



A Hybrid Flow Shop Scheduling Problem

Ayşegül Eda Özen, Gülce Çini, Merve Çamlıca, Nilay Çınar,
Hasan Bahtiyar Soydan, Levent Kandiller, and Hande Öztop^(✉)

Department of Industrial Engineering, Yaşar University, İzmir, Turkey
ayseguleda.ozen@gmail.com, gulce.cini@gmail.com,
mervecmlca@gmail.com, nilayycinar@gmail.com,
hasanbahtiyar96@gmail.com, {levent.kandiller,
hande.oztop}@yasar.edu.tr

Abstract. Hybrid flow shop environment generally refers to the flow shop with multiple parallel machines per stage. Hybrid flow shop scheduling problem (HFSP) is a complex combinatorial optimization problem that came across in many real-life problems. In this study, a real-life HFSP of a lubricant company is considered, where the aim is to minimize total weighted completion time of the jobs. Apart from classical HFSPs, the studied problem has additional constraints such as machine eligibility, sequence-dependent setup times and machine capacities. Due to the additional constraints in the system, a novel mixed integer linear programming model is proposed for the studied HFSP with three stages. As the problem is NP-hard, two constructive heuristic algorithms and an improvement heuristic algorithm are also developed. The performance of the proposed heuristic algorithms is evaluated by comparisons with the optimal results obtained from the mathematical model. The extensive computational results show that proposed heuristic algorithms find near optimal results in reasonable computational times. Sensitivity analysis is also performed for the weight parameter of the problem, which indicates that the proposed heuristic algorithms also perform very well for different weight parameter values. Finally, the proposed heuristic algorithms are integrated into a user-friendly decision support system using Microsoft Excel VBA interface to provide an efficient scheduling tool for the company.

Keywords: Hybrid flow shop scheduling · Total weighted completion time · Sequence-dependent setup times · Machine eligibility · Heuristic algorithm

1 Introduction

An efficient and applicable scheduling system is very important for the companies in order to decrease their costs by improving their production processes; provide accurate and realistic due dates for their customers and use existing resources more efficiently. This paper considers a real-life hybrid flow shop scheduling problem (HFSP) of a lubricant company in İzmir, Turkey. The company produces three different types of main products as automotive lubricants, industrial lubricants and auto-care products. These products have wide variety and each product has a different bill of materials, which makes company's production planning more challenging. Currently, in the

company, production planning is based on estimations and experience of the planners. As the current production planning does not follow a standard and efficient methodology, capacity gaps and financial losses may occur in the system. Therefore, the focus of this study is to minimize total weighted completion time of the jobs in a hybrid flow shop with machine eligibility, sequence-dependent setup times and machine capacity restrictions. Creating an efficient scheduling tool for the company is also one of the main purposes of this study.

Ruiz and Vázquez-Rodríguez [1] presented a detailed literature review on exact, heuristic and metaheuristic methods that have been proposed for the HFSP. They discuss solution approaches for various variants of the HFSP, considering different assumptions, constraints and objective functions. Another extensive literature review is also provided by Ribas et al. [2]. In the literature, sequence-dependent setup times and machine eligibility constraints are studied in several studies [3–9]. The makespan minimization for two-stage hybrid flow shop with dedicated machines and additional constraints is studied by Chikhi and Abbas [3]. Lin and Liao [4] proposed a heuristic algorithm for a real label sticker manufacturing problem, which corresponds to a two-stage HFSP with sequence-dependent setup times and machine eligibility. Ruiz et al. [5] have studied a complex HFSP in the ceramic tile industry, which includes many additional restrictions such as unrelated parallel machines, sequence-dependent setup times, machine eligibility and machine release dates. A genetic algorithm was proposed to minimize makespan for the HFSP with unrelated parallel machines, sequence-dependent setup times and machine eligibility by Ruiz and Maroto [6]. Yaurima et al. [7] also presented a genetic algorithm for the HFSP with unrelated machines, sequence-dependent setup times, availability constraints and limited buffers. Later, Pan et al. [8] proposed nine heuristic algorithms to minimize the makespan for the HFSP with sequence-dependent setup times. Yu et al. [9] also developed a genetic algorithm for the HFSP with unrelated machines and machine eligibility restrictions to minimize the total tardiness. Recently, a mathematical model and several metaheuristic algorithms were proposed for the HFSP considering makespan and total flow time criteria by Öztop et al. [10].

In this study, a novel mixed integer linear programming (MILP) model is proposed for the real-life 3-stage HFSP due to the additional constraints in the system. As the HFSP is known to be NP-hard [11], two constructive heuristic algorithms and an improvement heuristic algorithm are also proposed for the problem, based on well-known Shortest Processing Time (SPT) rule. The performance of the proposed heuristic algorithms is evaluated by comparisons with the optimal results obtained from the mathematical model on 50 real instances. A user-friendly decision support system has also been created by integrating the proposed heuristic algorithms.

