RESTRICTED ENUMERATION AND MACHINE GROUPING BASED APPROACH FOR HYBRID FLEXIBLE FLOW SHOP SCHEDULING PROBLEMS WITH SEQUENCE-DEPENDENT SETUP TIMES

Müjgan SAĞIR[1*], Hacer Defne OKUL[2]

[1]Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü, Eskişehir
ORCID No : https://orcid.org/0000-0003-2781-658X
[2]Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü, Eskişehir
ORCID No : https://orcid.org/0000-0002-9004-5469

Keywords

Hybrid Flexible Flow Shop Scheduling, Machine grouping, 0-1 Integer Programming Model, Makespan, Heuristic Approach.

Abstract

This paper presents a 0-1 Mixed Integer Programming Model and a heuristic approach based on restricted enumeration and machine grouping for solving k-stage Hybrid Flexible Flow Shop Problems with Sequence-Dependent Setup Times (HFFS_SDST) with Cmax. As presented in this paper in detail, recent reviews show that there is less research on flowshop with SDST than on regular flowshops, although Permutation Flow Shop (PFS) problems is extensively considered in the literature. Mathematical model presented in this work can solve the problems with five stages, eight machines, and six jobs in an acceptable time and this size is relatively bigger than of the ones previously have been solved by mathematical models for defined problems. The heuristic algorithm, based on sub-flow shops via forming machine groups in different clusters, is suggested to give good solutions for large-size problems and can to reach the optimal or near-optimal solutions for some instances and the computational time of the algorithm is shorter than solution time of the mathematical model as expected. In terms of proposed heuristic, to the best of our knowledge, this type of grouping algorithm to form flow shops in the machine environment is not introduced before and provides more efficient solutions by restricting the enumerations. There are no test problems for the problem under consideration. Therefore, real manufacturing environment instances are used. Computational results show that the proposed model and the heuristic algorithm are superior to the current scheduling approach in the company. The heuristic approach also provides a user-friendly environment and efficient scheduling.

SIRA BAĞIMLI HAZIRLIK SÜRELİ MELEZ ESNEK AKIŞ TİPİ ATÖLYELERDE ÇİZELGELEME PROBLEMİ İÇİN SINIRLI SAYIMLAMA VE MAKİNE GRUPLAMAYA DAYALI BİR YAKLAŞIM

Anahtar Kelimeler

Melez Esnek Akış Tipi Atölye Çizelgeleme Problemi, Makine Gruplama, 0-1 Tamsayılı Programlama Modeli, Yayılma Süresi, Sezgisel Yaklaşım

Öz

Bu çalışmada sıra bağımlı hazırlık zamanlı k kademeli melez esnek akış tipi atölye çizelgeleme problemi için bir 0-1 tamsayılı matematiksel model ve sınırlandırılmış sayımlama ve makine gruplama tabanlı bir sezgisel algoritma önerilmektedir. Son çalışmalar sıra bağımlı hazırlık zamanlı akış tipi atölye çizelgeleme problemlerinin klasik atölye çizelgeleme problemlerine göre daha az çalışıldığını göstermektedir. Çalışmada yer verilen matematiksel model beş kademe, sekiz makine, altı işin olduğu bir sistemi kabul edilebilir sürede çözebilmektedir. Akış atölyeleri oluşturma temelinde önerilen sezgisel algoritma ise farklı problem büyüklükleri için elde edilen sonuçlardan da görüleceği gibi ele alınan işletmede karşılaşılan büyük boyutlu problemleri eniyi ya da eniyi çözüme yakın düzeyde çözebilmektedir. Erişebildiğimiz yayınlara göre; bu tür problemler için akış atölyesi oluşturarak, işlerin işlem göreceği makina kombinasyonu sayısını azaltma yaklaşımı kullanılmamıştır. Ayrıca söz konusu problem için test verisine rastlanamamıştır. Bu nedenle gerçek bir üretim atölyesinden elde edilen veri setleri işletmenin onayı doğrultusunda kullanılmıştır. Sonuçlar, önerilen modelin ve geliştirilen sezgisel yaklaşımın işletmede kullanılan mevcut çizelgeleme yaklaşımına üstün olduğunu göstermektedir. Sezgisel yaklaşım, aynı zamanda kullanıcı dostu bir yaklaşım ile tasarlanarak, tezgâhların herhangi bir zamandaki performanslarını, işlerin tamamlanma seviyelerini ve benzeri bazı istatistikleri göstererek karar vericiye destek olabilecek özelliktedir.

Araştırma Makalesi		Research Article	
Başvuru Tarihi	: 18.03.2020	Submission Date	: 18.03.2020
Kabul Tarihi	: 16.10.2020	Accepted Date	: 16.10.2020

*Sorumlu yazar; e-posta : mujgan.sagir@gmail.com

1. Introduction

In a typical hybrid flow shop scheduling (HFS) problem (Figure 1), a set of jobs need to be processed through production stages and identical parallel machines are to be present at each stage. Once completed at a stage, a job can be directly and immediately sent to the following stage if at least one machine in this stage is available or can be stored at the infinite buffer between the consecutive stages. The task of the HFS problem is to establish a production schedule to attain performance optimization at some level (Ribas, Leisten, and Framinan, 2010). In the flexible version of HFS problem, on the other hand, all jobs are not necessarily to be processed at all stages. This type of HFS is referred to as hybrid flexible flowshop scheduling (HFFS) (Naderi, Zandieh, Balagh and Roshanaei, 2009b) and there is limited work focusing on this problem.

Setup times, which are not part of the processing times, involve operations that have to be performed on machines, such as repairing, cleaning, or releasing machines. Setup times may or may not depend on the job sequence. Dudek, Smith and Panwalkar (1974) reported that 70% of industrial activities include sequence-dependent setup times (SDST's). More recently, it has been reported that 50% of 250 industrial projects contain SDST and that 92% of the order deadlines are met when these setup times are taken into account (Sioud and Gagné, 2018).

