to the results, conclusions similar to those for small-size problems can be driven for these larger sized problems. That is, on average, the simplification of subproblems approach generates better solutions than dominance rules do in a much longer computation time.

Table 3. Testing results of LR approaches for large-size problems.

Test problem	Problem structure	Dominance rules		Simplification of subproblems	
		CPU	OG	CPU	OG
1	20J2S(23)	97.7	2.83	114.3	2.38
2	20J2S(33)	93.1	2.82	109.7	2.29
3	20J2S(43)	102.2	2.88	121.5	2.42
4	20J2S(44)	99.6	2.93	111.6	2.43
5	20J3S(232)	126.1	3.07	143.8	2.69
6	20J3S(432)	130.2	3.16	153.6	2.77
7	20J3S(434)	127.7	3.11	149.4	2.71
8	20J4S(2322)	150.2	3.23	165.2	2.78
9	20J4S(2324)	153.7	3.20	175.2	2.66
10	20J4S(4232)	161.6	3.21	193.9	2.89
11	50J2S(23)	259.4	3.54	303.5	2.96
12	50J2S(33)	274.0	3.56	328.8	2.89
13	50J2S(43)	270.4	3.78	299.0	3.06
14	50J2S(44)	285.3	3.63	315.2	3.13
15	50J3S(232)	310.3	4.07	359.9	3.61
16	50J3S(432)	325.9	4.01	371.5	3.51
17	50J3S(434)	318.0	4.08	349.6	3.59
18	50J4S(2322)	351.5	4.28	379.6	3.72
19	50J4S(2324)	390.0	4.24	412.9	3.81
20	50J4S(4232)	374.9	4.39	404.9	3.77



(a)



(b)

Figure 3. Interval plot of the proposed LR approaches: (a) Large-size problems and (b) small-size problems.

For more scrutiny and as a formal test, we apply 95% confidence interval for the average OG to compare the proposed LR approaches. The confidence intervals are illustrated in Figure 3.

In Figure 3, we can observe that the simplification of subproblems statistically outperforms the dominance rules in large-size problems; In addition, there is no significant difference between the proposed approaches in small-size problems.

Thus, we can conclude that the superiority of the simplification of subproblems approach becomes more significant with increase in the size of test problems.

4.3. Analysis of test problem parameters

In order to analyze the effects of number of jobs and number of stages in the test problems on the quality of the proposed approaches, a two-way ANOVA is

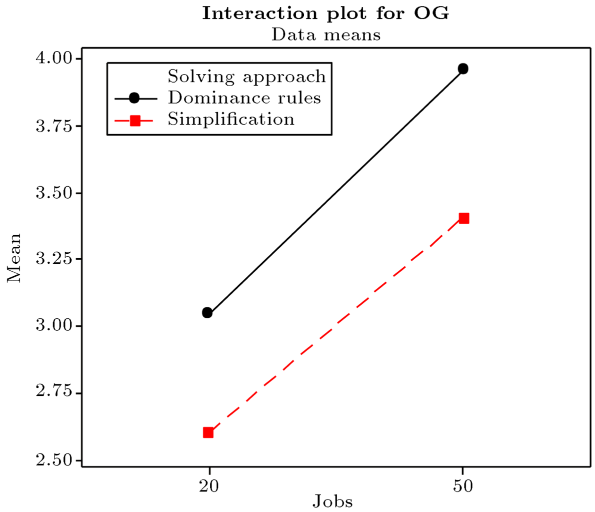


Figure 4. Interaction between the proposed approaches and number of jobs.

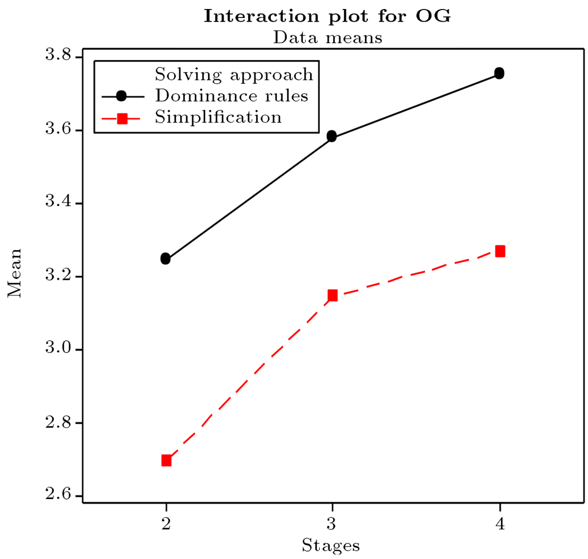


Figure 5. Interaction between the proposed approaches and number of stages.

