TEKSTİL VE KONFEKSİYON

VOL: 32, NO. 1

DOI: 10.32710/tekstilvekonfeksiyon.924028

A Novel Approach for Optimum Planning of Bobbin Boilers in Textile Industry

Yunus Demir[1] 0000-0003-3868-1860
Kemal İnan[2] 0000-0000-0000-0000

[1]Bursa Technical University /Department of Industrial Engineering / Bursa / Türkiye
[2]Bursalı Textile / Plant Production Group Manager / Bursa / Türkiye

Corresponding Author: Yunus Demir, yunus.demir@btu.edu.tr

ABSTRACT

As in all sectors, the fierce competition in the world affects the textile sector deeply. It is an inevitable necessity for companies to reduce their costs in order to survive while maintaining their profitability. In this study, an applied efficiency study was conducted for the bobbin dyeing process, which is one of the important elements of the textile industry. In the application made in Bursalı Tekstil Ltd. Şti., the problem of sequencing the lots waiting to be dyed in the bobbin dye boilers has been discussed. Integer linear model and iterative greedy-based heuristic are proposed to solve the problem. Sequencing made with the developed method resulted in an improvement of 17% in terms of the number of boiler washes and 13% in terms of total tardiness compared to the manual sequencing.

ARTICLE HISTORY
Received: 27.04.2021
Accepted: 07.10.2021

KEYWORDS

Bobbin (package) yard dyeing, sequencing, integer linear programming, meta-heuristics

1. INTRODUCTION

In this article, an optimization study has been carried out for the bobbin dyeing process, which is an important sub-process of textile products. Bobbin dyeing is one of the most common yarn dyeing techniques. Yarn dyeing process has been defined by [1] as follows: "The dyeing process performed after the fiber is spun into yarn and before it is made into woven or knitted fabric is called yarn dyeing". In the whole production process, yarn dyeing is evaluated under the subtitle of textile finishing. Finishing process covers activities (bleaching, mercerizing, dyeing, washing, rinsing, softening, finishing and drying) that aim to give the textile material the desired properties (non-flammability, appearance effect, waterproofness, etc.) within the framework of customer demands [2]. Dyeing process is defined as treating the textile product with a dye solution and various auxiliary chemicals (wetting agent, salt, alkali and acid) [3]. Dyeing processes are classified as open fiber & stock dyeing, tops dyeing, tow (cable) dyeing, yarn dyeing, piece (fabric) dyeing and dyeing of ready-made products (finished garments, carpets, etc.) [2]. The yarn dyeing process can be done in four different ways: bobbin dyeing, hank dyeing, muff dyeing and warp beam dyeing [4].

The scope of this study focused on the package dyeing process. The bobbin dyeing process basically operates as follows: By combining the yarns wound in cops, bobbins weighing 1.8–2.0 kg are formed [5]. Yarns prepared in bobbin form are wrapped in perforated dyeing bobbins. Coils are placed evenly and neatly on the lances on the portmaterial. The portmaterial consists of hollow spindles designed to allow the paint to pass from the inside to the outside and from the outside to the inside. Later, if the port material is to be placed in a vertical bodied bobbin dyeing machine, it is placed in the dyeing boilers with the help of a crane, if it is placed in the horizontal body bobbin dyeing machine, it is placed in the dyeing boilers using wheeled systems. Boilers perform the painting process indoors, under high pressure and at high temperatures (135-140°C) [6].

Operations research studies in the textile sector are generally in the form of parameter (process) optimization [7-9] or multi-criteria decision making [10, 11]. Linear

To cite this article: Demir Y, İnan K. 2022. A novel approach for optimum planning of bobbin boilers in textile industry. *Tekstil ve Konfeksiyon* 32(1), 24-36.

optimization studies using mathematical programming or heuristic techniques [12, 13] are relatively rare. In this study, two different solution approaches from linear optimization techniques are proposed to the problem of sequencing the jobs waiting to be dyed in bobbin dye boilers in a way that minimizes two objectives. The first objective is minimizing the total tardiness (on time delivery) and the second is minimizing the total number of boiler washes (sequence dependent setup time). In the literature, this problem is evaluated within the scope of textile batch dyeing scheduling problem. In terms of textile products, various studies have been carried out in two different groups as fabric dyeing [14-17] and yarn dyeing [18]. This problem consists of two sub-problems. The first sub-problem is batching the jobs (orders) into lots, the second sequencing the lots in the boilers. The second sub-problem is referred to as the sequence dependent setup time scheduling problem (Detailed literature is presented in Table 1). This second sub-problem appears in two different ways in the textile literature as single machine [19] and parallel machine [20-22] scheduling problem according to machine environments. Two basic approaches are followed for the solution of the problem: simultaneous [14, 21] and hierarchical [18, 17] approaches [21]. Although studies in this area generally aim for economic benefits, environmental effects such as water use/waste water discharge [22, 23], forklift routing [15], periodic maintenance [19] have also been taken into account in some studies. In the studies in the literature, setup (boiler washing) is required for all color transitions, but in this study, the necessity of setup depends on a special rule explained in the second chapter. In order to accelerate the convergence in the proposed solution method, a novel neighborhood structure has been developed, inspired by this special structure of problem. In the following part of this chapter, the concepts related to setup activities, one of the basic elements of the production process, are clarified.

Setup time is defined as the time required to prepare the relevant resource (machine) for the operation [24]. Setup activities include activities such as positioning the workpiece, cleaning the machine, setting the necessary equipment and fixtures, examining the material, and preparing the appropriate environment for the process [24, 25]. Setup is basically divided into two groups as sequence dependent and sequence independent. In sequence dependent setup, the setup time is variable according to the sequentially processed jobs. In sequence independent setup this period varies only according to the job to be processed (independent from the previous one) [26].

To reduce setup time, businesses tend to produce large volumes. Although this increases the rate of utilization, it is not an approach recommended especially according to lean production principles since it causes unnecessary inventory. Another way is to divide the jobs into families according to similar production needs. This approach, which does not require setup between consecutive jobs belonging to the same family and is called the principle of group technology, is a very effective method when the delivery dates of the works are not taken into account. However, in terms of maintaining customer satisfaction, compliance with deadlines is an element that cannot be ignored. Therefore, such problems consist of two sub-problems: dividing product families into several lots (batching) and sequencing of jobs [27].

In this study, a real-life problem with family setup time, which is a special case of sequence dependent setup, is discussed. In this type of problem, it does not require between jobs in the same family while it requires setup between jobs belonging to different families. It has been observed in the literature that family setup is evaluated in three different ways. The first of these is called family sequence-dependent setup. In this type, setup time varies according to k and l families processed in consecutive order (s_{kl}). The second is called family dependent or family sequence independent setup. In this second type the setup time varies according to the family to which the job belongs (s_k - independent from the previous one). And the third type of setup time is a fixed time, regardless of the ordinary and family (s). In the literature; although it is seen that family setup is considered for different scheduling environments such as parallel machine [28, 29], flow shop [30] and flexible job shop [31] this situation is mostly encountered in single machine scheduling problems.