Gupta (1988) showed the flow shop with multiple processors with only two stages to be NP-hard. The addition of precedence constraints does not simplify the problem, since the two parallel machines problem with makespan objective and general precedence constraints $(P/Prec/Cmax)$ are strongly NP-Hard according to Pinedo (2002). The same applies to the other characteristics considered as, for example, the regular flow shop with sequence-dependent setup times $(F/S_{ijk}/Cmax)$, which was shown to be *NP-Complete* by Gupta (1988). A one-machine scheduling problem with SDST is NP-hard (Zandieh, Ghomi and Husseini, 2006). Therefore, HFS with SDST is considered as an NP-hard problem in the strong sense (Jabbarizadeh, Zandieh and Talebi, 2009).

In this paper, we schedule k-stage HFFS with SDST under the minimization of makespan. It is assumed that jobs can wait between stages and preemption is not allowed. This problem will be denoted as $(HFFS/S_{ijk}/Cmax)$.

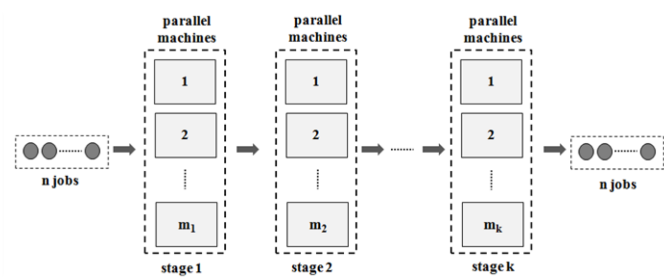

Figure 1. An Illustration of a k-stage Hybrid Flow Shop

The paper is organized as follows: Section 2 reviews the SDST hybrid flow shop literature. A mathematical model for the problem is presented in Section 3. The algorithm implemented in this study is given in Section 4. Computational results on real problem instances are presented in Section 5. Finally, Section 6 concludes the paper.

2. Literature review

The term 'hybrid flow shop' was first used by Gupta (1988). The problem was defined as scheduling with parallel processors to minimize makespan where it was demonstrated to be NP-hard.

Most studies assumed that either no setup has to be performed or that setup times are sequence-independent. Kurz and Askin (2003) compared several methods for a makespan minimization problem with sequence-dependent setup times, where jobs were allowed to skip stages. They also developed an integer model, insertion heuristics attempt to simultaneously equalize workload on all processors at a stage, and a random keys genetic algorithm for the problem. Andrés, Albarracín, Tormo, Vicens and García-Sabater (2005) considered the problem of product grouping in the tile industry. They propose an approach base on a genetic algorithm for a three-stage HFSP with SDST.

Ruiz and Maroto (2006) presented a genetic algorithm for hybrid flow shops with SDST. Moreover, Naderi et al., (2009b) hybridized the simulated annealing with a simple local search to solve the hybrid flow shops scheduling with SDST, transportation time, total completion time, and total tardiness as objective functions.

Gholami, Seyed, Hakimifar, Nazemi and Jolai (2017) considered a flow shop scheduling problem with SDST in an uncertain environment. Its objective function was to minimize weighted mean completion time. As for uncertainty, setup and processing times were considered not to be deterministic. The authors proposed two different approaches to deal with the uncertainty of input data: Robust optimization and fuzzy optimization.

Pan, Gao, Li, and Gao (2017) proposed some algorithms to minimize the makespan for the hybrid flow shop scheduling problem with SDST. Several adaptations of other recent well-known metaheuristics for the problem were presented to evaluate the proposed algorithms, and a comprehensive set of computational and statistical experiments were conducted to demonstrate the effectiveness of the presented algorithms.

Ribas, Leisten and Framinan (2010) classified papers in a review according to HFS characteristics and production limitations. According to this review, concerning for solution approaches, branch-and-bound (B&B) and mixed-integer programming are the most frequently used exact procedures and are the most efficient approaches to solve instances with 15 or 20 jobs and in five stages at most. He also stated that most of the papers in the flow shop literature assumed that there is one operation for all production stages. In some systems, some jobs can skip some stages, which is called "missing operations" in this review. Therefore, it is not always necessary to traverse each stage and equalize the processing time to zero at these stages for the production of an end product.

Morais, Godinho and Boiko (2013) dedicated his research to the production scheduling problem in a hybrid flow shop with and without setup times separated from processing times. The goal was to review the current literature to identify and analyze papers that developed methods to solve this problem. Analyses were performed with reference to the number of papers published over the years, the approach used in the development of the methods for solutions, the type of objective function, the performance criterion adopted, and the additional constraints considered. According to Morais, there are several applications of the hybrid flow shop scheduling problem that consider setup times in industry; thus, this field of study will continue to attract the attention of researchers. In general, the most difficult situation involving setup times is HFS with sequence-dependent, which has received more attention from researchers. We also see from this paper that the solution methods for the problem are mostly based on metaheuristics.

Recently, Allahverdi (2015) has surveyed scheduling problems with setup/cost times. Only a few out of 150 papers on flow shop problems dealt with PFS/SDST- $C\ max$, although PFS is extensively considered in the literature. Among these, the number of papers related to hybrid flow shops with SDST and k-stage, as does our work, is considerably low. Additionally, stage skipping is also considered in our study presented here.

Sioud and Gagné (2018) have presented an enhanced "Migrating Bird Optimization (MBO)" algorithm and a new heuristic for solving a scheduling problem. The proposed approaches are applied to a permutation flow shop with SDST and the objective of minimizing the makespan. Sioud and Gagné (2018) also pointed out that there are less researches on flow shop with SDST than on regular flow shops.

When problems grow in complexity or data volume, approximate methods are usually proposed. Several useful papers can be found for a basic review of HFS under the makespan criterion. Gupta (1988) introduced a new heuristic based on the longest processing time index. Over the last decade, metaheuristic and evolutionary approaches, which are used alone or in combination with traditional heuristics, have been proposed. Haouari and M'Hallah (1997) proposed two approximate methods developed in two phases. The first solution was generated using the longest remaining work rule. This schedule was improved using techniques based on simulated annealing and tabu search. Both solutions were encoded with the list used in the first phase. In the study by Portmann, Vignier, Dardilhac and Dezalay (1998), an improvement on the B&B proposed by Brah in 1996 was analyzed. The authors presented a metaheuristic using a GA in the B&B procedure to improve the values of the upper bound in certain stages. According to our survey of the literature, the number of hybrid flow shop scheduling problems with k-stage has risen over the years, as shown in Figure 2 below.