applied in this research. For this purpose, mean OG plot for interaction between the proposed approaches, and number of jobs and number of stages are shown in Figures 4 and 5, respectively.

As can be seen in both Figures 4 and 5, the simplification of subproblems works better than dominance rules in all the cases. Furthermore, by increasing the number of jobs, the difference between two proposed approaches is significantly increased. Also, if the number of stages is set to level 4, we can see the biggest difference between the two proposed approaches.

5. Conclusion

In this research, a well-known real-world problem, namely HFS scheduling problem with unrelated parallel machines in each stage, has been investigated. Since the problem has proven to be NP-hard, the monolithic MILP cannot achieve the optimal solutions in reasonable time. Thus, we proposed a Lagrangian Relaxation (LR) method to solve the problem. Furthermore, to solve the generated subproblems in LR, we developed two approaches, namely dominance rules and simplification of subproblems, to reduce the complexity. The results of the experiments show that the simplification of subproblems has better performance than the dominance rules in solving the small- and large-size problems.

Future research can be conducted to consider other characteristics of the HFS environments, such as availability constraints, sequence-dependent setup time (cost), group scheduling, and identical machines. In addition, other exact approaches such as branch-and-cut, benders decomposition, and branch-and-price can be used to solve the HFS problem with unrelated parallel machines.

References

1. Liao, C.J., Tjandradjaja, E., and Chung, T.P. "An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem", *Appl. Soft Comput.*, **12**, pp. 1755-1764 (2012).

2. Agnetis, L., Pacifici, A., Rossi, F., Lucertini, M., Nicoletti, S., and Nicolo, F. "Scheduling of flexible flow shop in an automobile assembly plant", *Eur. J. Oper. Res.*, **97**(2), pp. 348- 362 (1997).

3. Adler, L., Fraiman, N., Kobacker, E., Pinedo, M., Plotnicoff, J.C., and Wu, T.P. "BPSS: a scheduling support system for the packaging industry", *Oper. Res*, **41**, pp. 641-648 (1993).

4. Shahvari, O., Salmasi, N., Logendran, L., and Abbasi, B. "An efficient tabu search algorithm for flexible flow shop sequence dependent group scheduling problems", *Int. J. Prod. Res.*, **50**, 15, pp. 4237-4254 (2012).

5. Moursli, O. and Pochet, Y. "A branch and- bound algorithm for the hybrid flowshop", *Int. J. Prod. Econ.*, **64**, pp. 113-25 (2000).

6. Liu, C.Y. and Chang, S.C. "Scheduling flexible flow shops with sequence-dependent setup effects", *IEEE Transaction on Robotics and Automation*, **16**(4), pp. 408-419 (2000).

7. Fattahi, P., Hosseini, S.M.H., Jolai, F., and Tavakkoli-Moghaddam, R.A. "Branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations", *Appl. Math. Model*, **38**, pp. 119-134 (2014).

8. Riane, F., Artibas, A., and Elmaghraby, S.E. "A hybrid three stage flow shop problem: efficient heuristics to minimize makespan", *Eur. J. Oper. Res.*, **109**, pp. 321-329 (1998).

9. Linn, R. and Zhang, W. "Hybrid flow shop scheduling: a survey", *Comput. Ind. Eng.*, **37**, pp. 57-61 (1999).

10. Marichelvam, M.K., Prabaharan, T., Yang, X., and Geetha, M. "Solving hybrid flow shop scheduling problems using bat algorithm", *Int. J. Logist Econ Glob*, **5**(1), pp. 15-29 (2013).

11. Jolai, F., Asefi, H., Rabiee, M., and Ramezani, P. "Bi-objective simulated annealing approaches for no-wait two-stage flexible flow shop scheduling problem", *Scientia Iranica*, **20**(3), pp. 861-872 (2013).