The rest of the paper is organized as follows. The formal problem definition is provided in Sect. 2. In Sect. 3, the proposed MILP model and heuristic algorithms are presented, respectively. Computational results of the proposed solution approaches are presented in Sect. 4. Section 5 explains the developed decision support system. Finally, Sect. 6 provides the conclusions and future suggestions.

2 Problem Definition

The hybrid flow shop, which is also called as flexible flow shop, is a special variant of the generalized flow shop. In the standard HFSP, a set of n jobs must be processed on m ($m > 1$) serial stages following the same route, where each stage includes at least one parallel identical machines and each job must be assigned to the one of these machines in each stage. Note that, at least one of the stages include more than one machine. As mentioned in the previous section, we consider the real-life HFSP of a lubricant company, which has many additional constraints.

In the company, the production process consists of three main stages as blending, dropping and filling, respectively. In the first stage, base oil and additives are blended in the eligible blending machines based on blending codes of the products. As ingredients of the products are different for each blending code, there are several machine groups based on blending codes. Note that, capacities of the machines are different within each machine group. After the blending stage, samples of blended products are sent to the quality control process and the approved products proceed to the next stage, i.e., dropping stage. Similar to the blending stage, machine eligibility and capacity limitations also exist in the dropping stage. After the dropping stage, the ready products for the filling process are transferred to the filling stage via pipelines. In the filling stage, there are 3 different filling machines according to the packaging material type. Note that, during the transition from the dropping machines to the filling machines, pipe cleaning process is needed between different types of the products. Furthermore, mould change process is needed between different sizes of packaging materials in each filling machine.

As mentioned above, in blending and dropping stages, products are assigned to the eligible machines according to their blending codes. There are also machine capacity constraints in these two stages. In the filling stage, each product is assigned to an eligible filling machine according to its packaging material type. Note that, in the filling stage, there are also sequence-dependent setup times between different products due to the exchange of mould and pipeline cleaning processes. These additional constraints differentiate the studied problem from the standard HFSP, further complicating the structure. In the studied HFSP, it is assumed that all machines are available initially and there is no initial setup time for the jobs. It is also assumed that, the quality control durations between stages are included in the processing times of the jobs in the corresponding stages. Consequently, the aim of the problem is to minimize total weighted completion time of the jobs, where each job has a priority weight based on its profit. In order to determine the weights of the jobs, detailed ABC analyses are performed based on profits of the jobs, where the products are classified in three main groups, as A, B and C according to their profits.

3 Solution Methodology

3.1 Mathematical Model

The proposed MILP model is explained below for the studied real-life HFSP with machine eligibility, sequence-dependent setup times and machine capacity constraints, where the objective is to minimize total weighted completion time of the jobs.

Sets and Indices

- i, q : index of job
- j : index of blending code
- k : index of machine
- I : Set of jobs including the dummy job, $I = \{0, \dots, n\}$
- J : Set of blending codes
- F : Set of machines in filling stage
- B_j : Set of machines for blending code j in blending stage
- D_j : Set of machines for blending code j in dropping stage
- F_i : Set of machines for job i in filling stage

Parameters

- w_i = Weight of job i
- px_i = Processing time of job i in blending stage
- py_i = Processing time of job i in dropping stage
- pz_i = Processing time of job i in filling stage
- s_{iq} = Setup time when job q is processed immediately after job i
- N_k = Capacity of machine k in blending stage
- A_k = Capacity of machine k in dropping stage
- v_i = Volume of job i
- Q = Sufficiently large integer

Decision Variables

- $X_{ijk} = \begin{cases} 1, & \text{if job } i, \text{ which has blending code } j, \text{ is blended in machine } k \text{ in blending stage} \\ 0, & \text{otherwise. } i \in I, j \in J, k \in B_j \end{cases}$
- $Y_{ijk} = \begin{cases} 1, & \text{if job } i, \text{ which has blending code } j, \text{ is processed in machine } k \text{ in dropping stage} \\ 0, & \text{otherwise. } i \in I, j \in J, k \in D_j \end{cases}$
- $Z_{ijk} = \begin{cases} 1, & \text{if job } i, \text{ which has blending code } j, \text{ is filled in filling machine } k \text{ in filling stage} \\ 0, & \text{otherwise. } i \in I, j \in J, k \in F_i \end{cases}$
- $ox_{iqk} = \begin{cases} 1, & \text{if job } i \text{ precedes job } q \text{ in blending machine } k \\ 0, & \text{otherwise.} \end{cases}$
- $oy_{iqk} = \begin{cases} 1, & \text{if job } i \text{ precedes job } q \text{ in dropping machine } k \\ 0, & \text{otherwise.} \end{cases}$
- $oz_{iqk} = \begin{cases} 1, & \text{if job } i \text{ immediately precedes job } q \text{ in filling machine } k \\ 0, & \text{otherwise.} \end{cases}$
- cx_i = Completion time of job i in blending stage, $cx_i \geq 0$
- cy_i = Completion time of job i in dropping stage, $cy_i \geq 0$
- cz_i = Completion time of job i in filling stage, $cz_i \geq 0$

Based on the above definitions, the MILP formulation of the problem is given as follows.