The literature review on single machine scheduling problems with family setup time is classified according to certain parameters and presented in Table 1. In the first column of the related table, the authors who made the study are cited. In the second column, the type of family setup is specified. s_{kl}: family sequence dependent setup, sk: family sequence independent (family dependent) setup, and s: fixed time setup. In the third column, the situation whether the study is single-objective or multi-objective is stated. In the fourth column, the objective(s) to try to optimize is given. The proposed solution method is given in the fifth column. In the last column, application area of study or an important point about the study has been shared.

The problem addressed in this study is inspired by the planning process of bobbin (yarn) dye boilers in the textile industry. In the problem dealt with, the process order of the yarn lots waiting to be dyed in the boilers is tried to be determined. Jobs (batches) are divided into families according to the color darkness level. While no setup is required between sequential jobs in the same family, fixed-time setup (2 hours for boiler washing) may be required between sequential jobs between different families. In these problems coded as $1|s_{kl}|C_{max}, \Sigma T$, both the total setup time and the total tardiness are tried to be minimized. However, since the minimization of total delay covers both objectives, the studies in the literature have been arranged in a single-objective manner. Such an approach is insufficient in minimizing the total setup time when the total delay is 0. In this study, both objectives were evaluated, with total tardiness being the priority. Both goals are trying to be minimized. To the best of our knowledge, such a dual objective study in single machine family type setup time scheduling problems has not been done before. Details of the problem are given in section 3.

2. MATERIAL AND METHOD

This section is organized under two subtitles. In the first subtitle, the definition of the problem is given, and in the second subtitle, the solution methods are presented.

2.1. Definition of the Problem

The problem addressed in this study is a real-life problem. The study was carried out in Bursalı Tekstil San.Tic. A.Ş yarn dyeing workshop. In this study, in which the processing order of the batches waiting to be dyed in yarn dye boilers is determined, the aim is to minimize the total tardiness and the total number of boiler washing (Since the sequence dependent setup time is fixed-2 hours, the total setup time and the total number of setups are equivalent.).

Table 1. Literature review of single machine scheduling with sequence dependent set up

Reference	Type of Setup	Single / Multi	Objective(s)	Solution Method	Application Area (Explanation)
[32]	s_{kl}	Single	Total tardiness	Iterated greedy heuristic	steel production, multi-stage production processes
[26]	s_k, s	Single	Total tardiness	composite dispatching procedure	news-publisher
[33]	s_{kl}	Single	Total tardiness	Heuristic solution approach, MIP	Coninuous casting stage of steel production
[25]	s_{kl}	Single	Total tardiness	Iterated local search	
[34]	s_{kl}	Single	Maximum lateness	Hybrid genetic algorithm	release dates considered
[35]	s_{kl}	Single	Maximum lateness	Tabu search heuristic	Motivated steel wire production
[36]	s_{kl}	Single	Maximum lateness	Simulated annealing algorithm	Motivated steel wire production
[37]	s_{kl}	Single	Maximum lateness	Tabu search	
[38]	s_k	Single	Total tardiness	MIP, several heuristic algorithms	
[39]		Single	Min. total flow time, total tardiness (Separately)	branch-and-bound algorithm	
[40]	s_{kl}	Single	Total earliness and tardiness	branch-and-bound algorithm, problem specific heuristic	
[41]	s_k	Single	Total weighted tardiness	Tabu Search	motivated by resin manufacturing
[42]	s_k	Single	Maximum lateness	incomplete dynamic program, problem specific heuristic	turning center where rods of different diameters are processed
[43]	s	Single	Maximum lateness	neighborhood search procedure	rolling of steel strips, dyeing operations in the textiles
[44]	s_k	Single	Total tardiness	branch-and-bound algorithm, problem specific heuristic	
[45]	s_{kl}	Multi	Total completion time and total tardiness	integer programming model, tabu search	
[46]	s_k	Single	Cmax	Dynamic programming, problem specific heuristics	release dates considered
[47]	s_{kl}	Single	Total tardiness	Dynamic programming	
[48]	s_{kl}	Single	sum of completion times	branch-and-bound algorithm	
[49]	s_k	Single	total earliness and tardiness	problem specific heuristics	
[50]	s_k	Single	Maximum lateness	problem specific heuristics	
[51]	s	Single	Maximum lateness	problem specific heuristics	
[52]	s_k	Single	Maximum lateness	branch and bound algorithm, problem specific heuristics	
[53]	s_k	Single	Maximum lateness	branch and bound algorithm	release dates considered
[54]	s_k	Single	Number of late jobs	multi-start descent, simulated annealing, tabu search and a genetic algorithm	

The amount of yarn in various colors required for towel weaving is calculated by the MRP system and delivered to the yarn dye shop. Since the required yarn quantities for each towel order cannot fill the dye boilers, yarns belonging to different towel orders with the same color are combined and dyed in same boiler. Within the scope of the study, the issue of ranking the batches formed as a result of merging was discussed and the merging process was not included. The general operation of the process is presented in Figure 1. The delivery date of each batch is determined as the lowest of the delivery dates of the works it contains. The dyeing time of the batches varies according to the darkness of the color to be dyed. Colors are divided into 5 groups (family) according to their darkness (1- white, 2-light, 3-middle, 4-dark, 5-turquoise). The processing times of the batches in the same family are the same. In addition, if the batches belong to the same family, there is no need to wash the boiler (setup process) between the batches to be processed consecutively. In all of the studies in the literature, varying or fixed-time setup is required for sequential jobs belonging to two different families. However, in this study, fixed-time setup is applied for transitions between certain families. The process for batches of different families to be dyed in consecutive order is as follows:

- If the post batch is darker than the predecessor batch, the boiler is not washed (The darkness / lightness levels of the colors are determined by the laboratory according to the recipe and are graded from 1 to 5 (light: 1, dark: 4). Grade 5 colors are a special class).
- If the predecessor batch is one degree darker than the consecutive batch, it is not necessary to wash the boiler again. However, if the preliminary batch is more than a degree darker than the consecutive batch, 2 hours washing (setup) time is required. For example, if a batch of 3rd degree darkness is to be dyed after the dyeing process of a 4th degree dark batch, it is not necessary to wash the boiler. However, if a batch with a dark level of 2 is to be dyed after a batch with a 4th degree darkness, the boilers must be washed.
- The fifth degree dark family is a special class (turquoise) and every pass except this group requires 2 hours of setup time. The boiler does not need to be washed for the passages of these batches. The required setup time and processing times according to the color darkness levels are given in the Table 2.