12. Marichelvam, M.K., Prabaharanb, T., and Yang, X.S. "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan", *Appl. Soft Comput.*, **19**, pp. 93-101 (2014).

13. Jolai, F., Tavakkoli-Moghaddam, R., Rabiee, M., and Gheisariha, E. "An enhanced invasive weed optimization for makespan minimization in a flexible fowshop scheduling problem", *Scientia Iranica*, **21**(3), pp. 1007-1020 (2014).

14. Tang, L., Xuan, H., and Liu, J. "A new Lagrangian relaxation algorithm for hybrid flowshop scheduling to minimize total weighted completion time", *Comput. Oper. Res.*, **33**, pp. 3344-3359 (2006).

15. Liu, G.D. and Luh, P.B. "Scheduling permutation flow shops using the Lagrangian relaxation technique", *Ann. Oper. Res.*, **70**, pp. 171-89 (1997).

16. Luh, P.B. and Hoitomt, D.J. "Scheduling of manufacturing systems using the Lagrangian relaxation technique", *IEEE Transaction on Automatic Control*, **38**(7), pp. 1066-1079 (1993).

17. Chang, S.C., Liao, D.Y., and Hsieh, F.S. "Scheduling flexible flow shops of no setup cost by a Lagrangian relaxation and network flow approach", *Interactional Conference on Robotics and Automation* (1991).

18. Irohara, T. "Lagrangian relaxation algorithms for hybrid flow-shop scheduling problems with limited buffers", *Biomed. Fuzzy Hum. Sci.*, **15**(1), pp. 21-28 (2010).

19. Sun, X. and Noble, J.S. "An approach to job shop scheduling with sequence- dependent setups", *J. Manuf. Syst.*, **18**(6), 416-430 (1999).

20. Emami, S., Sabbagh, M., and Moslehi, Gh. "A Lagrangian relaxation algorithm for order acceptance and scheduling problem: a globalized robust optimisation approach", *Int. J. Comput. Integ. Manuf.*, **29**(5), pp. 535-560 (2016). DOI: 10.1080/0951192X.2015.1068452.

21. Shishebori, D., Yousefi Babadi, A., and Noormohammadzadeh, Z. "A Lagrangian relaxation approach to fuzzy robust multi-objective facility location network design problem", *Scientia Iranica*, **25**(3), pp. 1750-1767 (2018).

22. Nahavandi, N. and Asadi Gangraj, E. "A new lower bound foe flexible flow shop scheduling problem with unrelated parallel machines", *Int. J. Ind. Eng. Prod. Res.*, **25**(1), pp. 65-70 (2014).

23. Augusto, V., Xie, X., and Perdomo, V. "Operating theatre scheduling using Lagrangian relaxation", *Eur. J. Ind. Eng.*, **2**(2), pp. 172-189 (2008).

24. Smith, J.C. and Sonuc, S.B. "An introduction to integer and large-scale linear optimization, wireless network design", *Int. Ser. in Oper. Res. Manag. Sci.*, **158**, pp. 65-97 (2011).

25. Lenstra, J.K., Rinnooy Kan, A.H.G., and Bruker, P. "Complexity of machine scheduling problems", *Annals of Discrete Mathematics*, **1**, pp. 343-62 (1977).

26. Salmasi N., Logendran, R., and Skandari, M.R. "Total flow time minimization in a flow shop sequence-dependent group scheduling problem", *Computers & Operations Research*, **37**, pp. 199-212 (2010).

27. Polyak, B.T. "Introduction to optimization", *Optimization Software*, Inc., New York (1987).

Biography

Ebrahim Asadi-Gangraj received his BS degree in Industrial Engineering from Isfahan University of Technology, Isfahan, Iran, in 2005; MS degree in Industrial Engineering from the Tarbiat Modares University in 2008; and PhD degree in Industrial Engineering from Tarbiat Modares University, Iran, in 2014. He is currently Assistant Professor of Industrial Engineering at Babol Noshirvani University of Technology, Babol, Iran. His research interests include applied operations research, sequencing and scheduling, production planning, and supply chain management.