Journal of Industrial and Systems Engineering
Vol. 12, No. 1, pp. 198-214
Winter (January) 2019

Simultaneous production planning and scheduling in hybrid flow shop with time periods and work shifts

Iman Hassanzadeh Nodeh[1], Seyed Hessameddin Zegordi[1*]

[1]*Faculty of industrial and Systems Engineering, Tarbiat Modares University, Tehran, Iran*

im_hasanzade@yahoo.com, zegordi@modares.ac.ir

Abstract

Simultaneous production planning and scheduling has been identified as one of the most important factors that affect the efficient implementation of planning and scheduling operations for the production systems. In this paper, simultaneous production planning and scheduling is applied in a hybrid flow shop environment, which has numerous applications in real industrial settings. In this problem, it is assumed that each time period includes a number of discontinuous intervals called work shifts. A novel mixed integer linear programming model is formulated. Since this problem is NP-hard in the strong sense, a new heuristic algorithm is developed to construct a complete schedule from a solution matrix that is embedded in the proposed Tabu search. A number of test problems have been solved to compare the performance of the proposed method with the exact method. The results show that the proposed tabu search is an effective and efficient method for simultaneous production planning and scheduling in hybrid flow shop systems.

Keywords: Simultaneous production Planning and Scheduling, hybrid flow shop, mixed integer linear programming, Tabu search, work shifts

1- Introduction

Production planning and scheduling belong to different decision-making levels. However, they are closely related. Scheduling is the phase of production management that creates a detailed description of operations to be executed in a given period of time, typically short. Production planning, compared to scheduling, is characterized by a higher level of abstraction and a longer time interval. Production planning is created for a planning horizon which is divided into several planning periods. Both Planning horizon and planning periods are periodic and separated.

In recent years, particular attention is paid to integrated production planning and scheduling. Three different approaches are generally proposed for integration of production planning and scheduling: monolithic approach, hierarchical approach and iterative approach. In the monolithic approach, production planning and scheduling are solved simultaneously. This approach is typically based on mathematical models which are hard to solve because of high complexity. This is because detailed scheduling constraints are incorporated into the model in each planning period. The approach is also called simultaneous production planning and scheduling. In hierarchical and iterative strategies, the problem is decomposed into a higher-level problem that is used to determine production targets, and one or more lower level problems with detailed scheduling.

*Corresponding author
ISSN: 1735-8272,

The solution of the planning problem is used as an input for the scheduling problem. If the flow of information is only from the higher level problem to the lower level problem(s), then the strategy is hierarchical. If there is a feedback loop from the scheduling lower-level problems to the higher-level problem, then the strategy is iterative.

In most academic works in the area of integrated production planning and scheduling, each planning period is assumed to be continuous. The duration of a planning period is usually a week, a month or several months, which is mostly discontinuous in discrete manufacturing and process industries. That's because manufacturing is not possible during this period of time continuously due to the limitations of human and machines. It is necessary to consider work shifts within a time period and the related constraints. Aggregation of work shifts within a period is not generally useful to integrate production planning and scheduling. This is because in some process industries, processing times of operations are notable compared with the duration of shifts (e.g. chemical, pharmaceutical, food and oil industry). In these industries, an operation must be accomplished on the shift that it is started. It may be a considerable useless time for some machines at the end of work shift, because processing a new operation can't be finished on the same shift. Assignment of work shifts is considered by researchers in areas such as nurse scheduling, crew scheduling, surgery scheduling, and clinic scheduling, but is also essential for industrial environments. Labor forces in production industries mostly work in shifts and since they work with machines, machine scheduling must be done based on the limitations of work shifts.

Among the production scheduling systems, hybrid flow shop is one of the most particular environments. It is a generalization of flow shop where, at any stage, there may exist more than one machine. It is also called flexible flow shop and flow shop with multiple processors. The problem consists of assigning the jobs to machines at each stage and sequencing the jobs assigned to the same machine so that some criteria are optimized. Hybrid flow shop models differ in the type and number of parallel machines at the stages. The parallel machines may be identical, uniform or unrelated. The HFS has numerous applications in various industrials such as chemical, pharmaceutical, oil, food, tobacco, textile, paper and metallurgical industry (Zandieh et al, 2006). Furthermore, industrial settings like semiconductors, electronics manufacturing and petrochemical production can be modeled as a hybrid flow shop (Low et al, 2008). There is a vast literature on hybrid flow shop scheduling problem, but like other traditional scheduling problems, the most studied problem is minimizing makespan or other performance measures (e.g. earliness/tardiness, lateness, etc). It is assumed that the end of schedule time is undetermined and the objective is to sequence and schedule a set of jobs in a way that they are completed as soon as possible or due dates are satisfied. In most real cases, time is limited and the problem is what and when to manufacture in order to maximize productivity or to minimize costs.

In this study, we try to provide a mathematical model for Simultaneous production planning and scheduling in hybrid flow shop environment by considering work shifts. The objective of the model is to maximize profit. After presenting the mathematical model of the problem, a Tabu search is proposed and a constructive procedure is developed to create a complete schedule.

The paper continues as follows: In section 2, the literature review and the research background of studies in this field is presented. Problem definition and the expression of hypotheses are given in section 3. A novel Simultaneous planning and scheduling model is presented in Section 4. A Tabu search and a constructive algorithm are developed in section 5. Computational experimental results are shown in Section 6, and finally in section 7, conclusions of the study are presented and future research directions are recommended.

2-Literature

The hybrid flow shops scheduling problem is a complex combinatorial optimization problem observed in many real-world applications. The HFS is np-hard in most cases. Gupta (1988) showed that scheduling two-stage hybrid flow shop with more than one machine at one stage is NP-hard when the objective is to minimize the makespan. Similarly, Hoogeveen et al. (1996) proved that preemptive scheduling for the investigated problem by Gupta is strongly NP-hard. Therefore, our problem, since it contains the above problem with the integration of production planning and scheduling and consideration of work shifts, is also NP-hard.

Most of assumptions and special cases considered in hybrid flow shop scheduling in the literature have been confined to sequence dependent setup times (Ebrahimi et al, 2014; Pan et al, 2017), non-identical parallel machines (Attar et al, 2013; Rajaee Abyaneh & Gholami, 2015), limited buffer capacities (Li & Pan, 2015; Safari et al, 2017), machine eligibility constraints (Soltani & Karimi, 2015; Yu et al, 2018; Zhang & Chen, 2018), and more than one criterion (Hosseini, 2017; Li et al, 2018; Mousavi et al, 2018).

Although many researchers have addressed hybrid flow shop scheduling, few of them have studied integration of production planning and scheduling in hybrid flow shop environments. Riane et al (2001) developed a decision support system for hybrid flow shop organization. Their proposed DSS was composed of decomposition into planning and scheduling and closed loop or feedback mechanism. Ramezanian et al (2016) studied a simultaneous lot-sizing and scheduling for a hybrid flow shop production system with capacity constraint. They proposed a mixed integer programming model for the problem. They also applied a rolling horizon heuristic and particle swarm optimization algorithm to solve the problem. Cho and Jeong (2017) applied a two-level hierarchical method for production planning and scheduling of reentrant hybrid flow shop problems. Their purpose was to improve productivity and customer satisfaction. They considered an upper-level planning problem and a lower-level scheduling problem, and applied their proposed method to a real industry problem.