In summary, in the problem, the dyeing order of the batches in the boilers is tried to be determined by taking into account the due dates and the requirement for washing the boiler. The assumptions regarding the problem under consideration are as follows:

- Only one batch can be dyed in one boiler at a time.
- All batches are ready to be processed at the time of sorting ($r_j = 0$).
- A batch that has started to be dyed is not interrupted in any way until it is completed.

Table 2. Required ready times according to color darkness

Darkness	**1**	**2**	**3**	**4**	**5**
1	0	0	0	0	0
2	0	0	0	0	0
3	2	0	0	0	0
4	2	2	0	0	0
5	2	2	2	2	0
$\mathbf{p_j}$	5	8	10	12	14

2.2. Proposed Solution Methods

In this section, the methods presented for the solution of the problem dealt with are mentioned. First, integer linear model developed is explained. Later, the iterated greedy heuristic is included.

2.2.1. Integer linear programming model

In this subsection, an integer linear programming model based on assignment and positional date formulation [55] developed for the solution of the problem is explained. First, the indices, parameters and decision variables used in the model were explained, and then the mathematical formulation was included. The model was originally designed to sequence jobs in all boilers at the same time. However, due to the difficult nature of the problem, no solution could be obtained even for small-sized samples. Therefore, the model has been rearranged to be applied separately for each boiler i. This is because of the "i" phrase on the variables.

Indices

a: Order index (1,2,…, nmri)

i: Boiler index

j, f: Batch index

k, l: Family index (1, 2, 3, 4, 5)

Parameters

p_{ij}: processing time of batch j

due_{ij}: Due date of batch j

dns_{ij}: Color darkness of batch j (1-5)

nmr_i: Total number of batches to be processed in boiler I

M: A very large number

Variables

n_{ija}: If batch j is processed in ath order in boiler i, then 1
Otherwise,0

x_{ijf}: if batch f is processed immediately after batch j in boiler i, then 1
Otherwise, 0

pnt$_{ijf}$: If batch i that comes immediately after batch j causes the boiler to be washed, then 1

Otherwise, 0

trd$_{ij}$: Tardiness of batch j dyed in boiler i. It is tried to be minimized in the objective function.

With the constraint 1, it is ensured that each batch j is assigned to a position (order) of the boiler i. In constraint 2, it is ensured that only one batch is assigned to each position a of each boiler i. Constraints 3-6 are the linearized form of the nonlinear constraint (14) given below. dmy$_{ifa}$ is the auxiliary variable used for linearization.

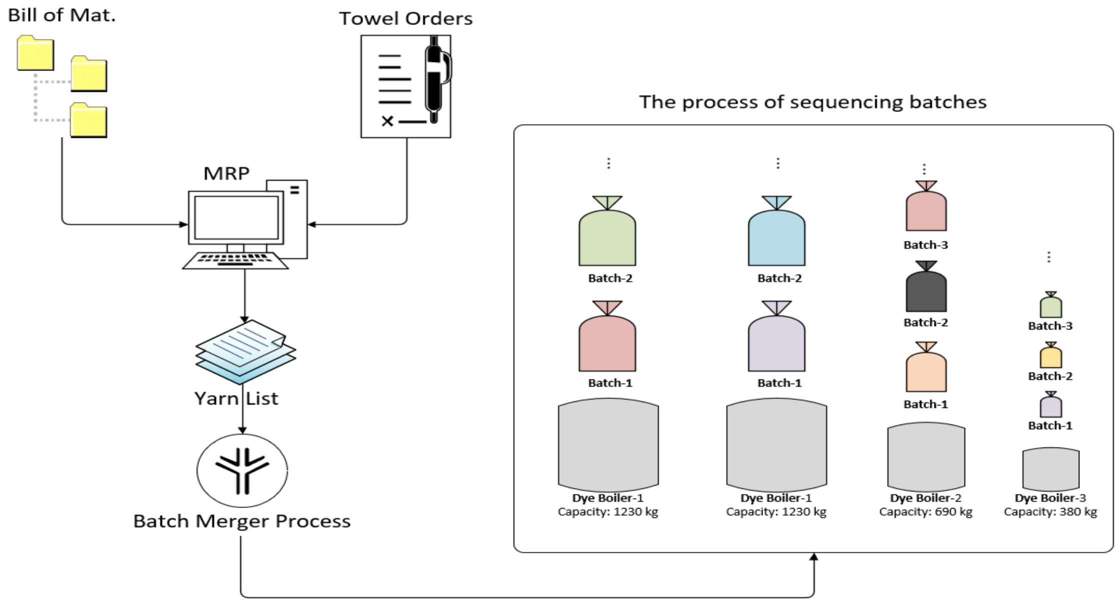

Figure 1. The general operation of the process

<u>Model Formulation</u>

$$\sum_{a \le due_j} n_{ja}^i = 1 \qquad \forall j \qquad (1)$$

$$\sum_j n_{ja}^i = 1 \qquad \forall a \le nmr_i \qquad (2)$$

$$dmy_{jfa} \le n_{ja}^i \qquad \forall j,f;j \ne f;\forall a \le nmr_i - 1 \qquad (3)$$

$$dmy_{jfa} \le n_{fa+1}^i \qquad \forall j,f;j \ne f;\forall a \le nmr_i - 1 \qquad (4)$$

$$dmy_{jfa} \ge \left(n_{ja}^i + n_{fa+1}^i\right) - 1 \qquad \forall j,f;j \ne f;\forall a \le nmr_i - 1 \qquad (5)$$

$$\sum_a^{nmr_i - 1} dmy_{jfa} = x_{jf}^i \qquad \forall j,f;j \ne f \qquad (6)$$

$$x_{jf}^i * \left(dns_j^i - dns_f^i\right) \le 1 + pnt_{jf}^i * M \qquad \forall j,f;j \ne f;j \ne 5 \qquad (7)$$

$$x_{jf}^i * \left(dns_j^i - dns_f^i\right) \le pnt_{jf}^i * M \qquad \forall j,f;j \ne f;j = 5 \qquad (8)$$

$$Cp_a \ge \sum_j (p_j^i * n_{ja}^i) \qquad \forall a \le nmr_i \qquad (9)$$

$$Cp_a \ge Cp_{a-1} + \left(n_{ja-1}^i + n_{fa}^i - 1\right) * \left(p_f^i\right) + 2 * pnt_{jf}^i \qquad \forall j,f;j \ne f;\forall a - \{1\} \qquad (10)$$

$$Cb_j \ge Cp_a - M * (1 - n_{ja}^i) \qquad \forall j;\forall a \le nmr_i \qquad (11)$$