$$\text{Minimize } \sum_{i \in I/\{0\}} w_i c z_i \tag{1}$$

$$\sum_{k \in B_j} X_{ijk} = 1 \quad \forall i \in I/\{0\}, \forall j \in J \tag{2}$$

$$\sum_{k \in D_j} Y_{ijk} = 1 \quad \forall i \in I/\{0\}, \forall j \in J \tag{3}$$

$$\sum_{k \in F_i} Z_{ijk} = 1 \quad \forall i \in I/\{0\}, \forall j \in J \tag{4}$$

$$Q(2 - X_{ijk} - X_{qjk} + o x_{iqk}) + c x_i - c x_q \geq p x_i \quad \forall i, q \in I/\{0\}, \forall j \in J, \forall k \in B_j | i < q \tag{5}$$

$$Q(3 - X_{ijk} - X_{qjk} - o x_{iqk}) + c x_q - c x_i \geq p x_q \quad \forall i, q \in I/\{0\}, \forall j \in J, \forall k \in B_j | i < q \tag{6}$$

$$Q(2 - Y_{ijk} - Y_{qjk} + o y_{iqk}) + c y_i - c y_q \geq p y_i \quad \forall i, q \in I/\{0\}, \forall j \in J, \forall k \in D_j | i < q \tag{7}$$

$$Q(3 - Y_{ijk} - Y_{qjk} - o y_{iqk}) + c y_q - c y_i \geq p y_q \quad \forall i, q \in I/\{0\}, \forall j \in J, \forall k \in D_j | i < q \tag{8}$$

$$Q(1 - o z_{iqk}) + c z_q - c z_i \geq p z_q + s_{iq} \quad \forall i, q \in I/\{0\}, \forall k \in F_i | i \neq q \tag{9}$$

$$\sum_{q \in I} o z_{0qk} = 1 \quad \forall k \in F \tag{10}$$

$$\sum_{q \in I} o z_{q0k} = 1 \quad \forall k \in F \tag{11}$$

$$\sum_{i \in I | i \neq q} o z_{iqk} = \sum_{j \in J} Z_{qjk} \quad \forall q \in I, \forall k \in F \tag{12}$$

$$\sum_{q \in I | i \neq q} o z_{iqk} = \sum_{j \in J} Z_{ijk} \quad \forall i \in I, \forall k \in F \tag{13}$$

$$c x_i \geq p x_i \quad \forall i \in I/\{0\} \tag{14}$$

$$c y_i - c x_i \geq \sum_{k \in D_j} Y_{ijk} p y_i \quad \forall i \in I/\{0\}, \forall j \in J \tag{15}$$

$$c z_i - c y_i \geq \sum_{k \in F_i} Z_{ijk} p z_i \quad \forall i \in I/\{0\}, \forall j \in J \tag{16}$$

$$v_i X_{ijk} \leq N_k \quad \forall i \in I/\{0\}, \forall j \in J, \forall k \in B_j \tag{17}$$

$$v_i Y_{ijk} \leq A_k \quad \forall i \in I/\{0\}, \forall j \in J, \forall k \in D_j \tag{18}$$

The objective function (1) minimizes the total weighted completion time of the jobs. Constraint sets (2), (3) and (4) ensure that each job is assigned to an eligible machine in each stage. Constraint sets (5) and (6) prevent any two job operations from being overlapped on the same blending machine. Similarly, constraint sets (7) and (8)

prevent any two job operations from overlapping on the same dropping machine. The constraint set (9) guarantees that the starting time of a job on a filling machine must be greater than or equal to the finishing time of the preceding job on that machine, regarding the sequence dependent setup time in between. Constraint sets (10) and (11) ensure that the dummy job 0 is the first and the last job in the job sequence on each filling machine. The constraint set (12) states that each job must be immediately preceded by another job on a filling machine. Similarly, the constraint set (13) ensures that each job must be immediately succeeded by another job on a filling machine. Constraint set (14) guarantees that the completion time of a job in the blending stage is greater than or equal to its processing time in that stage. Constraint sets (15) and (16) ensure that the starting time of a job in a stage is greater than or equal to its release time from the previous stage. Constraint sets (17) and (18) state that each job is assigned to a machine without exceeding the machine's capacity.

The proposed MILP model is solved for 5 instances with varying job sizes from 5 to 100, on IBM ILOG CPLEX Optimization Studio with CPLEX 12.8. The computational time of the mathematical model for different job sizes are shown in Table 1. As shown in Table 1, the solution time of the mathematical model increases exponentially as the job size increases. Note that, the entry '-' in the table means that the mathematical model cannot find the optimal solution within the computational time. This result is expected, as the studied problem has already been proven as NP-hard [11]. Consequently, heuristic algorithms have been proposed for the problem in the following subsection.