Furthermore, few researchers have addressed hybrid flow shop scheduling problem with work shift constraint. Nejati et al (2016) studied multi-job lot streaming problem in a hybrid flow shop environment. They considered work-in-process jobs, work shift constraint and sequence-dependent setup times with the objective of minimizing the sum of weighted completion times of jobs. They presented a mixed integer nonlinear formulation for the problem that was capable of solving small-size problems and proposed a genetic algorithm and simulated annealing for large-sized problems.

Nowadays, many companies are concerned with achieving higher profit or imposing lower cost. Therefore, scheduling problems with productivity related measures have more practical meaning than before. Nevertheless, there is far less literature on hybrid flow shops under maximization of productivity-based objectives than the other objectives. Productivity is about producing more output with the same available time and resources. With higher labor productivity, a company can produce more goods and services with the same amount of relative work. There is increasing attention in the last decade to integrated production planning and scheduling with the aim of optimizing productivity. Erdirik-Dogan and Grossmann (2008) investigated simultaneous planning and scheduling of single-stage multi-product continuous production with parallel machines. They presented a mixed integer linear programming model with multiple periods, and developed a bi-level decomposition technique and decomposed the problem into an upper-level planning and a lower-level scheduling problem. Their method was an extension of the work by Erdirik-Dogan and Grossmann (2006) to the case of parallel units. Terrazas-Moreno and Grossmann (2011) addressed problem of simultaneous planning and scheduling in a production–distribution network of continuous production. The production–distribution network includes a set of production sites that distribute products to different markets and involves different temporal and spatial scales. They extended MILP model by Erdirik-Dogan and Grossmann (2008) to multi-site, multi-market networks and presented a mathematical model for integrated planning and scheduling. They proposed two decomposition methods to solve the problem, based on bi-level decomposition including upper-level planning problem and lower-level scheduling problem. They presented four case studies to compare the performance of proposed decomposition methods. Kim and Lee (2016) proposed an iterative procedure using an optimization and simulation model for Synchronized production planning and scheduling in semiconductor fabrication. They tried to coordinate the input and output quantity of the production plan.

Aguirre and Papageorgiou (2018) addressed integration of production planning, scheduling and maintenance in a multi-product single-stage environment with parallel machines, limited resources and sequence-dependent setups. They also considered flexible recovery operations, resource availability and product lifetime. They presented a novel MILP based on the main ideas of travelling salesman problem and precedence-based constraints. Jankauskas et al (2019) studied capacity planning and scheduling of biopharmaceutical manufacture and proposed fast genetic algorithm approaches for solving both medium- and long-term capacity planning and scheduling of single-site and multi-site biopharmaceutical manufacture.

The contributions of this study can be stated as follows:

1. Most of the previous studies especially on the sequencing and scheduling, consider an undetermined length of time, and the objective is to optimize a traditional scheduling performance measure or a combination of some of them. However, in real-world problems, the durations of production are limited and specified during a certain amount of time, and companies generally try to optimize a productivity-based performance measure. For the problem of this study, it is assumed that there is determined available time for production, and a productivity-based performance measure (profit) is optimized.
2. Most of the previous studies that addressed integrated production planning and scheduling, or just scheduling, consider continuous time periods, but in most real cases only work shifts are continuous and a time period consists a number of work shifts that don't come immediately after each other. The mathematical model of this study considers separate work shifts within a time period. This is the first study that considers hybrid flow shop environment, and existence of the planning horizon, planning periods and work shifts in integrated production planning and scheduling.
3. Most of the academic studies consider unlimited flow time to manufacture a products, but in some industries like chemicals chemical, pharmaceutical, and food industry, there is limited flow time for some process stages or all of them. In this work, limited flow time is considered for every batch.

3- Problem description

Planning and scheduling integrated model can be considered for small size problems, but it is intractable for medium and large size problems. The problem is then to determine the jobs to be processed in each period and each work shift, sequencing and scheduling of these jobs, and the inventory levels for each job. The objective is to maximize the total profit, in terms of sales revenues, lost sales costs and inventory costs. Other operating costs are supposed to be fixed in each period and so are not considered in the objective function. Problem assumptions are as follow:

- The model parameters, such as due dates, prices, lost sales costs, processing times and inventory costs per period are assumed to be deterministic.
- Production planning horizon is divided into several planning periods. Each planning period includes a number of day shifts with equal time length. The numbers of day shifts that belong to different periods are not necessarily equal.
- Every job must be processed on a processing unit without interruption until it is processed completely.
- Each operation must be done on a processing unit within a day shift. It can't be interrupted at the end of a day shift and be resumed in the next day shift.
- There is an unlimited buffer capacity between two successive processing stages.
- Transfer times are assumed to be neglected.
- The setup times between the operations are sequence independent and thus included in the processing times.
- Machines are available at all times, with no breakdowns or scheduled/unscheduled maintenance.
- All jobs are available at zero time. Job processing cannot be interrupted (no preemption is allowed) and jobs have no associated priority values.
- Machines in parallel are identical in capability and processing rate.
- If a job is not completed at the time of its due date, lost sale cost is imposed on the manufacturer and the shortage is not compensable in future periods.

4- Simultaneous planning and scheduling model

The simultaneous production planning and scheduling problem is formulated as a mixed integer linear programming model. The proposed MILP model is as follow:

Indices

i,j	jobs
t	time periods
d	day shifts
l	manufacturing stages
m,n	machines

Parameters

PT_{il}	processing time of batch i at stage l
m_l	number of machines at stage l
d_t	number of day shifts in time period t
prc_i	sale price of job i
cs_inv_i	inventory cost of job i for every time period
cs_los_i	lost sale cost of job i
jb_due_{it}	binary variable to denote if due date of job i is at the end of time period t
bgn_shf_{td}	beginning time of day shift d in time period t
end_shf_{td}	ending time of day shift d in time period t
Y_{ilm}	binary variable to denote if job i can be processed on machine m of stage l
A	a large number

Variables

X_{ilmtd}	binary variable to denote if job i is processed on machine m at stage l in day shift d of time period t
W_{ijlmtd}	binary variable to denote if job j is processed after job i on machine m at stage l in day shift d of time period t
Str_{ilmtd}	start time of processing job i on machine m at stage l in day shift d of time period t
Fin_{ilmtd}	finish time of processing job i on machine m at stage l in day shift d of time period t
jb_pl_i	binary variable to denote if processing of job i is completed in the planning horizon
jb_cmp_{it}	binary variable to denote if processing of job i is completed in time period t
jb_str_{it}	binary variable to denote if processing of job i is started in time period t
Inv_{it}	binary variable to denote if job i is at inventory at the end of time period t
Los_{it}	binary variable to denote if the sale of job i is lost at the end of time period t

Mathematical model

$$max \quad Z = prc_i * jb\_pl_i - \sum_k \sum_t (cs\_inv_i \times Inv_{it}) - \sum_k \sum_t (cs\_los_i \times Los_{it}) \tag{1}$$

$$\sum_t jb\_cmp_{it} = jb\_pl_i \qquad \forall\, i \tag{2}$$

$\sum_t jb\_str_{it} = \ jb\_pl_i \quad \forall\, i \quad (3)$

$\sum_t \sum_d \sum_m X_{ilmtd} = \sum_t \sum_d \sum_m X_{iSmtd} \quad \forall\, i, l \in [1, S-1], t \quad (4)$