$$trd_j^i \ge Cb_j - due_j^i \qquad \forall j \qquad (12)$$

$$z^i = M * \sum_j trd_j^i + \sum_j \sum_f pnt_{jf}^i \qquad (13)$$

$$s_{ijf} = \sum_{a}^{nmr_i - 1}(n_{ija} * n_{ifa+1}) \quad (14)$$

With these constraints, the value of the s_{ijf} variable is determined. The value of this variable is 1, which means that the batches j and f assigned to the same boiler i are processed consecutively (j predecessor). In constraints 7 and 8, it is determined whether the batch f, which comes just after the batch j, causes the washing of the boiler. In the constraint 7, the boiler washing situation caused by the successive works that passes from two degrees darker to lighter in terms of color darkness has been formulated. If the color darkness difference of batches j and f ($dns_j^i - dns_f^i$) to be dyed consecutively in the same boiler i is 1 or 0, variable pnt_{jf}^i takes the value 0. So there is no need to wash the boiler. Otherwise, if the difference in color darkness of batches ($dns_j^i - dns_f^i$) j and f is more than 1, the pnt_{jf}^i variable takes the value 1, which means that the boiler will be washed. The situation belonging to the special group in the fifth family in constraint 8 has been formulated. In this case, setup is required for every transition outside the same family (fifth family). In constraint 9, the completion time of the job in the ath order (cp_a) is guaranteed to be at least the processing time of the relevant job. In constraint 10, it is ensured that between the completion times of the jobs processed in the ath and (a-1)th orders, at least as long as the process time of the job processed in the ath row (additionally if necessary, the setup time). In constraint 11 completion time of job processed in the ath order (cb_j) is found. In constraint 12 tardiness of batch j is calculated. Constraint 13 is objective function. This function consists of scalar sum of total tardiness and total number of setup. But it is weighted with a large number such as M to prioritize the total tardiness.

2.2.2. Iterated greedy heuristic based algorithm

In this section, the iterated greedy heuristic based method developed for the solution of the problem handled is mentioned. Iterative greedy heuristics (IGH) was first used by [56]. IGH is a heuristic method for combinatorial problems that is highly effective and easy to apply. IGH first creates an initial solution (S_0) using a constructive procedure. Then, the initial solution created goes through four phases and tries to improve it iteratively. The first of these phases is the destruction stage. In the destruction phase, a partial candidate solution (S_d) is created by removing a certain number of elements from the incumbent solution (S_i). In the second phase (construction), the extracted elements are added to the partial candidate solution with a greedy constructive heurist until a complete solution (S') is rebuilt. In the third stage, a solution is tried to be developed by applying the local improvement procedure to the re-created solution. In the last stage, the incumbent solution (Si) is replaced with the newly formed solution (S_c) according to a certain acceptance criteria. These four stages are continued until the predefined stop criteria are met. The pseudo code showing the general operation of IGH is given in Figure 2.

The proposed method is designed in two stages. First, the minimum number of tardy jobs is controlled by the Hodgson-Moore algorithm [57], neglecting sequence-dependent setup. Hodgson-Moore algorithm gives optimum results for the minimum number of tardy jobs in single machine scheduling (1 || nt). If the number of tardy jobs (ntj) is zero, the total number of boiler washing is minimized by keeping the total tardiness zero. A special local search procedure has been developed for this. Since the number of tardy jobs is 0, neighbors that minimize the total boiler washing are sought so that the total delay tolerance (tlr) is 0 (that is, keeping the total tardiness 0). In the first stage, if the minimum number of tardy jobs is greater than zero, two options are offered to the user. In the first (user preference = 1), the tardy jobs (tjs) are added to the job list (ctrList) to be sent to the subcontractor by leaving the list, and the local search procedure developed (tlr = 0) is applied to minimize the total number of boiler washing for the remaining jobs (njs). In the other option (user preference = 2), an iterative greedy algorithm is applied to minimize both goals. The second option is recommended for businesses with flexibility against tardiness. The pseudo code for the proposed approach is given in Figure 3.

In the proposed approach, permutation method, one of the most widely known representation types in the literature, is used. In this approach, a feasible solution is represented by a series of numbers corresponding to job indices. S_i: {2, 1, 3, 8, 7, 9, 5, 4, 10, 6} is a feasible solution for the sample problem, whose data is given in Table 3. For this feasible solution, a total of two boiler washes (between 1-3 and 9-5 jobs) and a total tardiness of 28 hours are calculated.

```
Procedure: Iterated Greedy Heuristic
    S0:= Generate initial solution
    Si:= Apply local search (S0)                          % Optional
    While (stopping criterion is not fulfilled)
        SD:= Destruction (Si)
        S':=  Construction (SD, SR)
        SC:= Local search (S')                            % Optional
        Si:=  Acceptance criterion (SC, Si)
    endWhile
```

return $S_{max/min}$

Figure 2. The pseudocode of iterative greedy heuristic

```
Procedure: Proposed approach (pd, up, tlr)
    tjs, ntj, njs:= HodgsonMoore(pd)
    if(ntj==0)
        njs:= FamilyBasedLocalSearch(njs, tlr)
    else
        if(up==1)
            ctrList= tjs
            njs:= Family Based Local Search(njs, tlr)
        else
            S0:= Generate initial solution
            Si:= Family Based Local Search(S0, tlr)
            While
                SD:= Destruction (Si)
                S':=  (Re)Construction (SD, SR)
                SC:= Family Based Local Search(S', tlr)
                Si:=  Acceptance criterion (SC, Si)
            endWhile
            return Smax/min
        endif
    endif
```

pd: problem data, up: user preference, tlr: delay tolaerance

Figure 3. Pseudo code of the proposed approach

Table 3. Data for sample problem

j	1	2	3	4	5	6	7	8	9	10
dns_j	5	5	1	2	1	3	4	1	3	4
p_j	14	14	5	8	5	10	12	5	10	12
due_j	82	99	83	92	46	98	65	29	91	118

For the initial solution, earliest due date constructive heuristic (dispatching rule) was preferred, which guarantees the optimum for the maximum tardiness objective in single machine scheduling problem. Then, the destruction and reconstruction phase was applied to the initial solution. During the destruction phase, randomly selected $d*S_i$ jobs are extracted from the complete solution. The level of destruction d takes value in the range [0, 1]. After the destruction phase, two sub-sets are formed. There are extracted jobs (S_d) in the first sub-set. In the second sub-set is the remaining jobs (S_r) after subtraction. During the reconstruction phase, two different greedy constructive heuristics were created to create a new solution (S'). In the first approach, the sub-set that is removed (S_d) is added to all positions of the remaining sub-set (S_r) as a whole. Every solution that occurs in this way has been named S'candidate. Then all elements in the sub-set S_d are also tested for all positions of S'candidate. The pseude code for the reconstruction procedure called classical approach is given in Figure 4.