Table 1. Results of the mathematical model for different job sizes

| Instance | Number of jobs | Number of blending machines | Number of dropping machines | Number of filling machines | Optimal solution | Computational time (sec.) |
|----------|----------------|-----------------------------|-----------------------------|----------------------------|------------------|---------------------------|
| 1 | 5 | 5 | 4 | 3 | 1221.1 | 0.27 |
| 2 | 10 | 5 | 4 | 3 | 2660.7 | 1.16 |
| 3 | 15 | 5 | 4 | 3 | 4180.7 | 66.13 |
| 4 | 20 | 5 | 4 | 3 | 6139.4 | 10846.5 |
| 5 | 100 | 5 | 4 | 3 | - | 43925.8 |

3.2 Heuristic Algorithms

Due to the NP-hardness of the studied problem, two constructive heuristic algorithms are developed based on well-known Shortest Processing Time (SPT) rule. According to the SPT rule, jobs are sorted in increasing order of their total processing times and they are assigned to the available machines in this order. In the first constructive heuristic algorithm, SPT rule is applied for each stage by considering only the processing times at that stage. While, in the second algorithm, SPT rule is applied for each stage by considering the total processing times at that stage and the following stages. Namely, SPT rule is applied: for the first stage by considering the total processing times of all (blending, dropping and filling) stages; for the second stage by considering the total

processing times of dropping and filling stages; and for the third stage by considering the processing times of that stage (filling). In both heuristic algorithms, after the jobs are sorted according to the SPT rule, they are scheduled to the eligible machines regarding their blending codes, ready times from previous stage, weights and volumes. The general steps of the proposed heuristic algorithms are listed below. Note that, only difference between the two algorithms is the aforementioned processing time calculations for SPT rule.

1. *Blending Stage*

- 1.1. Group the jobs according to their blending codes.
- 1.2. For each group, schedule the jobs to the eligible machines according to their blending codes as below:
 - 1.2.1. Sort the jobs in decreasing order of their weights.
 - 1.2.2. Sort the jobs with same weight according to the SPT rule among themselves.
 - 1.2.3. In line with the order in step 1.2.2, schedule the jobs to the available machines regarding their blending codes and machine capacity limitations.

2. *Dropping Stage*

- 2.1. Group the jobs according to their blending codes.
- 2.2. For each group, schedule the jobs to the eligible machines according to their blending codes as below:
 - 2.2.1. Sort the jobs in increasing order of their ready times from previous stage.
 - 2.2.2. According to the order in step 2.2.1, schedule the first job for each machine regarding the blending codes of jobs and the machine capacities.
 - 2.2.3. Calculate the ready time of each machine according to the completion time of the last job on the machine.
 - 2.2.4. Determine the machine m_{min} , which has the minimum ready time.
 - 2.2.4.1. According to the order in step 2.2.1, if the earliest starting time of the next job is greater than or equal to the ready time of m_{min} , schedule the next job to the machine m_{min} regarding its ready time from previous stage.
 - 2.2.4.2. According to the order in step 2.2.1, if the earliest starting time of the next job is less than the ready time of m_{min} :
 - 2.2.4.2.1. Sort the jobs, which have smaller ready time than the ready time of m_{min} , in decreasing order of their weights.
 - 2.2.4.2.2. Sort the jobs with same weight according to the SPT rule among themselves.
 - 2.2.4.2.3. Schedule the first job in this order to the machine m_{min} regarding the ready time of m_{min} .
 - 2.2.5. If all jobs are not scheduled, go back to step 2.2.3.

3. *Filling Stage*

- 3.1. Group the jobs according their packaging material types.
- 3.2. For each group, schedule the jobs to the eligible machine according to their packaging material types as below:
 - 3.2.1. Sort the jobs in increasing order of their ready times from previous stage.
 - 3.2.2. Schedule the first job to the machine according to the order in step 3.2.1.

- 3.3. Calculate the earliest starting time of each job that may follow the last job on the machine, regarding the sequence-dependent setup time in between.
 - 3.3.1. For the jobs which has an earliest starting time that is greater than its ready time:
 - 3.3.1.1. Sort these jobs in increasing order of their earliest starting times, and then sort them in decreasing order of their weights.
 - 3.3.1.2. Sort the jobs with highest weight according to the SPT rule among themselves.
 - 3.3.1.3. Schedule the first job in this order to the machine.
 - 3.3.2. For the jobs which has an earliest starting time that is less than or equal to its ready time:
 - 3.3.2.1. According to the order in step 3.2.1, schedule the next job to the machine regarding its ready time and the sequence-dependent setup time in between.

In order to improve the performance of the proposed heuristic algorithms, additional improvement heuristic algorithm is also employed after the aforementioned constructive heuristics. In this study, as an improvement heuristic, insertion local search is applied to the solutions obtained by the above heuristic algorithms. In the insertion local search, a job is removed from its current position and inserted into all possible positions except its current position. Then, the removed job is inserted into the best improving position. This procedure is repeated for all jobs.