$jb\_cmp_{it} \ = \sum_d \sum_m X_{iSmtd} \quad \forall\, i, t \quad (5)$

$jb\_str_{it} = \sum_d \sum_m X_{i1mtd} \quad \forall\, i, t \quad (6)$

$Str_{ilmtd} \geq bgn\_shf_{td} - \mathrm{A}(1 - X_{ilmtd}) \quad \forall\, i, l, m \in [1, m_l], t, d \in [1, d_t] \quad (7)$

$Fin_{ilmtd} \leq \ end\_shf_{td} + \mathrm{A}(1 - X_{ilmtd}) \quad \forall\, i, l, m \in [1, m_l], t, d \in [1, d_t] \quad (8)$

$Str_{ilmtd} = \ Fin_{ilmtd} - (PT_{il} * X_{ilmtd}) \quad \forall\, i, l, m \in [1, m_l], t, d \in [1, d_t] \quad (9)$

$X_{ilmtd} \leq Y_{ilm} \quad \forall\, i, l, m \in [1, m_l], t, d \in [1, d_t] \quad (10)$

$X_{ilmtd} + X_{i,l-1,n,t,f} \leq 1 \quad \forall\, i, l, m \in [1, m_l], n \in [1, m_{l-1}], t, (d, f) \in [1, d_t], f > d \quad (11)$

$X_{ilmtd} + X_{i,l-1,n,u,f} \leq 1 \quad \forall\, i, l, m \in [1, m_l], n \in [1, m_{l-1}], u > t, d \in [1, d_t], f \in [1, d_u] \quad (12)$

$Fin_{jlmtd} - Fin_{ilmtd} + \mathrm{A}\big(1 - W_{ijlmtd}\big) \geq PT_{il} \quad \forall\, i, j (i \neq j), l, m \in [1, m_l], t, d \in [1, d_t] \quad (13)$

$\sum_t \sum_d \sum_m Fin_{ismtd} - \sum_t \sum_d \sum_m Str_{i1mtd} \ \leq \max\_flow_i \quad \forall\, i \quad (14)$

$Fin_{i1mtd} \geq \ PT_{il} - \ \mathrm{A}(1 - X_{i1mtd}) \quad \forall\, i, m \in [1, m_{1l}], t, d \in [1, d_t] \quad (15)$

$Fin_{ilmtd} - Fin_{i,l-1,n,t,d} + \mathrm{A}\big(2 - X_{ilmtd} - X_{i,l-1,n,t,d}\big) \geq PT_{il} \quad \forall\, i, l > 1, m \in [1, m_l], n \in [1, m_{l-1}], t, d \in [1, d_t] \quad (16)$

$2\big(W_{ijlmtd} + W_{jilmtd}\big) - \big(X_{ilmtd} + X_{jlmtd}\big) \leq 0 \quad \forall\, i, j (i \neq j), l, m \in [1, m_l], t, d \in [1, d_t] \quad (17)$

$W_{ijlmtd} + W_{jilmtd} + \ \mathrm{A}\big(2 - X_{ilmtd} - X_{jlmtd}\big) \geq 1 \quad \forall\, i, j (i \neq j), l, m \in [1, m_l], t, d \in [1, d_t] \quad (18)$

$Inv_{it} = Inv_{i,t-1} \ + jb\_cmp_{it} - jb\_due_{it} + Los_{it} \quad \forall\, i, t \quad (19)$

$jb\_pl_i, jb\_str_{it}, jb\_cmp_{it}, Inv_{it}, Los_{it}, \ X_{ilmtd}, \ W_{ijlmtd} \in \{0,1\} \quad \forall\, i, j (i \neq j), k, l, m \in [1, m_l], t, d \in [1, d_t] \quad (20)$

$Str_{ilmtd}, Fin_{ilmtd} \ \geq 0 \quad \forall\, i, k, r, l, m \in [1, m_l], t, d \in [1, d_t] \quad (21)$

The objective (1) is to maximize profit. The profit is equal to sum of sales revenues minus inventory costs and lost sale costs. Labor costs and overhead costs are considered fixed for every planning period and so are not considered in the objective function. Constraints (2) ensure that if a batch is manufactured during the planning horizon, its manufacturing is completed in only one of planning periods. Constraints (3) ensure that if a batch is manufactured during the planning horizon, its manufacturing is started in only one of planning periods. Constraints (4) indicate that if a batch is processed at the last stage, it is processed at each of previous stages. Constraints (5) ensure that if manufacturing a batch is finished in time period t, it is processed at the last stage by only one machine in one of day shifts of the planning period. Constraints (6) ensure that if manufacturing a batch is started in time period t, it is processed at the first stage by only one machine in one of day shifts of the planning period. Constraints (7) enforce start time of processing a batch at a stage in a day shift of a planning period not to be lower than beginning time of that day shift. Constraints (8) enforce finish time of processing a batch in a stage in a day shift of a planning period not to be greater than ending time of that day shift. Constraints (9) define the relationship between start times and finish times of processing a batch at each stage. Constraints (10) enforce a batch not to be processed on a machine that is not capable of processing the batch. Constraints (11) state that if considering two successive stages in two different day shifts of a period, a batch can't be processed at the latter stage in the earlier day shift of the period. Constraints (12) state that if considering two successive stages in two different periods, a batch can't or be processed at the latter stage in a day shift of the earlier period. Constraints (13) define the relationship between finish processing times of two batches that are processed on a machine in a day shift of a planning period. Constraints (14) enforce flow time of a batch not to be greater than its maximum allowable flow time. Constraints (15) state that departure time of a batch is greater than its processing time in the first stage. Constraints (16) define the relationship between departure times of a batch between two consecutive stages in a day shift of a planning period. The relationships are between two processing finish times, two skip times, or a processing finish time and a skip time. If a batch skips a stage, processing time of the batch for that stage is set to zero and batch departure time is equal to the departure time of the batch at the previous stage. Constraints (17-18) link assignment variables with sequencing variables for batches that are processed on a machine of a processing stage in a day shift of a planning period. Constraints (17) state that if at most one of jobs i and j are processed on machine m, then the sequencing variables of these jobs equal to zero. Constraints (18) state that for any pair of jobs that are processed on a machine in a day shift, one of the jobs is processed after another, and thus one of the two sequencing variables regarding to those jobs equals to one. Constraints (19) link inventory, production, consumption, and lost sale variables. Constraints (20) imply the binary property of some of decision variables. Constraints (21) imply non-negativity of the corresponding variables.

5-Proposed Tabu search

Tabu search first proposed by Glover (1989a, 1989b), is an iterative improvement approach designed for getting a local optimum solution for combinatorial optimization problems. For hybrid flowshop scheduling problems, TS has been proven to be one of the most effective local search techniques (Wardono and Fathi, 2004; Wang and Tang, 2009). Therefore, in this paper we take TS as the solution method.