In addition, the operation of the classical approach is schematized in Figure 5 on a small sample with seven threads. Jobs denoted by a, b and c form the S_d sub-set. Numbers (1-7) refer to the elements belonging to the S_r sub-set.

```
Procedure: Classical (re)construction
Best=M
    for i:=1 to | Sr | do
        S':= Sr + Sd
        for j:=1 to | Sd | do
            for k:=1 to | S' | do
                if(Best>S'candidate)
                    S':=S'candidate
                    Update Best
                endif
            endfor
        endfor
    endfor
```

```
return S'
```

Figure 4. Classic (re)construction process pseudo code

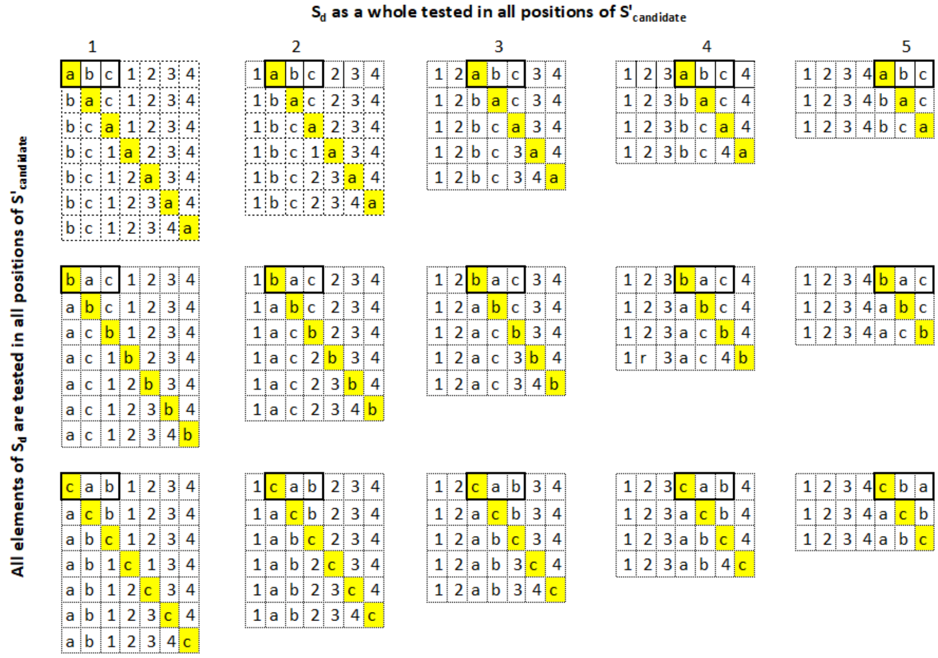

Figure 5. Functioning of the classical (re)construction approach

In the second approach used for reconstruction, all elements in the subset Sd are tested for positions randomly determined from S_r. This approach, which is named as random reconstruction, is repeated n times and the best solution reached is accepted as the new solution.

The algorithm has been tested for both reconstruction approaches and the results are shared in Table 4. For small and medium sized problems, the classical approach gives an average of 4% better results than the random approach. Both approaches give almost the same result in large scale problems. However, the random approach seems to give results significantly faster. According to these results, classical method was used for small and medium scale problems, and random method was used for large scale problems.

A new neighborhood structure has been developed that will minimize the total number of washings by preserving the total tardiness value reached after the destruction and construction phase. This structure, which is called family-based neighborhood, is based on the principle of changing the list of jobs according to a certain rule in order to reduce the number of washing machines without additional tardiness. For this, three types of jobs are defined as α, β and γ. Here, α and β are two jobs (α job predecessor and β job successor) waiting to be processed consecutively. Boiler washing is required to start the job β. So, the color darkness level of job α is at least two degrees higher than the job β. Job γ is a candidate job that can start after the job α (that is, the level of darkness is more than α, the same or at most one degree lower) without the need for boiler washing and can replace the job β. The γ job is searched up or down in the list according to the β job. The general structure of the neighborhood structure developed is given in the Figure 6.

Table 4. Comparison of classical and random (re)construction approach

	Classic		Random		
(n, r)	**Avrg. Tard.**	**CPU**	**Avrg. Tard.**	**CPU**	**Tard. Dev.**
(10, 0.5)	268	1	281	1	4,50%
(15, 0.5)	503	2,33	521	2,33	3,45%
(20, 0.5)	789	8,6	839	4	5,96%
(30, 0.5)	2020	51	2098	6	3,72%
(40, 0.5)	3490	165	3541	8,33	1,44%
(50, 0.5)	6267	483	6312	11	0,71%
(60, 0.5)	8215	1099	8234	13	0,23%

(80, 0.5)	14708	2008	14737	19	0,20%
(100, 0.5)	25780	3256	25920	23	0,54%

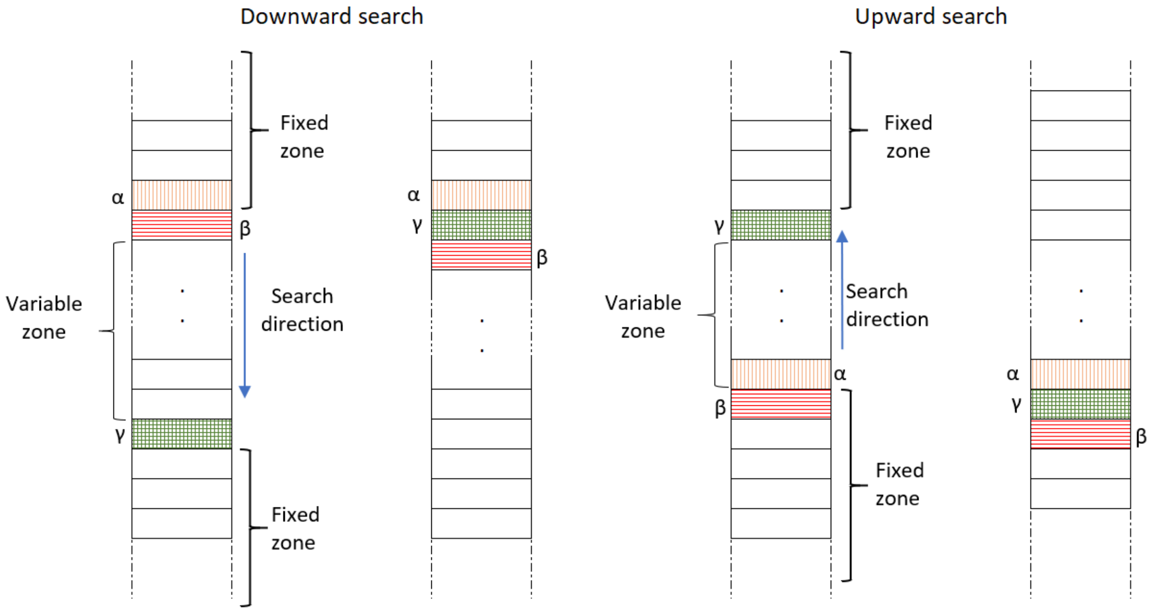

Figure 6. Family-based local search process

In downward search,

$c_j + p_\gamma < due_j \qquad \forall j \in VZ$

condition is sought. Here, VZ refers to the job set between the order in which the job γ sought and the order in which the job β is located. As a result of the downward search, after the location of the job γ changes, the completion time of the jobs in the variable zone is updated to increase by pγ. There is no change in the completion time of jobs in the fixed zone. If job γ cannot be found within these limits, no change will be made. In other words, in this iteration, the boiler washing is inevitable for the related job (β) to start. In upward search, job γ (job to fall between α and β and eliminate boiler washing) searched for moving is searched towards the top of the list by job β. This time, jobs in the variable zone are completed earlier than before the change. In other words, this change does not have a negative effect in terms of tardiness. In the upward search, a search is made according to the tardiness varying depending on the updated completion time of job γ. For this, the γ job to be relocated must fulfill the following condition:

$$\sum_j p_j + p_\gamma < due_\gamma$$

3. RESULTS AND DISCUSSION

Random problem samples were created to test the performance of the proposed approach. The created test problems are divided into three groups according to the number of jobs: large-sized samples (($n_b \in \{60, 80, 100\}$)), medium-sized samples ($n_m \in \{10, 15, 20\}$), and small-sized samples ($n_s \in \{6, 7, 8, 9, 10\}$). Small-sized examples are used to compare the mathematical model with the proposed heuristic. The preparation time is fixed for all test problems, with 2 hours. The number of families (5) and setup time (2 hours) are fixed for all test problems. Family types are distributed uniformly between [1, 5]. Processing times (pj) vary according to family type ($p_j \in \{5, 8, 10, 12, 14\}$). The due dates ($due_j$) of the jobs have been created according to the approach developed by [37]. Accordingly, due dates are integers distributed uniformly in the range of [0, r* $\sum_j p_j$]. Here r factor is used to control the due date range ($r \in \{0.5, 0.6, 0.7, 0.8\}$). A low r value creates small due dates, a large r value creates large due dates.

First, the proposed iterative greedy heuristic based approach is compared with the integer programming model. For comparison, a total of 24 different random test problems were created, one for all ns (small size) and r values. Results are given in Table 5.

The second column in Table 5 shows the total tardiness. The third column shows the total number of washes, the fourth column shows the gap value. The integer model was terminated after 60 seconds and the achieved result was shared. As can be seen, the optimum result was achieved with the integer model for only 6 and 7 job samples within 60 seconds. It has been observed that the proposed heuristic reaches the optimum for 6 and 7 job samples in under 1 second. Larger samples are integer model, results with high gap value have been achieved. However, the proposed heuristic still produced quality results in less than 1 second. In larger samples, the integer model could reach feasible solutions with high gap value, while the proposed heuristic again produced quality results in less than 1 second.

For medium and large sized problems (a total of 24 problems), since solutions cannot be obtained with the integer model, the results obtained only with the suggested heuristic are shared in Table 6.

Table 5. Comparative results

	Integer Linear Programming Model				IGH Based Algorithm		
(n, r)	**Tardiness**	**Num. of Washing**	**GAP (%)**	**CPU Time**	**Tardiness**	**Num. of Washing**	**CPU Time**
(6, 0.5)	144	0	0	<1	144	0	<1
(6, 0.6)	80	0	0	<1	80	0	<1
(6, 0.7)	37	0	0	<1	37	0	<1
(6, 0.8)	55	0	0	<1	55	0	<1
(7, 0.5)	138	1	0	24	140	1	1
(7, 0.6)	65	0	0	10	65	0	<1
(7, 0.7)	99	0	0	19	99	0	<1
(7, 0.8)	102	0	0	18	102	0	<1
(8, 0.5)	250	0	88	60	250	0	<1
(8, 0.6)	162	0	94	60	162	0	<1
(8, 0.7)	75	0	96	60	75	0	<1
(8, 0.8)	159	1	79	60	161	1	<1
(9, 0.5)	189	0	100	60	189	0	1
(9, 0.6)	226	0	94	60	233	1	1
(9, 0.7)	226	0	93	60	226	0	1
(9, 0.8)	119	1	79	60	121	1	1
(10, 0.5)	287	0	93	60	287	0	1
(10, 0.6)	229	0	94	60	226	1	1
(10, 0.7)	63	1	98	60	65	0	1
(10, 0.8)	104	1	89	60	107	0	1

Table 6. Test results for medium and large sized problems

(n, r)	Best	Worst	Avg.	RPD	Avg. CPU
(10, 0.5)	335	335	335	0,00%	1
(10, 0.6)	217	222	218	0,40%	1
(10, 0.7)	149	160	152	1,90%	1
(10, 0.8)	287	308	289	0,60%	1
(15, 0.5)	715	745	719	0,60%	1
(15, 0.6)	346	358	349	0,80%	1,1
(15, 0.7)	462	483	466	0,80%	4,5
(15, 0.8)	252	271	254	0,70%	3,8
(20, 0.5)	869	869	869	0,00%	4
(20, 0.6)	825	846	830	0,60%	4
(20, 0.7)	378	381	379	0,10%	4
(20, 0.8)	284	292	286	0,60%	4
(60, 0.5)	6815	7394	7150	4,60%	7
(60, 0.6)	6697	7283	6990	4,20%	6
(60, 0.7)	5462	6312	5779	5,50%	6
(60, 0.8)	4077	4080	7079	0,70%	6
(80, 0.5)	13492	14518	13904	2,90%	9
(80, 0.6)	14483	14935	14418	2,70%	9
(80, 0.7)	10397	11685	10979	3,60%	9
(80, 0.8)	7232	7578	7525	3,90%	9
(100, 0.5)	22836	24261	23554	3,00%	18
(100, 0.6)	18794	19594	19490	3,50%	18
(100, 0.7)	17150	18306	17774	3,50%	18
(100, 0.8)	11353	11353	11353	0,00%	18

In the first column, the size of the problem (number of jobs) and the coefficient r are given. The best, worst and average results of heuristic after 30 runs are presented in order. In the fifth column, the relative percentage deviation (RPD) value is shared. In studies conducted in the literature, this value generally expresses the deviation from the best-known value. However, in this study, it expresses the deviation of the best solution (Sln_{best}) found after 30 repetitions from the mean (Sln_{avg}) and is calculated as follows:

$$RPD = \frac{Sln_{avg} - Sln_{best}}{Sln_{avg}}$$

The small RPD value indicates that the best value reached is not by chance and the algorithm produces consistent results. Solution time (seconds) is given in the last column. In addition, the convergence graphs in Figure 7 are shared to give an idea about the performance of the proposed heuristic.

The graphs reflect the total tardiness values reached by the proposed heuristic over 100 iterations for three different examples.

In addition, the heuristic developed was tested with 6 months of real data belonging to the company. The number of boiler washing has been reduced by 17% compared to the manual order. An improvement of 13% was achieved in the total tardiness, which is the other performance criterion that is tried to be improved.