4 Computational Results

In this section, computational results are presented for the proposed heuristic algorithms. In order to evaluate the performance of the heuristic algorithms, 50 real instances with 15 jobs are obtained from the company, which have 4 blending machines, 4 dropping machines and 3 filling machines. The proposed MILP model in Sect. 3 is coded on IBM ILOG CPLEX Optimization Studio and all instances are solved optimally with CPLEX 12.8. All proposed heuristic algorithms are coded in Microsoft Visual Basic for Applications (VBA) Software and all instances are solved by heuristic algorithms within few minutes. The relative percent deviations (RPD) from the optimal results are reported for each heuristic algorithm, which is calculated as below:

$$RPD = \frac{\text{Heuristic Solution} - \text{Optimal Solution}}{\text{Optimal Solution}} \times 100 \quad (19)$$

4.1 Comparison of Constructive Heuristic Algorithms

Table 2 reports the RPD values for each constructive heuristic algorithm. As shown in the table, the average RPD value of the first constructive heuristic algorithm is 9.97%, while it is 11.32% for the second constructive heuristic algorithm over 50 instances. “Best Heuristic Algorithm” column of the table reports the best RPD values among

these two constructive heuristic algorithms. As shown in the table, the average RPD value of the best constructive algorithm is 9.26%. Note that, only difference between the two constructive heuristics is the aforementioned processing time calculations for SPT rule. As the first constructive heuristic slightly performs better than the second one, these processing time calculations slightly affects the performance of the algorithm.

Table 2. Comparison of constructive heuristics

| Constructive heuristic algorithms | | | | | | | |
|-----------------------------------|-----------------------|-----------------------|------------------------------|----------------|-----------------------|-----------------------|------------------------------|
| Instance | RPD (%) | | | Instance | RPD (%) | | |
| | Heuristic algorithm 1 | Heuristic algorithm 2 | The best heuristic algorithm | | Heuristic algorithm 1 | Heuristic algorithm 2 | The best heuristic algorithm |
| 1 | 8.42 | 6.63 | 6.63 | 26 | 2.15 | 0.07 | 0.07 |
| 2 | 5.47 | 5.86 | 5.47 | 27 | 8.78 | 12.73 | 8.78 |
| 3 | 12.28 | 12.28 | 12.28 | 28 | 7.19 | 9.26 | 7.19 |
| 4 | 13.03 | 12.47 | 12.47 | 29 | 26.17 | 22.90 | 22.90 |
| 5 | 1.94 | 2.51 | 1.94 | 30 | 23.78 | 35.60 | 23.78 |
| 6 | 2.69 | 4.44 | 2.69 | 31 | 9.99 | 9.99 | 9.99 |
| 7 | 4.43 | 4.43 | 4.43 | 32 | 9.03 | 9.03 | 9.03 |
| 8 | 8.04 | 8.34 | 8.04 | 33 | 11.05 | 11.05 | 11.05 |
| 9 | 8.38 | 4.32 | 4.32 | 34 | 15.92 | 15.92 | 15.92 |
| 10 | 9.21 | 9.60 | 9.21 | 35 | 8.86 | 2.88 | 2.88 |
| 11 | 12.62 | 12.62 | 12.62 | 36 | 7.58 | 7.33 | 7.33 |
| 12 | 4.23 | 5.68 | 4.23 | 37 | 6.69 | 6.69 | 6.69 |
| 13 | 18.24 | 26.14 | 18.24 | 38 | 8.78 | 8.96 | 8.78 |
| 14 | 4.18 | 4.18 | 4.18 | 39 | 1.03 | 14.47 | 1.03 |
| 15 | 24.90 | 24.42 | 24.42 | 40 | 28.54 | 31.53 | 28.54 |
| 16 | 9.74 | 14.44 | 9.74 | 41 | 17.26 | 24.05 | 17.26 |
| 17 | 6.26 | 4.39 | 4.39 | 42 | 7.77 | 13.20 | 7.77 |
| 18 | 10.74 | 12.77 | 10.74 | 43 | 6.97 | 3.10 | 3.10 |
| 19 | 7.92 | 20.30 | 7.92 | 44 | 20.56 | 24.49 | 20.56 |
| 20 | 2.23 | 6.14 | 2.23 | 45 | 5.80 | 16.86 | 5.80 |
| 21 | 3.88 | 2.98 | 2.98 | 46 | 23.62 | 22.93 | 22.93 |
| 22 | 6.50 | 6.46 | 6.46 | 47 | 8.73 | 11.77 | 8.73 |
| 23 | 3.29 | 3.84 | 3.29 | 48 | 2.94 | 4.25 | 2.94 |
| 24 | 2.43 | 2.17 | 2.17 | 49 | 13.40 | 7.12 | 7.12 |
| 25 | 8.64 | 9.39 | 8.64 | 50 | 15.95 | 12.95 | 12.95 |
| | | | | Average | 9.97 | 11.32 | 9.26 |

4.2 Comparison of Constructive Heuristic Algorithms with Insertion Local Search

As mentioned before, an insertion local search is applied to the aforementioned constructive heuristic algorithms in order to improve the solution quality. Table 3 reports the RPD values for each constructive heuristic algorithm with insertion local search, for the same 50 real instances in the previous subsection. As shown in the table, after applying the insertion local search procedure, the average RPD value of the first

constructive heuristic algorithm is decreased to 5.45%, while it is decreased to 5.12% for the second constructive heuristic algorithm. When the best heuristic algorithm is chosen for each instance, the average RPD value is reduced to 4.78%, as shown in the “Best Heuristic Algorithm” column of the table. Hence, it can be said that insertion local search enhances the solution quality significantly.