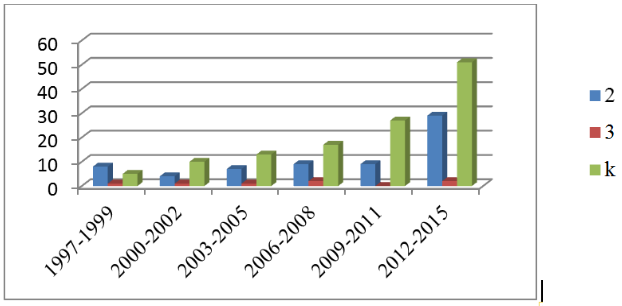

Figure 2. Hybrid Flow Shop Scheduling Problems with Different Stages

Among the exact methods for the HFS problem, B&B is the most preferred solution method. However, it should be noted that the size of the problems solved through B&B is relatively small. For this reason, many researchers turned to develop dispatching rules and constructive heuristics to quickly obtain a near-optimal solution for complex HFS problems (Tian, Li, and Lius, 2016).

There are some mathematical models proposed for hybrid flow shop problems with SDST, and they often have common decision variable definitions based on consecutiveness. Naderi, Zandieh and Shirazi (2009a) developed a mixed-integer linear programming model for flexible flow shop problems with SDST. In this study, the decision variable is a binary variable that takes value 1 if job *j* is processed immediately after job *k* at stage *i*, and 0 otherwise. Since the problem is NP-hard, a metaheuristic is proposed. Ruiz, Şerifoğlu and Urlings (2008) also proposed a formulation along with a mixed-integer model and some heuristics for the problem of scheduling *n* jobs on *m* stages, where there are several unrelated machines at each stage. The jobs might skip stages. The decision variable X_{iljk} is 1 if job *j* precedes job *k* on machine *l* at stage *i*. The proposed MIP model can only be tested in moderately sized instances. To analyze the effect of the different characteristics of the HFFL problem in larger instances, some simple heuristics are employed. Jungwattanakita, Chaovalitwongsea and Wernerb (2009) proposed a 0–1 mixed-integer linear programming formulation for the problem and the decision variable was 1 if job *j* was scheduled immediately before job *l* on the machine *i* at stage *t*, and 0 otherwise. For this problem, the mathematical model can solve the problems up to seven jobs, and four stages (the data for the number of the machines is not provided) in acceptable time and a constructive heuristic is proposed. Zandieh, Ghomi and Husseini (2005) presented two mixed-integer programming models for scheduling hybrid flow shops with SDST. In the first model, the machines at each stage were assumed as identical, and as different in the second stage. The models can find optimal solutions for moderate sizes.

Azadeh, Goodarzi, Kolaee and Jebreili (2019) developed an integrated approach based on artificial neural network (ANN), genetic algorithm (GA) and computer simulation to explore all the solution space in stochastic flexible flow shop with sequence-dependent setup times, job deterioration and learning effects. The objective of this study is minimizing total tardiness of jobs in the sequences.

Qin, Zhuang, Liu and Tang (2019) investigated a multi-stage hybrid flow shop scheduling problem with lot sizing, calendar constraints and sequence-dependent setup times. They developed a hierarchical approach to decompose the original problem into subproblems. An ant colony algorithm with lot sizing to evolve best results in the makespan performance is proposed.

Very recently, Cai, Zhou and Lei (2020) considered uncertainties often neglected in the previous works and fuzzy distributed scheduling is addressed in two-stage hybrid flow shop with sequence-dependent setup times. A collaborative variable search (CVS) is proposed to optimize total agreement index and fuzzy makespan simultaneously.

There are more researches published in very recent years related to hybrid flow shop problems but they do not always consider sequence dependent setup times.

3. Mathematical model

There are mathematical models for hybrid flow shop problems in the literature and some of them are explained in the previous section. Often, heuristics and priority rules found in the literature do not take sequence-dependent setup times (SDST) into consideration (Allahverdi, Cheng and Kovalyov, 2008; Zhu and Wilhelm, 2006). Yet, our model has the following characteristics: Hybrid flexible flow shop with SDST, *k*-stages, stage skipping, and *Cmax*. We propose a decision variable to decide the 'sequence' of a job on a machine at a stage instead of using a variable to define the consecutiveness of two jobs on a machine at a stage as in Naderi et al. (2009a).

The following section presents our mathematical model and the solution performances.

In the considered ($HFFS/S_{ijk}/Cmax$) it is assumed that: 1) Problem data is deterministic, 2) Each stage has at least one machine, and at least one stage has more than one machine, 3) A machine can process only one job at a time, 4) Each job must be processed by at most one machine in each stage, 5) No preemption or interruption is allowed, 6) Jobs are available for processing at a stage immediately after releasing from the previous stage. The following notation is used throughout the rest of this paper:

n	Number of jobs
k	Number of stages
m_t	Number of machines at stage t
p_{jt}	Processing time of job j at stage t
s_{ijt}	Setup time of job j if job j is assigned to at the first position at stage t
h_{jt}	Sequence-dependent setup time from job i to job j at stage k
q_j	Demand for job j

Sets		Indexes	
$N = \{1,\ldots,n\}$	Jobs	$i, j \in N$	Jobs
$M = \{1,\ldots,mt\}$	Machines	$s \in N$	Job sequence
$K = \{1,\ldots,k\}$	Stages	$l \in M$	Machines
		$t \in K$	Stages

<u>Decision Variables:</u>

C_{jt} : completion time of j at stage t

$Cmax$: makespan

$$X_{jtls} = \begin{cases} 1, & \text{if job } j \text{ is allocated to sequence } s \text{ of machine } l \text{ at stage } t \\ 0, & \text{otherwise} \end{cases}$$

Now we can formulate the problem as follows:

Minimize Cmax

subject to

$$\sum_{l=1}^{m_t}\sum_{s=1}^{n} X_{jtls} = 1 \qquad \forall(j,t) \quad j \in N,\ t \in K \tag{1}$$

$$\sum_{j=1}^{n} X_{jtls} \le 1 \qquad \forall(t,l,s) \quad t \in K,\ l \in M,\ s \in N \tag{2}$$

$$C_{jt} + M * (1 - X_{jtls}) \ge C_{j(t-1)} + p_{jt} * q_j + h_{jt} \tag{3}$$

$$\forall(j,l) \quad s = 1, t > 1, (j,s) \in N, l \in M, t \in K$$

$$C_{jt} - C_{it} + M * (2 - X_{itl(s-1)} - X_{jtls}) \ge p_{jt} * q_j + s_{ijt} \tag{4}$$

$$\forall(i,j,t,l) \quad i \ne j, s > 1, (i,j,s) \in N, l \in M, t \in K$$

$$C_{jt} \ge C_{j(t-1)} + p_{jt} * q_j \tag{5}$$

$$\forall(j,t) \quad t > 1,\ j \in N,\ t \in K$$

$$\sum_{j=1}^{n} X_{jtls} - \sum_{i=1}^{n} X_{itl(s-1)} \leq 0 \quad (6)$$

$\forall(t,l) \quad s > 1, i \neq j, (i,j,s) \in N, l \in M, t \in K$

$$C_{max} \geq C_{jt} \qquad \forall(j,t)\ j \in N,\ t \in K \quad (7)$$

$$C_{jt} \geq 0 \qquad \forall(j,t)\ \ j \in N,\ t \in K \quad (8)$$

$$X_{jtls} \in \{0,1\} \quad \forall(j,t,l,s)\ \ (j,s) \in N,\ l \in M,\ t \in K \quad (9)$$

Eq. (1) ensures that every job at every stage is assigned to only one sequence position on a machine in that stage. Eq. (2) specifies that a machine can process at most one job at a time.

Eq. (3) and Eq. (4) specify the completion time of job *j*. If the job is assigned to the 'first sequence' on machine *l* at stage *t*, the completion time of job *j* is specified by Eq. (3). The value of *M* is set to a very large constant. On the other hand, if job *j* is assigned to second or greater sequence on machine *l* at each *t* and if job *j* is immediately scheduled after job *i* then the completion time of job *j* is specified by Eq. (4). Eq. (5) reflects the precedence relations. Eq. (6) ensures the assignment of jobs successively. Eq. (7) specifies the makespan and finally Eq. (8) and Eq. (9) represent the state of the decision variables.

An optimal solution can be obtained by running GAMS 24.2.2 with IBM ILOG CPLEX optimizer and Intel® Core™ i5 CPU / 2.50 GHz / 4 GB RAM. We have found that the mathematical model can solve the problems up to the following configurations in an acceptable time: '2 stages, 3 machines, 8 jobs', '4 stages, 5 machines, 7 jobs', '6 stages, 7 machines, 6 jobs'. Later, we will turn back to the performance of the mathematical model in comparison with the proposed heuristic. The next section presents the heuristic algorithm suggested in this paper.

In this study, research and publication ethics were followed.

4. Proposed heuristics

In the hybrid flow shop under consideration, the process has multiple stages. We generate sub-algorithms for single or multi-machine ($m_t \geq 2$) situations and algorithm steps will be performed according to the number of machines in the stages.

We suggest that if there are successive stages that have the same number of parallel machines, various facilities can be provided to reduce the number of combinations on sequencing the jobs on those machines; and therefore, enabling us to increase the efficiency of the algorithm. The approach proposed is based on creating flow shops by forming machine groups under some conditions. All jobs might be processed at these successive stages. Even if one of the jobs in the set of jobs to be scheduled is not processed at these stages, flow shops cannot be created through these stages. Forming the machine groups under these conditions are defined in algorithm steps.

If a job does not have any process at any stage, the processing time of the job for that stage is assigned zero. Therefore our problem allows jobs to skip some work stations.

The following section summarizes pseudo code of the algorithm.

Begin
Read data **(**all processing times, setup times, routes, demands, the stage number and information of machines at stages)
Initialize the system parameters
Initial:
$t = 0$, $j = 0$, $a = 0$, $l = 0$ (t: stage, j: job, a: flow shop, l: machine)
$Cmax$ = a big number
For t := 0 **to** k **do**
If there is a flow shop **then**
For j := 0 **to** n **do**
For a := 0 **to** m_t **do** **(**m_t: the number of machines in stage t)
Read the last processed job assigned the first machine of flow shop a.
Calculate the total spending time for job j at flow shop a,
$$ts_{ja} = \left(p_{jn} * q_j + s_{ijt}\right) + \left(p_{j(n+1)} * q_j + s_{ij(t+1)}\right) + \cdots + \left(p_{j(n+x)} * q_j + s_{ij(t+x)}\right)$$
$$j\epsilon USP, i\epsilon SP, t\epsilon K, x\epsilon R$$
Calculate the completion time of job j at flow shop a,
$$C_{j(t+x)a} = B_{jt} + ts_{ja} + Z_{j(t+1)} + \cdots + Z_{j(t+x)}$$

End for
Sort the jobs in ascending order of total completion times, $j\epsilon USP$.
Choose the first job in the sorted list (this job is a scheduled job (e)).
$e = arg\{enk\{C_{j(t+x)a}\}\}$, $(j \epsilon USP)$
Put the job (e) behind the last assigned job at the machine in the first stage of chosen flow shop,
$USP = USP/\{e\}$
$SP = SP \cup \{e\}$
Save the completion time of job (e) at flow shop (a), $C_{e(t+x)}$. It is equal to the completion time of job (e) at stage (t+x),
$C_{e(t+x)} = C_{e(t+x)a}$ $(e \epsilon SP, (t+x) \epsilon K)$
End for
Else read the machine number of stage t, m_t
If $m_t \geq 2$ **then**
For l **:= 0 to** m_t **do**
Read the last job assigned the machine l at the stage t.
Calculate the total spending time for job j at stage t for machine l,
$ts_{jtl} = p_{jt} * q_j + s_{ijt}$ $(i \epsilon SP, j\epsilon USP)$
Calculate the completion time of job j at stage t for machine l,
$C_{jtl} = B_{jt} + ts_{jtl}$ $(j \epsilon USP, i \epsilon SP, i \neq j, l \epsilon M)$

End for
Sort the jobs in ascending order of total completion times, $j\epsilon USP$.
Choose the first job in the sorted list (this job is a scheduled job (e)).
$e = arg\{enk\{C_{jtl}\}\}$, $(j \epsilon USP)$
Put the job *(e)* behind the last assigned job of chosen machine
$USP = USP/\{e\}$
$SP = SP \cup \{e\}$
Save the completion time of job (e) at stage t
$C_{et} = C_{etl}$ $(e \epsilon SP, t \epsilon K, l \epsilon M)$
Else

Read the last job assigned the machine at the stage t.