4. CONCLUSION

In this article, a productivity study was carried out for the bobbin dyeing process, which is an important component of textile manufacturing industry. The bobbin dyeing process starts with the batching of yarns belonging to different orders. Then these batches are dyed in a certain sequence. The batching problem has been excluded from this study, focusing on the sequencing problem. Additionally, in this study similar papers in the textile finishing industry were reviewed and presented by classifying them according to various criteria. To the best of our knowledge, there is only one study [18] on yarn dyeing in the context of the subject discussed. While the related study is parallel machine scheduling problem, this work is single machine scheduling problem. In addition, while the related study [18] is in the single-objective optimization class, this study is in the multi-objective optimization class. In the first objective, it is aimed to sequence the batches in accordance with their due dates, that is, to minimize the total tardiness. In the other, it is aimed to minimize the number of boiler washes by sequencing the batches of different colors in an ideal way. To solve the problem, an integer linear programming model and a heuristic technique based on iterative greedy search are proposed. Both approaches have been tested with randomly generated problems. Due to the NP-Hard nature of the problem, the integer model yielded results only for small size problems. It has been observed that the proposed heuristic reaches the optimum result in a short time for small sized problems. Heuristics has also been tested for medium and large sized problems. The results obtained show that the proposed heuristic achieves very high-quality results in an acceptable time. In addition, the proposed algorithm has been tested with real data of Bursalı Tekstil bobbin dye shop. For this, 6 months of retrospective data were used. The results produced by the heuristic were compared with the manual sequencing. There was 17% improvement in the number of boiler washes and 13% improvement in total tardiness. The developed heuristic has many applications from the rubber extractor to the steel manufacturing sector. In the future, it is aimed to use the solution method developed widely by designing a user-friendly interface.

Acknowledgments

I would like to thank all Bursalı Group family in the presence of Bursalı Group IT Manager Ayhan Mutlu and Engineering Manager Ahmet Bedir for their valuable contribution to this study.

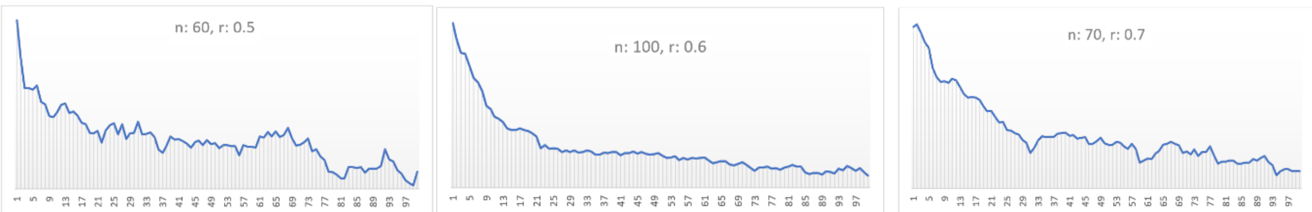

Figure 7. Convergence graphs

REFERENCES

1. Altıntaş M. 2010. Investigation of the effect of pattern structure on bobbin dyeing (in Turkish) (Unpublished master's thesis). Pamukkale University, Institute of Science, Denizli.
2. Finishing Procces of Textile Products, Resource Efficiency Guide (in Turkish) 2018. No: e-7 Resource Efficiency Guide in industry-1, December 2018, ISBN: 978-605-4889-34-1.
3. (https://www.temizuretimmerkezi.org/source/Tekstil%20U%CC%88ru%CC%88nlerinin%20Rehberi%20Tasar%C4%B1m_ekitap.pdf)
4. Özkan N. 2016. Assesment of chemical exposure to skin and respiration at dyeing and finishing processes of textile products (in Turkish). Occupational Health and Safety Specialization Thesis.