Table 3. Comparison of constructive heuristics with insertion local search

| Constructive heuristic algorithms with insertion local search | | | | | | | |
|---|-----------------------|-----------------------|------------------------------|----------------|-----------------------|-----------------------|------------------------------|
| Instance | RPD (%) | | | Instance | RPD (%) | | |
| | Heuristic algorithm 1 | Heuristic algorithm 2 | The best heuristic algorithm | | Heuristic algorithm 1 | Heuristic algorithm 2 | The best heuristic algorithm |
| 1 | 7.70 | 5.90 | 5.90 | 26 | 2.10 | 0.10 | 0.10 |
| 2 | 5.50 | 5.90 | 5.50 | 27 | 1.40 | 4.50 | 1.40 |
| 3 | 12.30 | 12.30 | 12.30 | 28 | 7.20 | 8.90 | 7.20 |
| 4 | 4.90 | 8.10 | 4.90 | 29 | 9.80 | 6.70 | 6.70 |
| 5 | 1.50 | 1.50 | 1.50 | 30 | 15.20 | 15.20 | 15.20 |
| 6 | 2.70 | 3.60 | 2.70 | 31 | 10.00 | 10.00 | 10.00 |
| 7 | 2.50 | 2.00 | 2.00 | 32 | 3.00 | 3.00 | 3.00 |
| 8 | 8.00 | 8.30 | 8.00 | 33 | 7.00 | 7.00 | 7.00 |
| 9 | 6.70 | 4.30 | 4.30 | 34 | 4.20 | 4.20 | 4.20 |
| 10 | 3.10 | 2.70 | 2.70 | 35 | 7.40 | 2.90 | 2.90 |
| 11 | 11.00 | 11.00 | 11.00 | 36 | 3.70 | 3.70 | 3.70 |
| 12 | 4.20 | 5.70 | 4.20 | 37 | 4.20 | 4.20 | 4.20 |
| 13 | 7.00 | 5.70 | 5.70 | 38 | 5.00 | 6.20 | 5.00 |
| 14 | 2.80 | 2.80 | 2.80 | 39 | 1.00 | 2.60 | 1.00 |
| 15 | 10.60 | 10.00 | 10.00 | 40 | 5.70 | 4.80 | 4.80 |
| 16 | 6.20 | 6.20 | 6.20 | 41 | 1.90 | 1.90 | 1.90 |
| 17 | 2.10 | 1.50 | 1.50 | 42 | 4.30 | 3.70 | 3.70 |
| 18 | 5.00 | 3.60 | 3.60 | 43 | 4.50 | 3.10 | 3.10 |
| 19 | 2.60 | 2.70 | 2.60 | 44 | 14.20 | 14.20 | 14.20 |
| 20 | 0.10 | 0.10 | 0.10 | 45 | 4.90 | 4.90 | 4.90 |
| 21 | 3.30 | 3.00 | 3.00 | 46 | 7.80 | 7.70 | 7.70 |
| 22 | 6.50 | 4.90 | 4.90 | 47 | 1.30 | 1.30 | 1.30 |
| 23 | 3.00 | 3.00 | 3.00 | 48 | 1.70 | 1.00 | 1.00 |
| 24 | 2.20 | 2.20 | 2.20 | 49 | 13.00 | 3.70 | 3.70 |
| 25 | 4.00 | 5.70 | 4.00 | 50 | 6.90 | 8.00 | 6.90 |
| | | | | Average | 5.45 | 5.12 | 4.78 |

4.3 Comparison of Proposed Solution Approaches with the Current System of the Company

Another 15 real instances are also obtained from the company in order to compare the current system with the proposed solution approaches. Note that, these instances have varying job sizes from 15 to 24, where there are 21 blending machines, 10 dropping

machines and 3 filling machines in each instance. All these 15 real instances are optimally solved by MILP model and the feasible results are obtained by using heuristic algorithms. Table 4 reports the improvement percentage for each solution approach against the current system. Note that, the “Heuristic Algorithm” column reports the best results among the results obtained by the two heuristic algorithms with insertion local search. As shown in the table, 27.80% improvement is observed in the total weighted completion time of the jobs in average, by using the developed heuristic algorithms. On the other hand, the optimal results are 31.67% better than the results of the current system, in average.