Calculate the total spending time for job j at stage t
$ts_{jt} = p_{jt} * q_j + s_{ijt}$ $(i \epsilon SP, j\epsilon USP)$
Calculate the completion time of job j at stage t
$C_{jt} = B_{jt} + ts_{jt}$ $(j \epsilon USP, i \epsilon SP, i \neq j)$
Sort the jobs in ascending order of total completion times, $j\epsilon USP$.
Choose the first job in the sorted list (this job is a scheduled job (e)).
$e = arg\{enk\{C_{jt}\}\}$, $(j \epsilon USP)$
Put the job behind the last assigned job at the machine.
$USP = USP/\{e\}$
$SP = SP \cup \{e\}$
Save the completion time of job (e) at stage t, $C_{et,}$

End if

End for

Find the maximum completion time among the completion time of all jobs.
$Cmax = max(C_{jt}),\ (j \in SP)$
End

5. Computational evaluation

The presented mathematical model and the heuristic algorithm are compared by using real- data instances from a refrigerator factory with the approval of their management. The workshop under consideration produces door plates and has 5 production stages. The machine environment is as follows: There are two punch presses in the first stage, two bending presses in the second stage, one door line in the third stage, one door line in the fourth stage, and two hydraulic presses in the fifth stage. Therefore, the number of stages and machines for the problem can be specified as k=5, m_1=2, m_2=2, m_3=1, m_4=1, m_5=2. The stage and machine configuration of the workshop is given in Figure 3. It is not necessary to traverse each stage and equalize the processing time to zero at these stages for the production of an end product.

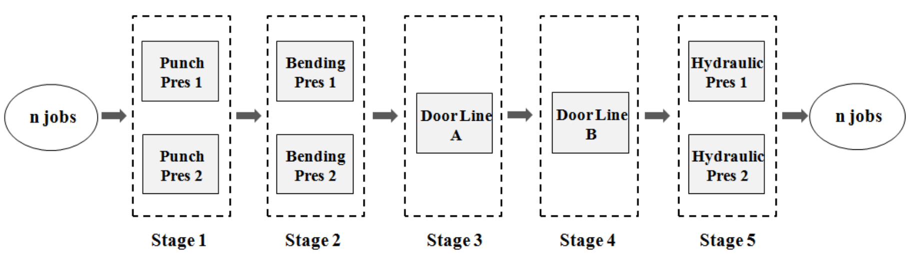

Figure 3. Stage and Machine Configuration of the Workshop

As seen in Figure 3, the first and second stages are successive and each has two identical parallel machines. All jobs are processed at these first two stages. Therefore, as explained in Section 4, we can form flow shops by having one punch press (Punch Pres 1 from Stage 1) and one bending press (Bending Press 1 from Stage 2) for the first flow shop. The second flow shop is also constructed by one punch press (Punch Press 2 from Stage 1) and one bending press (Bending Press 2 from Stage 2) for the second flow shop as in Figure 4.

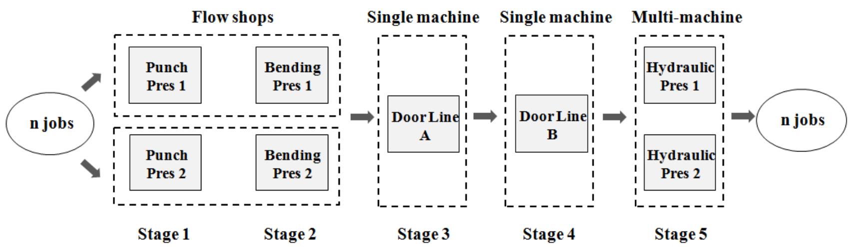

Figure 4. Forming Flow Shops

This arrangement increases our efficiency since the Punch Press 1 and Punch Press 2 in the first stage are identical, and similarly, the Bending Press 1 and Bending Press 2 in the second stage are identical. Since all the jobs must first be processed in one of the punch presses and then in one of the bending presses, we can form flow shops in such a way that each includes one punch press and one bending press. Then, by assigning the jobs to one of these flow shops, we can decrease the number of combinations that will be considered to assign the jobs to one of the two machines of each consecutive stage.

The algorithm given in Section 4 is coded in *C#* and run with an Intel® Core™ i5 CPU / 2.50 GHz / 4 GB RAM under a Microsoft Windows 8 environment. A thorough computational experiment has been conducted to compare the performance of the developed mathematical model and the heuristic algorithm. The makespan values are given in Table 1 based on real data sets.

It is seen that the makespan values obtained from the heuristic algorithm are considerably close to the optimal values in most cases; on the other hand, the computational time of the algorithm is much shorter as the number of the jobs increases. Therefore, the solutions reached by the heuristic algorithm promises a good trade-off and offer a preferable method. Besides, the mathematical model is able to solve the problems up to five stages, eight machines, and six jobs in an acceptable time, yet we are not able to find a solution for seven jobs.

The comparison of the *Cmax* values obtained from the mathematical model with the heuristic algorithm by using Table 1 is given in Figure 5.