A Tabu search begins with an initial solution and improves the solution through a series of iterations. In each iteration, neighborhoods of solutions are evaluated that are similar to the current solution. Each of these neighbors differs from the current solution by a move. It is possible to define different neighborhoods based on different definitions of a move. In each iteration of the Tabu search, the entire neighborhood, or part of the neighborhood (if the neighborhood is very large), is explored and the attractiveness of each move is evaluated by using an evaluation function.

5-1- Solution representation

It is very difficult to apply Tabu search in hybrid flow shop scheduling with several periods each including a number of shifts. This is because the denotation of complete schedules of this problem is complex, and becomes more complex under consideration periods and day shifts. To overcome this obstacle, a multi-permutation of jobs is used. Each permutation includes the jobs related to one period and represents their processing order in the first stage of that period instead of a complete schedule (Wardono and Fathi, 2004; Wang and Tang, 2009). So, number of permutations in a multi-permutation matrix and periods are equal. A constructive procedure is required to obtain a corresponding complete schedule of each permutation.

5-2- Procedure for constructing a complete schedule

T : Number of periods

i_t : Corresponding job to the position i in permutation of period t

n_t: Number of jobs to be manufactured in period t

m_l: Number of machines in stage l

d_t: Day shift d in period t

ds_t: Last day shift in period t

S: Number of stages

$bgn(d_t)$: Beginning time of day shift d in period t

$end(d_t)$: Ending time of day shift d in period t

et_l : Earliest available time of machine(s) in stage l

$ef_{i_t,l}$: Earliest finish time of processing batch i_t in stage l

$f_{i_t,m,l}$: Finish time of processing batch i_t on machine m in stage l

The procedure for constructing a complete schedule of a batch multi-permutation is essentially a method based on the following pseudo-code:

```
for all t ≤ T do
    for all i_t ≤ n_t do
        for all l ≤ S do
            d_t ← bgn(d_t) ≤ et_l < end(d_t)
            if d_t ≤ ds_t
                if ef_{i_t,l} ≤ end(d_t)
                    Select machine m in stage l so that f_{i_t,m,l} = ef_{i_t,l}
                    if n_m>1
                        Select the machine with the latest available time (m)
                    end
                    Assign batch i_t to machine m and schedule it in stage l
                    Update the available time of machine m
                else
                    d_t := d_t + 1
                end
            else
                Delete batch i_t from previous stages (1 up to l-1) and Update the available
                times of selected machine(s)
            end
        end
    end
end
```

Example: An illustration for the procedure

To illustrate the procedure described above, an example is presented. In this example, there are twelve batches and three stages, and the number of machines in each stage is 3, 2 and 2, respectively. There are two periods, each including 2 day shifts. The processing times and sequence dependant setup times of batches are given in Table 1and Table 2 respectively. The batches must be processed in the first and second stages in a day shift.

Solution matrix $P1 = \begin{bmatrix} 4 & 1 & 3 & 2 & 5 & 0 & 0 & 0 \\ 8 & 11 & 12 & 10 & 7 & 9 & 6 & 13 \end{bmatrix}$ indicates that jobs 4,1,3,2 and 5 must be manufactured until the end of time period 1 and the others have to be manufactured before the end of time period 2. It also indicates the order that the jobs are selected and assigned according to the procedure for constructing a complete schedule. A complete schedule is constructed using the procedure and is illustrated in Fig 1.

Also, another schedule corresponding to matrix $P2 = \begin{bmatrix} 4 & 5 & 1 & 3 & 2 & 0 & 0 & 0 \\ 10 & 11 & 6 & 12 & 8 & 7 & 9 & 13 \end{bmatrix}$ is shown in figure 2. As shown in figure 1, only 10 jobs can be finished during the planning horizon but it is possible to finish all of jobs in figure 2. Thus, the order in which jobs of each period are entered into schedule might change the objective function.

Table 1. Processing times of batches in each stage

	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5	Batch 6	Batch 7	Batch 8	Batch 9	Batch 10	Batch 11	Batch 12	Batch 13
Stage1	11	9	12	8	7	9	8	10	13	7	10	13	14
Stage2	5	6	9	10	8	6	7	11	8	6	9	9	11
Stage3	6	8	7	5	7	11	9	8	5	10	8	6	9