Table 4. Comparison of proposed solution approaches with the current system of the company

| Instance | Improvement (%) | |
|----------------|---------------------|-------------------------|
| | Heuristic algorithm | Optimal solution (MILP) |
| 1 | 45.88 | 49.82 |
| 2 | 49.17 | 56.82 |
| 3 | 10.41 | 12.32 |
| 4 | 49.63 | 77.64 |
| 5 | 37.30 | 37.30 |
| 6 | 55.02 | 55.79 |
| 7 | 27.29 | 32.88 |
| 8 | 4.92 | 4.92 |
| 9 | 44.24 | 44.24 |
| 10 | 23.02 | 27.95 |
| 11 | 0.89 | 0.89 |
| 12 | 2.08 | 5.22 |
| 13 | 4.08 | 4.08 |
| 14 | 60.00 | 60.00 |
| 15 | 3.14 | 5.11 |
| Average | 27.80 | 31.67 |

4.4 Sensitivity Analysis

Sensitivity analysis is also performed for the weight parameter of the problem. The aim of this analysis is to observe that how the performance of the solution approaches will be affected by the change of the weight parameter. For this analysis, 15 instances with 15 jobs are obtained from the company, which have 4 blending machines, 4 dropping machines and 3 filling machines. Table 5 reports the RPD values and computational times of the solution approaches for both weighted and unweighted form of the problem. As shown in Table 5, the proposed solution approaches perform robustly for both unweighted and weighted form of the problem, in terms of both solution quality and time. Note that, the average RPD value is 5.71% for the weighted form of the problem, while it is 3.97% for the unweighted form of the problem over 15 instances.

Table 5. Sensitivity analysis

| Instance | MILP | | Heuristic algorithm | | RPD (%) | |
|----------------|---------------------------|--------------|---------------------------|--------------|-------------|-------------|
| | Computational time (sec.) | | Computational time (sec.) | | Weighted | Unweighted |
| | Weighted | Unweighted | Weighted | Unweighted | | |
| 1 | 31.03 | 20.79 | 98.16 | 53.16 | 2.30 | 3.31 |
| 2 | 30.73 | 20.01 | 86.17 | 40.27 | 10.75 | 6.24 |
| 3 | 27.32 | 20.42 | 76.41 | 41.39 | 7.31 | 2.36 |
| 4 | 11.19 | 27.27 | 94.04 | 30.60 | 3.29 | 2.14 |
| 5 | 3.19 | 12.21 | 157.52 | 36.53 | 3.68 | 1.26 |
| 6 | 6.76 | 12.01 | 84.62 | 31.45 | 3.10 | 2.53 |
| 7 | 25.15 | 3.92 | 40.72 | 33.19 | 2.68 | 1.82 |
| 8 | 26.28 | 5.84 | 40.17 | 32.47 | 4.85 | 8.19 |
| 9 | 28.77 | 16.25 | 35.35 | 38.12 | 1.51 | 4.29 |
| 10 | 8.00 | 12.52 | 32.59 | 33.27 | 3.67 | 2.27 |
| 11 | 7.30 | 8.95 | 34.55 | 33.45 | 4.38 | 3.01 |
| 12 | 8.04 | 5.16 | 29.95 | 29.29 | 26.42 | 13.76 |
| 13 | 31.70 | 7.84 | 39.59 | 34.27 | 3.98 | 0.62 |
| 14 | 9.70 | 19.73 | 37.27 | 30.23 | 6.53 | 1.66 |
| 15 | 7.64 | 20.56 | 40.30 | 32.69 | 1.17 | 6.15 |
| Average | 17.52 | 14.23 | 61.83 | 35.36 | 5.71 | 3.97 |

5 Decision Support System

A user-friendly decision support system (DSS) is created by embedding proposed heuristic algorithms in Microsoft Visual Basic for Applications (VBA) Software. In the developed DSS, the user is able to work in a practical way by entering only product codes and units. When the product codes and units are entered by the user, other information about the product automatically appears on the page. In order to provide flexibility for the user, the user can also edit, delete and add the input data easily. The main page of the DSS is shown in Fig. 1.



Fig. 1. Main page of the decision support system

The developed DSS employs both of the two heuristic algorithms and it offers several options to see different solution alternatives. The options include the first heuristic algorithm, the second heuristic algorithm and the best option among these two heuristic algorithms. Note that, insertion local search is included in all heuristic algorithms. After choosing the solution approach, the DSS reports the results to the user with detailed lists and Gantt Charts. Solution report screen of the DSS is shown in Fig. 2. As shown in Fig. 2, the starting time, the processing time and the completion time of each job at each stage can be seen in these reports as well as detailed Gantt Charts.