Table 1
Comparison of the mathematical model with the heuristic algorithm

Sample no	# of jobs	Makespan ($Cmax$) (min)		Computational time (min)		
		Mathematical model	Heuristic algorithm	Mathematical model	Heuristic algorithm	Difference (%)
1	4	320,50	355,50	0,02	0,02	0,11
2		1901,40	2068,90	0,03	0,02	0,09
3		587,17	633,50	0,02	0,02	0,08
4		398,67	455,92	0,87	0,02	0,14
5	5	320,50	372,92	0,13	0,02	0,16
6		1958,17	2303,27	2,00	0,02	0,18
7		587,17	633,50	0,03	0,02	0,08
8		404,25	528,92	2,62	0,02	0,31
9	6	320,50	372,92	104,92	0,02	0,16
10		2221,6	2587,3	369,20	0,02	0,16
11		591,50	698,00	1161,85	0,02	0,18
12		469,58	556,75	592,27	0,02	0,19

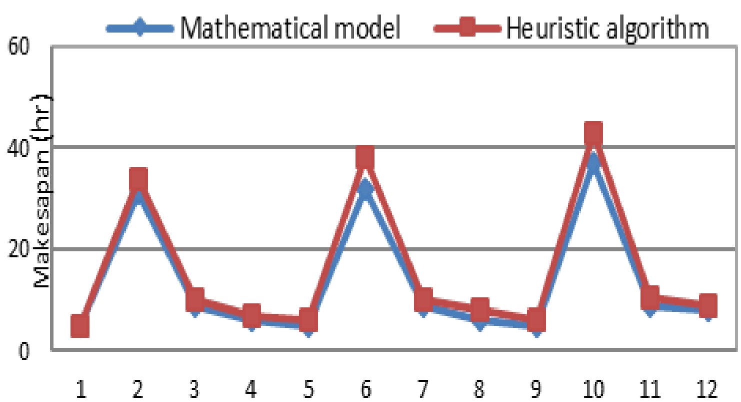

Figure 5. Graphical comparison of the mathematical model with the heuristic algorithm

The performance of the improved heuristic algorithm has also been compared with the results of the current approach applied in the business as seen in Table 2. The monthly production schedule data published by the business are used. The results are exhibited in Table 2.

Table 2
Comparison of the current approach with the heuristic algorithm

Test sample no	Number of jobs	Makespan ($Cmax$) (min) Current approach (approx.)	Makespan ($Cmax$) (min) Proposed approach	Improvement (%)
1	8	1440	1479,33	0
2	17		1324,16	8
3	16		1095,83	24
4	12		1117,87	22
5	8		1323,5	8
6	10	4320	3203,6	26
7	20		2762,7	36
8	24		3773,6	13
9	34		3604,33	17
10	28		3075,83	29
11	39	10080	8371,13	17
12	58		7660,2	24
13	60		6535,48	35
14	101		6772,35	33
15	68		6251,86	38
16	102	21600	16386,95	24
17	148		15368,75	29
18	109		13142,98	39
19	188		12756,1	41
20	147		12530,78	42
21	186	31680	22465,7	29
22	211		20681,97	35
23	183		18125,48	43
24	254		18168,18	43
25	210		18262,37	42
The average improvement in makespan (%)				28

The proposed approach gives, as seen from Table 2, better results in all tests except for Sample 1, and the current schedule is improved by %28 in terms of $Cmax$, averagely. Figure 6 presents $Cmax$ values for both the heuristic and the current approaches.

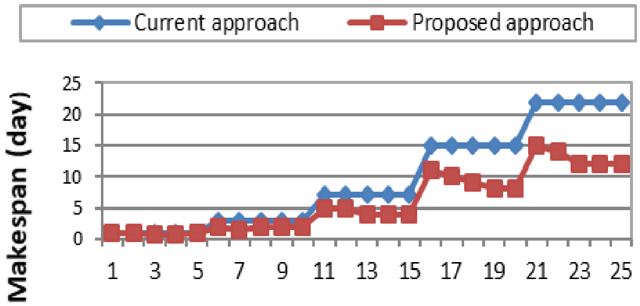

Figure 6. Graphical comparison of the current approach with the heuristic algorithm

As seen from Figure 6 that the proposed approach is capable to produce better quality solutions than the current approach.

6. User-friendly interfaces

An interface, in this context, is a program that allows users to monitor which job is to be processed by which machine. Problem data is loaded to the system. The user, through the interface, can easily upload data from Microsoft Office Excel environment to the system. A warning message is displayed on the screen in case of an error during data entry. Once it is fixed, the error-free data is restored to the system by restarting the program.

The machine names are located in the upper left part of the interface screen. PU1 and PU2 are punch presses in the first stage, where AB1 and AB2 machines are bending presses at the second stage. KH06 and KH07 machines are door lines respectively at third and fourth stages, and HP1 and HP2 machines represent hydraulic presses at the fifth stage. The empty area next to the machine names is designed for the Gantt Chart that is to be generated after running the chosen algorithm by clicking *calculate* button.

Gantt Charts allow users to monitor which job will be processed by which machine and in which sequence. Machine-based job counts, operation and waiting times, machine utilization rates, completion times of the last jobs on the machines, and makespan (C_{max}) can be seen in the *results* section. A job can be tracked by its starting and completion times on the Gantt Chart on a machine, and the time spent on that machine can also be displayed by positioning the mouse on any job. Figure 7 is the screenshot of the interface.

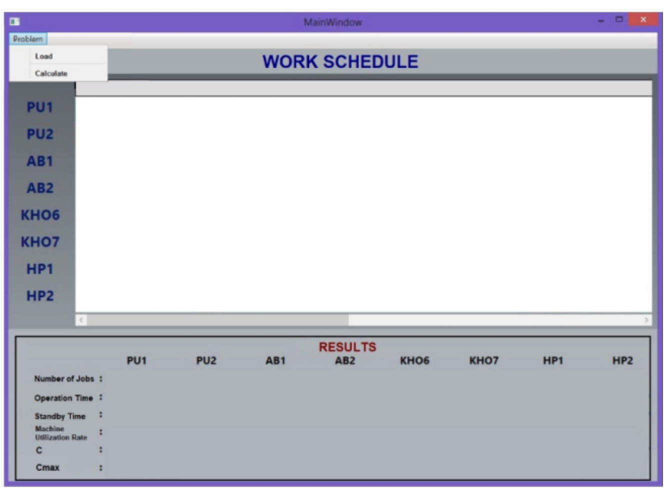

Figure 7. The Image of the Program Interface

In the upper left corner, there is the load button which allows the problem data to be loaded into the system. The calculate button is clicked next, once the data is loaded into the system. This button allows the Gantt Chart to be projected onto the screen. The results appear at the bottom area of the interface. Figure 8 is the screenshot of an interface which exemplifies the scheduling of six jobs.