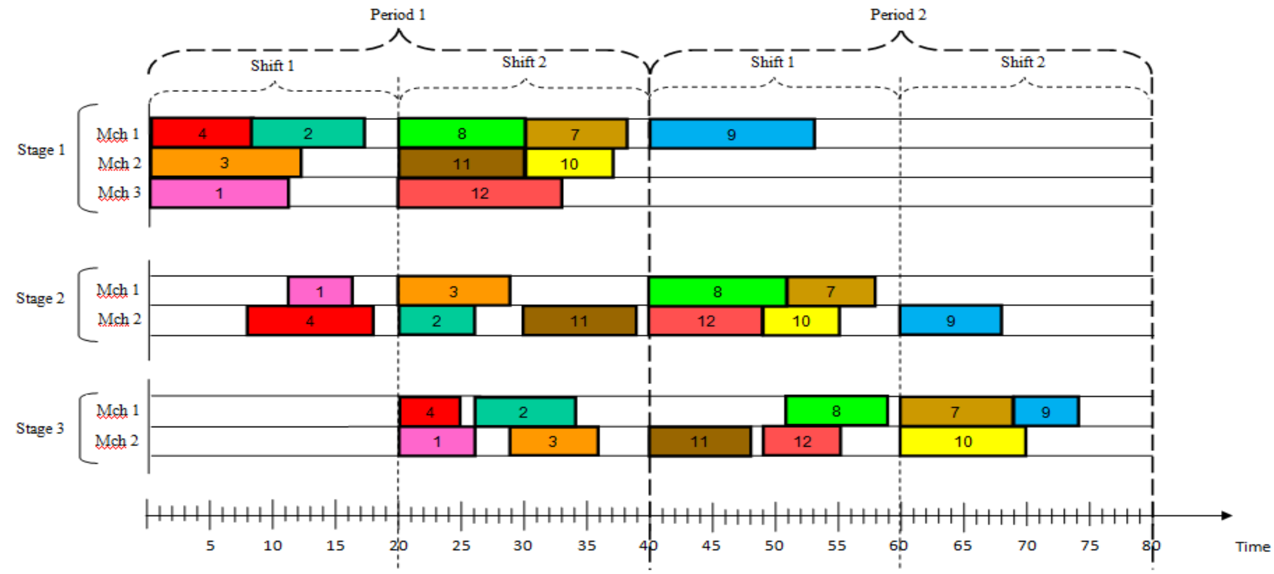

Fig 1. Gantt chart for the solution of P1

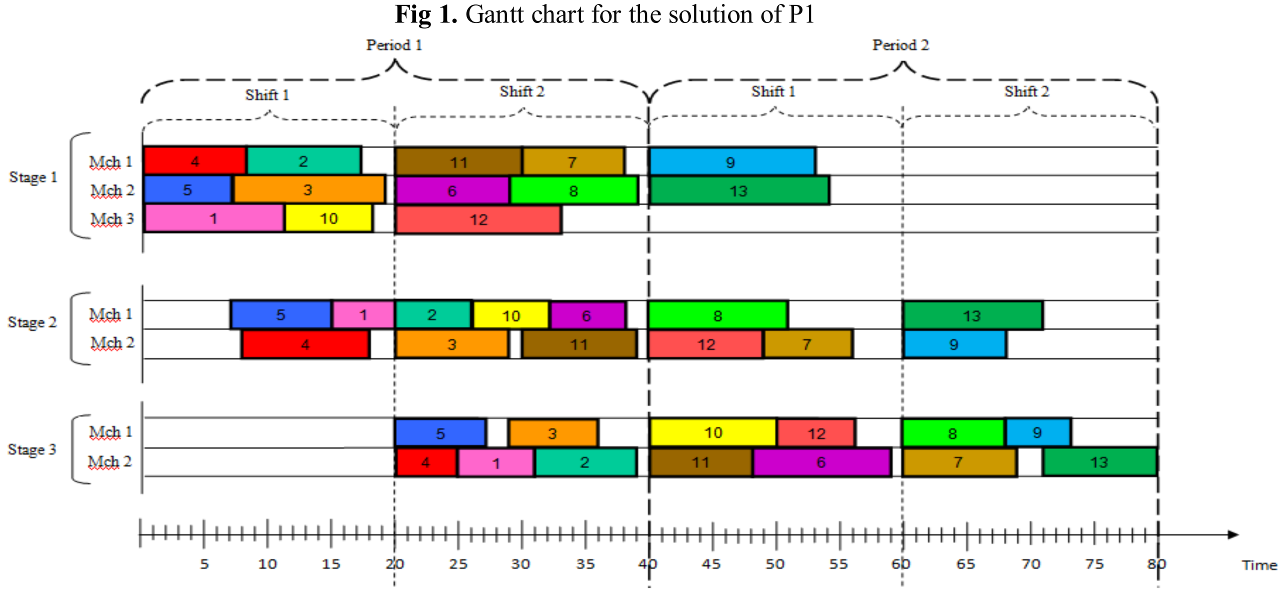

Fig 2. Gantt chart for the solution of P2

5-3- Initial solution

The initial solution space includes a multi-permutation matrix to start the Tabu search algorithm. It is generated randomly. Each permutation of matrix is related to one period of size n_t (quantity of batches to be manufactured in period t), so the permutations of different periods are not necessarily equal. The number of columns of the matrix is equal to the maximum number of batched belonging to a period. As shown in figure 3 this maximum is 9 and hence, there are 9 columns.

6	3	4	1	8	5	7	2	9
11	14	12	15	13	10	16	0	0
23	21	24	18	20	17	22	19	0

Fig 3. An initial solution randomly generated

The number of batches to be manufactured in period 1,2 and 3 are respectively 9, 7 and 8. Therefore, the last two entries in the second row and the last entry in the third row is represented by “0” that means there is no corresponding batch to manufacture.

5-4- Neighbourhood

Since the solution of the problem is represented by permutations of batches, it is easy to generate neighborhoods. A prevalent kind of neighborhood generation is adopted called insertion move. This kind of move has been proven very effective for scheduling problems. One period of a solution is selected randomly and the move is applied to a batch in the corresponding permutation of the period in the matrix (figure 4). The move is not applied to entries represented by “0”.

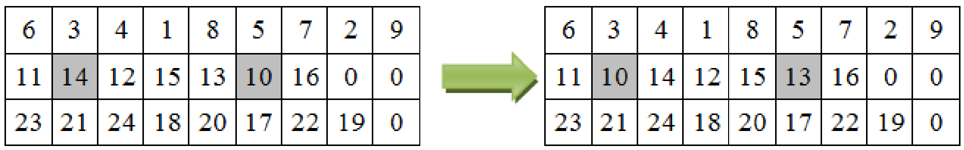

Fig 4. Insertion move (time period 2 is selected)

5-5- Search strategy

In order to reduce the time required for neighborhood generation and evaluation, a certain number of neighborhoods are generated in each iteration. Each neighborhood is generated by applying the move to permutation of a selected period (figure 5). If there are better solutions in the generated solutions, the best one is selected. If none of generated solutions is better than the current solution, then the entire of remaining neighborhood for that permutation is examined (figure 6). If a better solution is not obtained, the entire neighborhood is generated and evaluated for batches belonging to another permutation
(figure 7), and similarly if the search fails to find a better solution, the whole of remaining neighborhood is explored for a greater number of periods (figure 8).

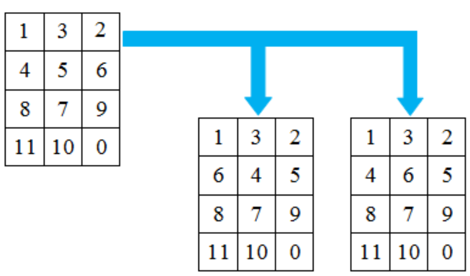

Fig 5. Time period 2 and batch 6 is selected

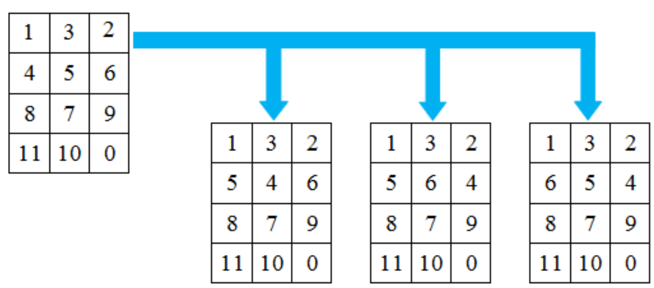

Fig 6. Time period 2 and the remaining neighborhood

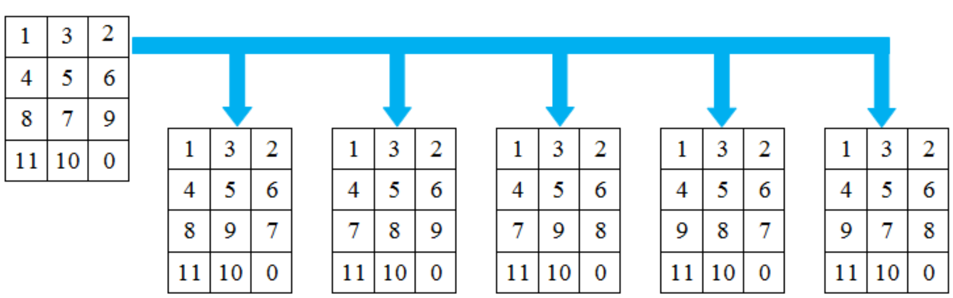

Fig 7. Time period 3 and the entire neighborhood

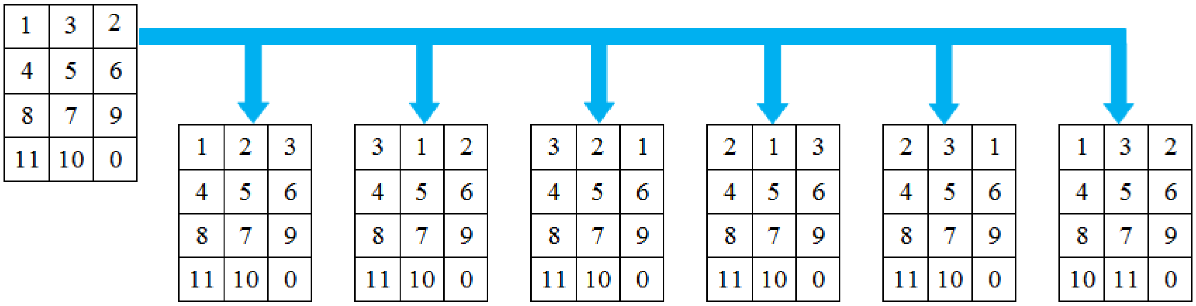

Fig 8. The whole of remaining neighborhood

5-6-Tabu list

A Tabu list is a short term memory that contains prohibited moves. Once a move is accepted, its reverse move is memorized in the Tabu list as a prohibited move. The number of iterations that a move remains in the Tabu list is called the Tabu list length. In the proposed search, Tabu list length is set to a fixed number. The list content is refreshed each time a new basic permutation is found; the oldest element is removed and the new one is added.