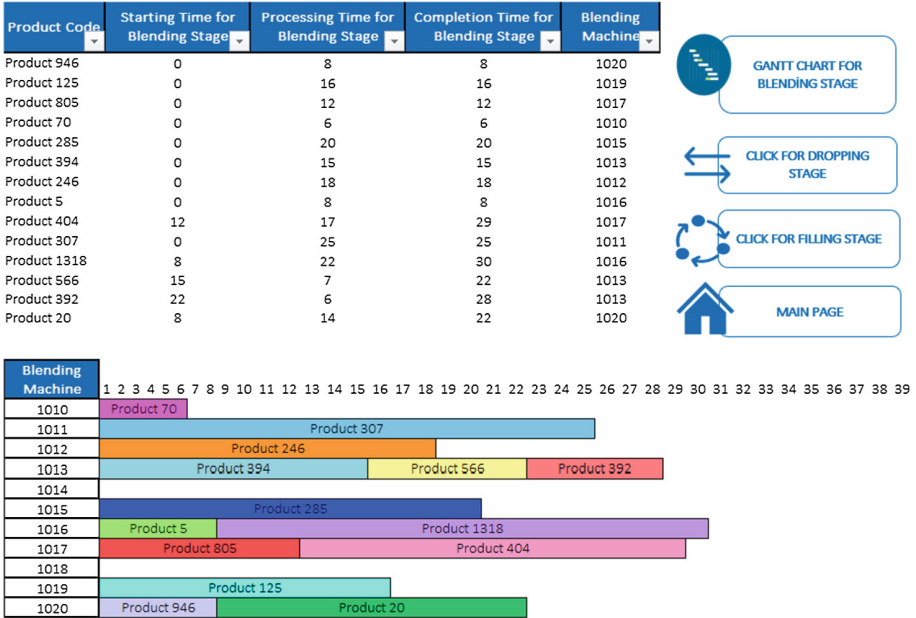


Fig. 2. Solution report of the decision support system

6 Conclusions

In this study, a real-life HFSP of a lubricant company is considered, where the aim is to minimize total weighted completion time of the jobs. Different from the standard HFSP, the studied problem has additional limitations such as machine eligibility, sequence-dependent setup times and machine capacities. A novel mixed integer linear programming model is proposed for the studied HFSP with three stages. As the problem is NP-hard, two constructive heuristic algorithms, which are based on SPT rule, and an improvement heuristic algorithm are also developed. The performance of the proposed heuristic algorithms is evaluated by comparisons with the optimal results. The extensive computational results show that proposed heuristic algorithms find near optimal results in reasonable computational times. Based on 50 real instances, average RPD value is 9.26% for the best constructive heuristic. After applying insertion local search to the constructive heuristics, the average RPD value is further decreased to

4.78%. When the current system is compared with the heuristic algorithms, it can be seen that there is 27.80% improvement on the total weighted completion time of the jobs. Finally, the proposed heuristic algorithms are integrated into a user-friendly decision support system using Microsoft Excel VBA interface, to provide an efficient scheduling tool for the company. Note that, the developed decision support system reports the results to the user with detailed lists and Gantt Charts.

As a future work, the decision support system can be made more professional by integrating to the current resource management system of the company. Furthermore, metaheuristic algorithms can be developed for the studied problem employing the proposed constructive heuristics. The developed metaheuristic algorithms can also be embedded into the decision support system. Different objectives or constraints can also be considered for the studied problem in future studies.

Acknowledgments. This study is supported by TUBITAK (The Scientific and Technological Research Council of Turkey) under the program of “2209-B Undergraduate Research Projects for Industrial Applications Fellowship Program”.

We would like to thank Deniz Türsel Eliyi for her guidance and enlightenment. Additionally, we would like to thank İ. Çağcan Çatıkkaş and Alp Arslan for their support in this study. Finally, we are grateful to the company and Dilek Turan for their co-operation and support.

References

1. Ruiz, R., Vázquez-Rodríguez, J.A.: The hybrid flow shop scheduling problem. *Eur. J. Oper. Res.* **205**(1), 1–18 (2009)
2. Ribas, I., Leisten, R., Framinan, J.M.: Review and classification of hybrid flowshop scheduling problems from a production system and a solutions procedure perspective. *Comput. Oper. Res.* **37**, 1439–1454 (2010)
3. Chikhi, N., Abbas, M.: Makespan minimization for two-stage hybrid flow shop with dedicated machines and additional constraints. In: 9th International Conference on Modeling, Optimization & Simulation – MOSIM 2012, France, pp. 123–133 (2012)
4. Lin, H.-T., Liao, C.-J.: A case study in a two-stage hybrid flow shop with setup time and dedicated machines. *Int. J. Prod. Econ.* **86**(2), 133–143 (2003)
5. Ruiz, R., Serifoglu, F.S., Urlings, T.: Modeling realistic hybrid flexible flowshop scheduling problems. *Comput. Oper. Res.* **35**(4), 1151–1175 (2008)
6. Ruiz, R., Maroto, C.: A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *Eur. J. Oper. Res.* **169**(3), 781–800 (2006)
7. Yaurima, V., Burtseva, L., Tchernykh, A.: Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. *Comput. Ind. Eng.* **56**(4), 1452–1463 (2009)
8. Pan, Q.K., Gao, L., Li, X.Y., Gao, K.Z.: Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times. *Appl. Math. Comput.* **303**, 89–112 (2017)
9. Yu, C., Semeraro, Q., Matta, A.: A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Comput. Oper. Res.* **100**, 211–229 (2018)
10. Öztöp, H., Tasgetiren, M.F., Eliyi, D.T., Pan, Q.K.: Metaheuristic algorithms for the hybrid flowshop scheduling problem. *Comput. Oper. Res.* **111**, 177–196 (2019)
11. Gupta, J.N.D.: Two-stage hybrid flow shop scheduling problem. *J. Oper. Res. Soc.* **39**(4), 359–364 (1988)