As can be seen from Figure 8; 4th, 6th, and 1st jobs are scheduled in PU1, while 3rd, 2nd, and 5th jobs are scheduled in PU2 in the first stage. Once the 4th job is completed at the first stage, it is delivered to the second stage, to machine AB1. It is also easy to see how long each machine runs until the end of production, and how long they become idle. In terms of the machine utilization rates, two bending presses operate at close density with approximate occupancy rates of 55% and 59%. The completion times of the last jobs processed in the machines (C values) are also reflected on the screen.

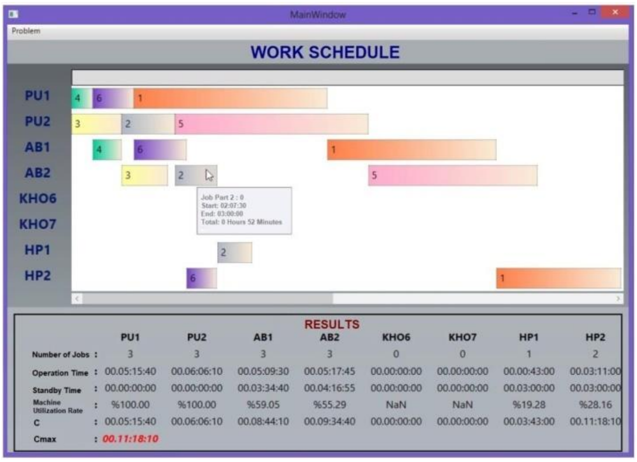

Figure 8. Gantt Chart for a Sample Problem

It can be seen from the section under PU1 in the *results* part that there are 3 jobs assigned to this machine for this schedule. The last job processed in PU1 left this stage at the time of 00:05:15:40 to be processed in the AB1 and completion time is calculated as 00:11:18:10. The statistics for Job 2 which is processed in the AB2 are also visible when the mouse is pointed on this job.

7. Conclusion

In this paper, we have introduced a 0-1 mixed-integer programming formulation and a heuristic algorithm to solve the k-stage hybrid flow shop scheduling problem with sequence-dependent setup times, with the target to minimize the makespan. The heuristic algorithm, based on sub-flow shops via forming machine groups in different clusters, suggests good solutions for large-size problems. To the best of our knowledge, this type of machine grouping algorithm to form flow shops in the machine environment is introduced for the first time and provides more efficient solutions by restricting enumerations.

To the best of our knowledge there is no test problem in the literature for the problem under consideration, therefore a real case study on a refrigerator company, which is one of the leading businesses in Europe, is studied. Real data sets have been established to evaluate the mathematical model and the proposed algorithms. All the numerical results illustrated that the proposed heuristic algorithm gives better performance in terms of computational times in comparison with the mathematical model, and provides nearly optimal solutions. Besides, the mathematical model gives optimal solutions for small-size problems in the business.

The main contribution of this paper can be summarized as follows: 1) Forming flow shops by grouping machines among consecutive stages provides more efficient solutions in terms of computational times due to the reduced enumerations. For the real system under consideration, if there is no flow shop determined, there may be about 90 different routings for the first two stages, on which the products can traverse the hybrid flow shop. 2) The proposed mathematical model can solve the problems with the following configurations: 2 stages, 3 machines, 8 jobs, or 4 stages 5 machines, 7 jobs or 6 stages, 7 machines, 6 jobs. The problem includes symmetry which is known to complicate the problem because of identical machines in the stages. We created flow shops in our heuristic approach to cope with this. On the other hand symmetry breaking constraints should be considered for further researches. 3) The proposed heuristic approach provides a promising contribution to the real system in terms of makespan values. It is seen that the proposed heuristic contributes from % 8 to % 35 better C_{max} values. 4) The algorithm considers SDST which is relatively less studied in the literature. 5) The proposed heuristic approach considers stage skipping. 6) A user-friendly environment and dynamic scheduling is provided. The system can modify the current schedule when there are new arrivals or job cancellations. It allows users to directly import demand data from the production program. Besides, new jobs and/or machines can easily be included in the system, since all process-specific parameters, such as processing times, stage-based SDST, are stored in the Microsoft Excel environment.

Considering the NP-hard nature of the defined problem, it is difficult to solve large-size problems by mathematical models developed. Yet, new scientific techniques can be developed targeting the solution processes of these models in the future. In this respect, this model encourages researchers for further improvements.

It could be interesting to apply the proposed algorithm to other hybrid flexible flow shop problems. Some other criteria, such as the total weighted earliness and tardiness, flow time can be considered as well. A limited buffer capacity between machines can be assumed. Finally, performing simulation studies together with dispatching rules may be more practicable.

Conflict of Interest

No conflict of interest was declared by the authors.

Credit Author Statement

Müjgan Sağır: Conceptualization, Methodology, Modeling, Data Curation, Software, Validation, Writing-Reviewing and Editing; Hacer Defne Okul: Methodology, Modeling, Data Curation, Software, Writing, Validation.

Acknowledgments

This study is supported by Eskisehir Osmangazi University Scientific Research Projects Committee (Ogubap-Project No: 2014-176).

References

Allahverdi, A., Ng, C., Cheng, T., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research,* 187 (3), 985–1032. Doi : https://doi.org/10.1016/j.ejor.2006.06.060

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research,* 246 (2), 345–378. Doi : https://doi.org/10.1016/j.ejor.2015.04.004

Andrés, C., Albarracín, J.M., Tormo, G., Vicens, E., García-Sabater, J.P. (2005). Group technology in a hybrid flowshop environment: A case study. *European Journal of Operational Research*, 167, 272–281. Doi : https://doi.org/10.1016/j.ejor.2004.03.026

Azadeh, A., Goodarzi, A.H., Kolaee, M.H., Jebreili, S. (2019). An efficient simulation-neural-network genetic algorithm for flexible flow shops with sequence-dependent setup times, job deterioration and learning effects. *Neural Computing & Applications*, 31, 9, 5327-5341. Doi : https://doi.org/10.1007/s00521-018-3368-6