5-7- Diversification mechanism

After specified number of iterations, if there is no improvement in the objective function, the search is at a local optimum. Diversification avoids the search being trapped on a local optimum by guiding it in order to explore new regions. The neighborhood structure used by us to apply diversification is to select three randomly positions of a permutation (related to a period) and relocate corresponding batches into three other randomly selected positions (figure 9). A certain number of neighborhoods for that permutation are generated and the best solution is selected, even if it is not better than the current solution. This procedure is first applied to the first permutation (period) and after predefined number of diversifications without improvement in the objective function, it is applied to the next permutation and this procedure is repeated until the last permutation.

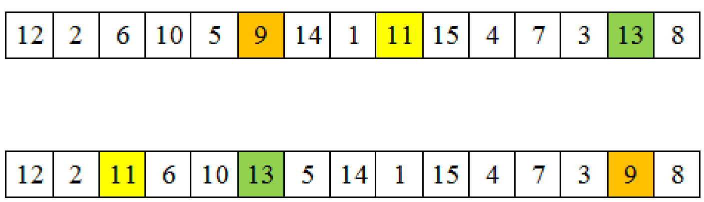

Fig 9. Diversification mechanism

5-8- Stopping criterion

The search process terminates when a predefined number of iterations are reached or the best solution found so far has not changed during a predefined number of diversifications for the last permutation (period).

6- Results

In this section, the results of computational experiments are conducted to evaluate the effectiveness of the proposed Tabu search. The full-space model was run in GAMS[1]. GAMS facilitates formulating, solving, and analyzing mathematical models, particularly optimization models. GAMS is tailored for complex, large scale modeling applications and is specifically designed for solving linear, nonlinear and

[1] General Algebraic Modelling System

mixed integer optimization models. The proposed Tabu search was coded in Matlab and implemented on a personal computer with intel core i7 2.30-GHz CPU and 8 GB RAM. Because of the novelty of the problem, there was not any benchmark problem, and generating a number of instances was necessary. Computational experiments were carried out on randomly generated instances. The instances are generated in two categories: small and medium-sized, and large-sized. For each category, 30 instances are generated. The details of instances are as follows:

The number of periods is equal to 2 for small/medium-sized instances and is chosen from $s\in\{2,3\}$ for large-sized instances. The number of shifts in each period is chosen from $d\in\{2,3,4,5\}$ for small/medium-sized and $d\in\{3,4,5,6,7,8,9\}$ for large-sized instances. The number of shifts in small-sized instances and number of periods and shifts in medium-sized instances are chosen in such a way that for an instance with more batches and more stages, the number of shifts (and periods) is relatively more. The duration of each shift is set to 45 for all instances. The number of batches is chosen from $n \in\{6,8,10,12,14\}$ for small-sized instances and $n \in\{20,25,30,35,40\}$ for medium-sized instances. Due date of each batch is specified at the end of one of periods. Batches are supposed to be different. The number of stages is chosen from $s\in\{3,4,5\}$ for small-sized instances and $s\in\{4,5,6,7\}$ for medium-sized instances. The processing times of batches in each stage are generated uniformly from [10,25] for all instances. The number of machines at each stage is chosen from $m\in\{1,2,3\}$ for small-sized instances and $m\in\{2,3,4\}$ for medium-sized instances. Sale price of each batch is randomly generated from $sp\in\{200,300,400,500\}$ for small-sized and medium-sized instances. Inventory cost of each batch for every period is chosen randomly from a uniform distribution [10, 30]. Lost sale cost of each batch is chosen from $lsc\in\{100,200,300,400\}$. The maximum flow time is set to 90, 120, 150 and 180 respectively for the instances with 3, 4, 5 and 6 stages (regardless of the size of problem).

The maximum flow time may cause infeasibility of some instances, but for the generated instances, the maximum flow time is large enough compared to processing times, that the flow time of every batch does not exceed the maximum flow time. The procedure for constructing a complete schedule, explained in section 5, selects a batch and schedules it's start time and finish time from the first stage to the last stage as soon as possible, and then is continued with other batches. Therefore, the procedure maintains feasibility for all batches.

6-1- Computational results

The following tables (tables 2 and 3) show the experimental results of each problem set. The columns "Run time" report the computational times in seconds. Here, we have allocated a maximum running time of one hour (i.e. 3600s). It is a reasonable time for production planners to determine when and which offers to be manufactured during the planning horizon. It is noteworthy that in parameters selection for each of generated instances, parameters shown in Tables 2 and 3 are selected in such a way that available time is generally not sufficient to produce all of the batches, because if the time is long enough to produce all of the batches, the two compared methods, especially proposed Tabu search, would give the best solution relatively in a short time. Therefore, we ignored this kind of instances that lead to find the optimal solution in a short time for the proposed Tabu search.

Table 2. Numerical results for small-sized and medium-sized instances

Instance	No. of periods	No. of shifts in each period	No. of batches	No. of stages	No. of machines in each stage	Full-space model		Proposed Tabu search		Gap (%)
						Objective Function (profit)	Run time (s)	Objective Function (profit)	Run time (s)	
1	2	2-1	6	3	1-2-1	1581	0.20	1581	14.63	0.00
2	2	2-2	6	3	1-1-2	1200	0.09	1200	18.72	0.00
3	2	2-2	6	4	1-1-2-2	1800	0.44	1800	27.89	0.00
4	2	2-2	6	4	2-1-2-1	1300	0.27	1300	25.30	0.00
5	2	3-2	6	5	2-1-2-1-2	1700	0.83	1700	44.82	0.00
6	2	3-2	6	5	1-1-1-2-1	1600	0.38	1600	36.03	0.00
7	2	2-2	8	3	1-2-2	2300	1.39	2300	22.08	0.00
8	2	3-1	8	3	2-1-2	2100	3.98	2100	30.71	0.00
9	2	2-2	8	4	1-2-2-2	1300	34.48	1300	32.27	0.00
10	2	3-2	8	4	1-2-1-1	2000	2.63	2000	40.42	0.00
11	2	3-2	8	5	2-2-2-1-1	2200	31.56	2200	69.65	0.00
12	2	3-3	8	5	1-1-2-2-2	2700	38.78	2700	82.17	0.00
13	2	2-1	10	3	2-2-2	2900	106.25	2900	55.29	0.00
14	2	3-1	10	3	1-2-1	2944	53.59	2944	43.09	0.00
15	2	3-2	10	4	2-1-2-1	2600	95.03	2600	76.10	0.00
16	2	2-3	10	4	1-2-2-1	2700	43.16	2700	65.22	0.00
17	2	4-2	10	5	1-1-2-2-1	2100	68.83	2100	128.59	0.00
18	2	3-2	10	5	2-1-2-2-1	2100	116.64	2100	112.66	0.00
19	2	2-1	12	3	2-2-2	2700	819.58	2700	61.72	0.00
20	2	3-2	12	3	2-2-1	3675	3600	3675	85.44	0.00
21	2	2-2	12	4	2-2-2-2	2900	247.22	2900	97.28	0.00
22	2	3-2	12	4	2-1-2-2	3079	3418.84	3079	128.40	0.00
23	2	3-2	12	5	2-2-2-2-2	3700	3600	3700	165.94	0.00
24	2	4-2	12	5	1-2-2-1-2	3700	3600	3700	180.35	0.00
25	2	3-3	14	3	1-2-2	3500	189.34	3500	158.41	0.00
26	2	2-2	14	3	2-2-2	2300	634.20	2300	132.17	0.00
27	2	3-3	14	4	2-2-2-1	3481	3600	3881	183.94	(11.49)
28	2	2-2	14	4	2-2-2-2	3300	565.22	3300	205.66	0.00
29	2	4-3	14	5	1-1-2-2-2	3100	3600	3500	259.85	(12.90)
30	2	3-2	14	5	2-2-2-2-2	4200	3600	4200	282.71	0.00
									average	(0.81)