5. Tomruk E. 2008. Analysing winding structure of prepared bobbins for dyeing (in Turkish) (Unpublished master's thesis). Pamukkale University, Institute of Science, Denizli.
6. Öztürk E. 2014. Applications of integrated pollution prevention and control and clean production in textile industry (in Turkish) (Unpublished doctoral dissertation). Süleyman Demirel University, Institute of Science, Isparta.
7. Özdemir AO. 2009. Investigation of cationization conditions to improve the productivity of colouration of cellulosic textile materials (in Turkish) (Unpbulished graduation thesis). Erciyes University, Institute of Science, Kayseri.
8. Ayhan F, Akpolat O. 2021. Experimental design optimization and decolorization of an azo dye by cross-linked peroxidase aggregates. *Textile and Apparel* 31(1), 34-42.
9. Boominathan S, Balakrishanan S. 2020. Optimization of process parameters on color strength and antimicrobial activities of cotton fabric dyed with Rubia cordifolia extract. *Journal of Natural Fibers* 1-15.
10. Chitra V, Ramachandran S, Anandaraj V. 2021. Determination of specific wear rate of steel coated with tungsten carbide using Taguchi technique for textile applications. Materials Today: Proceedings.
11. Duru SC, Göcek İ. 2020. Assessing water-related comfort performance of knitted fabrics made of rayon microfibers and lyocell fibers for ıntimate wear. *Textile and Apparel* 30(3), 220-230.
12. Özbek A, Yildiz A. 2020. Digital supplier selection for a garment business using interval type-2 fuzzy topsis. *Textile and Apparel* 30(1), 61-72.
13. Karacizmeli IH, Ogulata, SN. 2017. Energy consumption management in textile finishing plants: A cost effective and sequence dependent scheduling model. *Textile and Apparel* 27(2), 145-152.
14. Yaghin RG, Sarlak P. 2021. Textile supply chain management considering carbon emissions and apparel demand fuzziness: a fuzzy mathematical programming approach. *International Journal of Clothing Science and Technology* 34(2), 137-155.
15. Zhang R, Chang PC, Song S, Wu C. 2017. A multi-objective artificial bee colony algorithm for parallel batch-processing machine scheduling in fabric dyeing processes. *Knowledge-Based Systems* 116, 114-129.
16. El Hachemi N, Saddoune M, El Hallaoui I, Rousseau LM. 2018. A two-phase approach to solve the synchronized bin–forklift scheduling problem. *Journal of Intelligent Manufacturing* 29(3), 651-657.
17. Laoboonlur P, Hodgson TJ, Thoney KA. 2006. Production scheduling in a knitted fabric dyeing and finishing process. *Journal of the Textile Institute* 97(5), 391-399.
18. Saydam C, Cooper WD. 2002. A decision support system for scheduling jobs on multi‐port dyeing machines. *International Journal of Operations & Production Management* 22(9), 1054-1065.
19. Hsu HM, Hsiung Y, Chen YZ, Wu MC. 2009. A GA methodology for the scheduling of yarn-dyed textile production. *Expert Systems with Applications* 36(10), 12095-12103.
20. Chen WJ. 2009. Scheduling with dependent setups and maintenance in a textile company. *Computers & Industrial Engineering* 57(3), 867-873.
21. Li D, Wang J, Qiang R, Chiong R. 2021. A hybrid differential evolution algorithm for parallel machine scheduling of lace dyeing considering colour families, sequence-dependent setup and machine eligibility. *International Journal of Production Research* 59(9), 2722-2738.
22. Huynh NT, Chien CF. 2018. A hybrid multi-subpopulation genetic algorithm for textile batch dyeing scheduling and an empirical study. *Computers & Industrial Engineering* 125, 615-627.
23. Zhou L, Xu K, Cheng X, Xu Y, Jia Q. 2017. Study on optimizing production scheduling for water-saving in textile dyeing industry. *Journal of Cleaner Production* 141, 721-727.
24. Jiang W, Yuan Z, Bi J, Sun, L. 2010. Conserving water by optimizing production schedules in the dyeing industry. *Journal of Cleaner Production* 18(16-17), 1696-1702.
25. Allahverdi A, Soroush HM. 2008. The significance of reducing setup times/setup costs. *European Journal of Operational Research* 187(3), 978-984.
26. Jacob V, Arroyo JEC. 2016. ILS Heuristics for the single-machine scheduling problem with sequence-dependent family setup times to minimize total tardiness. *Journal of Applied Mathematics* 2016.7.
27. Abdallah KS, Jang J. 2019. Family splitting algorithm for a single machine total tardiness scheduling problem with job family setup times. *International Journal of Industrial Engineering* 26(4), 452-470.
28. Potts CN, Van Wassenhove LN. 1992. Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. *Journal of the Operational Research Society* 43(5), 395-406.
29. Mecler D, Abu-Marrul V, Martinelli R, Hoff A. 2022. Iterated greedy algorithms for a complex parallel machine scheduling problem. *European Journal of Operational Research* 300(2), 545-560.
30. Schaller JE. 2014. Minimizing total tardiness for scheduling identical parallel machines with family setups. *Computers & Industrial Engineering* 72, 274-281.
31. Ebrahimi M, Ghomi SF, Karimi B. 2014. Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. *Applied Mathematical Modelling* 38(9-10), 2490-2504.
32. Deliktas D, Torkul O, Ustun O. 2019. A flexible job shop cell scheduling with sequence-dependent family setup times and intercellular transportation times using conic scalarization method. *International Transactions in Operational Research* 26(6), 2410-2431.
33. Pinheiro JC, Arroyo JEC. 2020. Effective IG heuristics for a single-machine scheduling problem with family setups and resource constraints. *Annals of Mathematics and Artificial Intelligence* 88(1), 169-185.
34. Herr O, Goel A. 2016. Minimising total tardiness for a single machine scheduling problem with family setups and resource constraints. *European Journal of Operational Research* 248(1), 123-135.
35. Sels V, Vanhoucke M. 2012. A hybrid genetic algorithm for the single machine maximum lateness problem with release times and family setups. *Computers & Operations Research* 39(10), 2346-2358.
36. Jin F, Gupta JN, Song S, Wu C. 2010. Single machine scheduling with sequence-dependent family setups to minimize maximum lateness. *Journal of the Operational Research Society* 61(7), 1181-1189.
37. Jin F, Song S, Wu C. 2009. A simulated annealing algorithm for single machine scheduling problems with family setups. *Computers & Operations Research* 36(7), 2133-2138.
38. Feng J, Shiji S, Cheng W. 2009. Dominance property based tabu search for single machine scheduling problems with family setups. *Journal of Systems Engineering and Electronics* 20(6), 1233-1238.
39. Gupta JN, Chantaravarapan S. 2008. Single machine group scheduling with family setups to minimize total tardiness. *International Journal of Production Research* 46(6), 1707-1722.
40. Bigras LP, Gamache M, Savard G. 2008. The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times. *Discrete Optimization* 5(4), 685-699.
41. Schaller JE, Gupta JN. 2008. Single machine scheduling with family setups to minimize total earliness and tardiness. *European Journal of Operational Research* 187(3), 1050-1068.
42. Omar MK., Suppiah Y, Teo SC, Bennell JA. 2008. Scheduling incompatible job families on a single machine: A two-level heuristic approach. In 2008 IEEE International Conference on Industrial Engineering and Engineering Management (pp. 317-320).
43. Uzsoy R, Velásquez JD. 2008. Heuristics for minimizing maximum lateness on a single machine with family-dependent set-up times. *Computers & Operations Research* 35(6), 2018-2033.
44. Taner MR, Hodgson TJ, King RE, Schultz SR. 2007. Satisfying due-dates in the presence of sequence dependent family setups with a

special comedown structure. *Computers & Industrial Engineering* 52(1), 57-70.

45. Schaller J. 2007. Scheduling on a single machine with family setups to minimize total tardiness. *International Journal of Production Economics* 105(2), 329-344.

46. Eren T, Güner E. 2006. A bicriteria scheduling with sequence-dependent setup times. *Applied Mathematics and Computation* 179(1), 378-385.

47. Yuan JJ, Liu ZH, Ng CT, Cheng TE. 2006. Single machine batch scheduling problem with family setup times and release dates to minimize makespan. *Journal of Scheduling* 9(6), 499-513.

48. Kacem I. 2006. Lower bounds for tardiness minimization on a single machine with family setup times. In The Proceedings of the Multiconference on" Computational Engineering in Systems Applications" (Vol. 1, pp. 1034-1039). IEEE.

49. Karabati S, Akkan C. 2006. Minimizing sum of completion times on a single machine with sequence-dependent family setup times. *Journal of the Operational Research Society* 57(3), 271-280.

50. Suriyaarachchi RH, Wirth A. 2004. Earliness/tardiness scheduling with a common due date and family setups. *Computers & Industrial Engineering* 47(2-3), 275-288.

51. Schultz SR, Hodgson TJ, King RE, Taner MR. 2004. Minimizing L max for the single machine scheduling problem with family set-ups. *International Journal of Production Research* 42(20), 4315-4330.

52. Baker KR. 1999. Heuristic procedures for scheduling job families with setups and due dates. *Naval Research Logistics (NRL)* 46(8), 978-991.

53. Hariri AMA, Potts CN. 1997. Single machine scheduling with batch set-up times to minimize maximum lateness. *Annals of Operations Research* 70, 75-92.

54. Schutten JM, Van de Velde SL, Zijm WH. 1996. Single-machine scheduling with release dates, due dates and family setup times. *Management Science* 42(8), 1165-1174.

55. Crauwels HAJ, Potts CN, Van Wassenhove LN. 1996. Local search heuristics for single-machine scheduling with batching to minimize the number of late jobs. *European Journal of Operational Research* 90(2), 200-213.

56. Wagner HM. 1959. An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly* 6(2), 131-140.

57. Jacobs LW, Brusco MJ. 1995. A local search heuristic for large set-covering problems. *Naval Research Logistics Quarterly* 42(7), 1129–1140.

58. Moore JM. 1968. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science* 15(1), 102-109.