Cai, J.C., Zhou, R., Lei, D.M. (2020). Fuzzy distributed two-stage hybrid flow shop scheduling problem with setup time: Collaborative variable search. Journal of Intelligent &Fuzzy Systems, 38, 3, 3189-3199. Doi : https://doi.org/10.3233/JIFS-191175

Dudek, R., Smith, M., Panwalkar, S.(1974). Use of a case study in sequencing-scheduling research. *Omega,* 2, 253–261. Doi : https://doi.org/10.1016/0305-0483(74)90094-2

Gupta, J.N.D. (1988). Two-stage, hybrid flow shop scheduling problem. *Journal of the Operational Research Society*, 39, 359-364. Doi : https://doi.org/10.1057/jors.1988.63

Gholami, Z., Seyed M., Hakimifar, M., Nazemi, N., Jolai, F. (2017). Robust and Fuzzy Optimisation Models for a Flow Shop Scheduling Problem with Sequence-dependent Set Up Times: A real case study on a PCB assembly company. *International Journal of Computer Integrated Manufacturing*, 30, 552-563. Doi : https://doi.org/10.1080/0951192X.2016.1187293

Haouari, M., M'Hallah, R. (1997). Heuristic algorithms for the two-stage hybrid flow shop problem. *Operations Research Letters*, 21, 43-53. Doi : https://doi.org/10.1016/S0167-6377(97)00004-7

Jabbarizadeh, F., Zandieh, M., Talebi, D. (2009). Hybrid flexible flow shops with sequence-dependent setup times and machine availability constraints. *Computers & Industrial Engineering*, 57, 949–957. Doi: https://doi.org/10.1016/j.cie.2009.03.012

Jungwattanakita, J., Chaovalitwongsea, P., Wernerb, P. (2009). A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Computers & Operations Research*, 36, 358 – 378. Doi : https://doi.org/10.1016/j.cor.2007.10.004

Kurz, M.E., Askin, R.G. (2003). Comparing scheduling rules for flexible flow lines. *International Journal of Production Economics*, 85, 371-388. Doi : https://doi.org/ 10.1016/S0925-5273(03)00123-3

Morais, M.D., Godinho, M., Boiko, T.J.P. (2013). Hybrid flow shop scheduling problems involving setup considerations: a literature review and analysis. *International Journal of Industrial Engineering Theory Applications and Practice*, 20, 614-630. Retrieved from : http://apps.webofknowledge.com/full_record.do?product=WOS&search_mode=GeneralSearch&qid=1&SID=E33qxsfjVeRX5aVTvvG&page=1&doc=1

Naderi, B., Zandieh, M., Shirazi, M.A.H.A. (2009a). Modeling and scheduling a case of flexible flow shops: Total weighted tardiness minimization. *Computers & Industrial Engineering*, 57, 1258–1267. Doi : https://doi.org/10.1016/j.cie.2009.06.005

Naderi, B., Zandieh, M., Balagh, A.K.G., Roshanaei, V. (2009b). An improved simulated annealing for hybrid flow shops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert Systems with Applications*, 36, 9625-9633. Doi : https://doi.org/10.1016/j.eswa.2008.09.063

Portmann, M.C., Vignier, A., Dardilhac, D., Dezalay, D., (1998). Branch and bound crossed with GA to solve hybrid flow shops. *European Journal of Operational Research,* 107, 389-400. Doi : https://doi.org/10.1016/S0377-2217(97)00333-0

Pan, Q.K., Gao, L., Li, X., Gao, K. (2017). Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times. *Applied Mathematics and Computation*, 303, 89-112. Doi : https://doi.org/10.1016/j.amc.2017.01.004

Pinedo, M. (2002). *Scheduling: Theory Algorithms and Systems*, Second Ed., Prentice-Hall, Upper Saddle, NJ.

Ribas, I., Leisten, R., Framinan, J.M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers and Operations Research*, 37, 1439-454. Doi : https://doi:10.1016/j.cor.2009.11.001

Ruiz, R., Şerifoğlu, F.S., Urlings, T. (2008). Modeling realistic hybrid flexible flow shop scheduling problems. *Computers & Operations Research*, 35, 1151-1175. Doi : https://doi.org/10.1016/j.cor.2006.07.014

Ruiz, R., Maroto, C. (2006). A genetic algorithm for hybrid flow shops with sequence-dependent setup times and machine eligibility. *European Journal of Operational Research*, 169, 781-800. Doi : https://doi.org/10.1016/j.ejor.2004.06.038

Sioud, A., Gagné, C. (2018). Enhanced migrating birds optimization algorithm for the permutation flow shop problem with sequence-dependent setup times. *European Journal of Operational Research*, 264, 66-73. Doi : https://doi.org/10.1016/j.ejor.2017.06.027

Tian, H., Li, K., Lius, W.(2016). A Pareto-Based adaptive variable neighborhood search for bi-objective hybrid flow shop scheduling problem with Sequence-Dependent Setup Time. *Mathematical Problems in Engineering*, Volume 2016, Article ID 1257060, 11 pages. Doi : https://doi.org/10.1155/2016/1257060

Zandieh, M., Ghomi, S.M.T., Husseini, S.M. (2005). General models for hybrid flow shop sequencing problems with sequence-dependent setup times. *Journal of Faculty of Engineering*, 39, 1-9. Retrieved from : https://www.sid.ir/En/Journal/ViewPaper.aspx?ID=41251

Zandieh, M., Ghomi, S.M.T., Husseini, S.M. (2006). An immune algorithm approach to hybrid flow shops scheduling with the sequence-dependent setup times. *Applied Mathematics and Computation*, 180, 111–127. Retrieved from https://www.researchgate.net/publication/313073224_An_immune_algorithm_approach_to_hybrid_flow_shops_scheduling_with_sequence-dependent_setup_times

Zhu, X., Wilhelm, W. E. (2006). Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE Transactions*, 38, 11, 987–1007. Doi : https://doi.org/10.1080/07408170600559706

Qin, W., Zhuang, Z.L., Liu, Y., Tang, O. (2019). A two-stage ant colon algorithm for hybrid flow shop scheduling wiht lot sizing and calender constraints in printed circuit board assembly. Computers & Industrial Engineering, 138, Article ID 106115. Doi : https://doi.org/10.1016/j.cie.2019.106115