The characteristics and numerical results for small and medium size instances are presented in Table 2. The running time of each solution approach was limited to one hour, which is an acceptable time in order to support planning and scheduling decisions in practical settings. The proposed Tabu search obtained optimal solution in all of 24 instances, which were optimally solved by simultaneous planning and scheduling model. The computational time required for proposed Tabu search to achieve optimal solution was shorter in 11 instances of these 24 instances (mostly more complex instances). For 6 other instances which model couldn't reach optimal solution in the specified time, the proposed Tabu search achieved solutions equal or better than the mathematical model. The average gap for the proposed method, as shown in table 2, is 0.81%, and the gaps between two methods are notable in two instances (27 & 29) which are more complicated in the category of medium and small-sized instances. This indicates that the gap can become considerable as the size of the problem increases, but more instances are required to test this deduction that are large size instances.

Table 3. Numerical results for large size instances

Instance	No. of periods	No. of shifts in each period	No. of batches	No. of stages	No. of machines in each stage	Full-space model		Proposed Tabu search		Gap (%)
						Objective Function (profit)	Run time (s)	Objective Function (profit)	Run time (s)	
1	2	4-4	20	4	2-2-2-1	6500	3600	6500	373.14	0.00
2	2	3-3	20	4	2-2-2-2	4800	3600	5700	338.52	(18.75)
3	2	4-3-3	20	5	2-1-2-2-2	7400	3600	7400	521.27	0.00
4	2	4-3	20	5	2-2-2-2-2	6800	3600	6800	472.01	0.00
5	3	4-5-4	20	6	2-2-2-1-2-1	5500	3600	5500	706.33	0.00
6	2	4-4	20	6	2-3-3-2-2-2	4500	3600	5600	845.65	(24.44)
7	3	5-5	25	4	2-1-2-2	7400	3600	8600	589.90	(16.22)
8	2	4-3	25	4	2-2-3-2	7400	3600	7400	635.58	0.00
9	3	4-4-3	25	5	2-2-2-1-2	8280	3600	8680	829.38	(4.83)
10	2	4-4	25	5	2-3-2-3-2	7400	3600	8100	934.31	(9.46)
11	3	5-4-5	25	6	2-2-1-2-2-2	8800	3600	9300	1176.63	(5.68)
12	2	5-5	25	6	3-3-2-2-3-2	9100	3409.31	9100	1365.97	0.00
13	2	3-4	30	4	3-2-2-2	7400	3600	8300	977.14	(12.16)
14	2	3-2	30	4	3-3-3-3	5453	3600	7153	884.65	(31.18)
15	3	5-3	30	5	2-2-3-3-3	8078	3600	9278	1325.70	(14.86)
16	2	3-3	30	5	3-3-3-3-3	4200	3600	7400	1248.37	(76.19)
17	2	4-4-3	30	6	2-2-2-2-2-2	8098	3600	9398	1652.11	(16.05)
18	2	5-5	30	6	2-3-2-2-3-2	4400	3600	8700	1814.06	(97.73)
19	2	4-3	35	4	2-3-3-3	7637	3600	9600	1461.29	(25.70)
20	2	3-3	35	4	3-4-4-3	8100	3600	8700	1275.02	(7.41)
21	2	3-4-2	35	5	2-3-3-3-2	6387	3600	10100	1938.19	(58.13)
22	2	4-2	35	5	4-3-4-3-4	8600	3600	9200	1799.53	(6.98)
23	3	5-4-3	35	6	3-3-3-2-2-3	3685	3600	8885	2616.74	(141.11)
24	2	4-3-2	35	6	3-3-3-3-3-3	3789	3600	9800	2846.05	(158.64)
25	2	3-3-2	40	4	3-2-3-3	7576	3600	11500	1959.34	(51.79)
26	2	3-3	40	4	4-3-4-4	9471	3600	12671	1717.47	(33.79)
27	2	4-3-3	40	5	3-3-2-3-3	4254	3600	11000	2771.16	(158.58)
28	2	4-2	40	5	3-3-4-4-4	4575	3600	8875	2362.66	(93.99)
29	3	5-5-3	40	6	2-2-2-2-2-2	4750	3600	10400	3527.82	(118.95)
30	2	4-3-2	40	6	3-4-3-4-3-4	3639	3600	12600	3086.04	(246.25)
									average	(47.63)

In table 3, numerical results for large size instances are presented. The model was able to achieve optimal solution for only one instance. The proposed Tabu search was capable of obtaining better solutions in 24 instances within remarkably short times. The results of Table 3 unlike table 2, indicate that there is a wide range of values for the gaps between the exact and proposed method. It is obvious that overall, the larger the size of instance leads to larger gap. Because of Np-hardness of the problem, the exact method becomes intractable for more complicated problems and it is not capable of obtaining a good solution in a reasonable time, but the proposed method, since it does not evaluate all the feasible solutions can achieve better solutions. Therefore, the gap between the two methods becomes larger.

Comparing tables 2 and 3 also indicates that there are extremely larger amount for gaps in table 3, implying the superiority of proposed method. It is significant to mention this superiority is occurred despite the shorter running time.

7- Conclusions and future research

Integration of production planning and scheduling has received more attention in recent years. It is important to consider both production planning and scheduling in order to improve productivity. Planning horizons and periods are typically considered in integrated production planning and scheduling, but importing the assumption of a number of work shifts existing in each period is also necessary for this category of problems. This is because production is not generally continuous within time periods.

In this paper, the simultaneous production planning and scheduling in a hybrid flow shop environment problem was considered. Each time period contains several work shifts. This is the first study that considers hybrid flow shop environment, and existence of the planning horizon, planning periods and work shifts in integrated production planning and scheduling. The problem was first formulated as a mixed integer linear programming model. The objective is to maximize profit that is a productivity-based performance measure. The problem and the formulation differ significantly from the ones being investigated in the literature. Since the problem is novel, there was not any benchmark problem to compare exact method and the proposed Tabu search. Therefore, sixty different hybrid flow shop instances were generated and solved. The mathematical model was able to obtain optimal solutions in some of the generated instances, but it becomes intractable as the size of the problem increases. This is because of NP-hardness nature of the problem. Therefore, a Tabu search and a novel procedure for constructing a complete schedule were proposed and its computational results were compared with those obtained from the mathematical model. The computational results for generated instances showed that the proposed Tabu search obtained optimal solutions on all the instances solved optimally by the mathematical model, and it outperformed the exact method in other instances. Therefore, the proposed Tabu search performance is satisfactory in profit maximization for the simultaneous production planning and scheduling in hybrid flow shop organization.

In the case of future studies, we can suggest other kinds of scheduling environments like job shop or flexible job shop. Furthermore, adding other considerations like non-identical parallel machines, limited buffer capacities, availability of raw material, and overtime work to the model. Moreover, applying other methods of integration like decomposition methods or using other metaheuristics. Finally, changing the objective function by adding or removing some terms to it (e.g. backorder cost instead of lost sale cost) or consideration of more than one objective in the mathematical model.

References

Aguirre, A.M., & Papageorgiou, L.G. (2018). Medium-term optimization-based approach for the integration of production planning, scheduling and maintenance. *Computers and Chemical Engineering*, 116, 191–211.

Attar, S.F., Mohammadi, M., & Tavakkoli-Moghaddam, R. (2013). Hybrid flexible flow-shop scheduling problem with unrelated parallel machines and limited waiting times. *International Journal of Advanced Manufacturing Technology*, 68(5-8), 1583-1599.

Cho H.-M., & Jeong I.-J. (2017). A two-level method of production planning and scheduling for bi-objective reentrant hybrid flow shops. *Computers and Industrial Engineering*, 106, 174-181.

Ebrahimi, M., Fatemi Ghomi, S.M.T., & 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.

Erdirik-Dogan, M., & Grossmann, I.E. (2006). A decomposition method for simultaneous planning and scheduling of single-stage continuous multiproduct plants. *Industrial and Engineering Chemistry Research*, 45(1), 299–315.

Erdirik-Dogan, M., & Grossmann, I.E. (2008). Simultaneous planning and scheduling of single-stage multi-product continuous plants with parallel lines. *Computers and Chemical Engineering*, 32(11), 2664–2683.

Glover, F. (1989a). Tabu search, part I. *ORSA Journal on Computing*, 1(3), 190–206.

Glover, F. (1989b). Tabu search – Part II. *ORSA Journal on Computing*, 2(1), 4-32.

Gupta, J.N.D. (1988). Two-stage hybrid flow shop scheduling problem. *Journal of the Operational Research Society*, 39 (4), 359–364.

Hoogeveen, J.A., Lenstra, J.K., & Veltman, B. (1996). Preemptive scheduling in a two-stage multiprocessor flow shop is np-hard, *European Journal of Operational Research*, 89 (1), 172–175.

Hosseini, S.M.H. (2017). A multi-objective genetic algorithm (MOGA) for hybrid flow shop scheduling problem with assembly operation. *Journal of Industrial and Systems Engineering*, 10, 132-154.

Jankauskas, K., Papageorgiou, L.G., & Farid, S.S. (2019). Fast genetic algorithm approaches to solving discrete-time mixed integer linear programming problems of capacity planning and scheduling of biopharmaceutical manufacture. *Computers and Chemical Engineering*, 121, 212–223.

Kim, S.H., & Lee, Y.H. (2016). Synchronized production planning and scheduling in semiconductor fabrication. *Computers & Industrial Engineering*, 96, 72-85.

Li, J.-Q., & Pan, Q.-K. (2015). Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. *Information Sciences*, 316, 487-502.

Li, J.-Q., Sang, H.-Y., Han, Y.-Y., Wang, C.-G., & Gao. K.-Z. (2018). Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions. *Journal of Cleaner Production*, 181, 584-598.

Low, C., Hsu, C.-J., & Su, C.-T. (2008). A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. *Computers & Operations Research*, 35 (3), 845-853.

Mousavi, S.M., Mahdavi, I., Rezaeian, J., & Zandieh, M. (2018). Bi-objective scheduling for the re-entrant hybrid flow shop with learning effect and setup times. *Scientia Iranica E*, 25(4), 2233-2253.

Nejati, M., Mahdavi, I., Hassanzadeh, R., & Mahdavi-Amiri N. (2016). Lot streaming in a two-stage assembly hybrid flow shop scheduling problem with a work shift constraint. *Journal of Industrial and Production Engineering*, 33(7), 459-471.

Pan, Q.-K., Gao. L., Li, X.-Y., & Gao. K.-Z. (2017). Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times. Applied Mathematics and Computation, 303, 89-112.

Rajaee Abyaneh, F., & Gholami, S. (2015). A comparison of algorithms for minimizing the sum of earliness and tardiness in hybrid flow-shop scheduling problem with unrelated parallel machines and sequence-dependent setup times. *Journal of Industrial and Systems Engineering*, 8(2), 67-85.

Ramezanian, R., Fallah Sanami, S., & Shafiei Nikabadi, M. (2016). A simultaneous planning of production and scheduling operations in flexible flow shops: case study of tile industry. *International Journal of Advanced Manufacturing Technology*, 88 (9-12), 2389-2403.

Riane, F., Artiba, A., & Iassinovski, S. (2001). An integrated production planning and scheduling system for hybrid flowshop organizations. *International Journal of Production Economics*, 74 (1–3), 33–48.

Safari, G., Hafezalkotob, A., & Khalilzadeh, M. (2017). A novel mathematical model for a hybrid flow shop scheduling problem under buffer and resource limitations-A case study. *Journal of Industrial and Systems Engineering*, 10, 58-77.

Soltani, S., & Karimi, B. (2015). Cyclic hybrid flow shop scheduling problem with limited buffers and machine eligibility constraints. *International Journal of Advanced Manufacturing Technology*, 76(9-12), 1739-1755.

Terrazas-Moreno, S., & Grossmann, I.E. (2011). A multiscale decomposition method for the optimal planning and scheduling of multi-site continuous multiproduct plants. *Chemical Engineering Science*, 66(19), 4307–4318.

Wang, X., & Tang, L. (2009). A tabu search heuristic for the hybrid flowshop scheduling with finite intermediate buffers. *Computers & Operations Research*, 36(3), 907–918.

Wardono, B., & Fathi, Y. (2004). A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities. *European Journal of Operational Research*, 155(2), 380–401.

Yu, C., Semeraro, Q., & Matta,A. (2018). A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Computers & Operations Research*, 100, 211–229.

Zandieh, M., Fatemi Ghomi, S.M.T., & Moattar Husseini, S.M. (2006). An immune algorithm approach to hybrid flow shops scheduling with sequence dependent setup times. *Applied Mathematics and Computation*, 180 (1), 111–127.

Zhang, X.Y., & Chen, L. (2018). A re-entrant hybrid flow shop scheduling problem with machine eligibility constraints. *International Journal of Production Research*, 56(16), 1-13.