Heuristics for Scheduling a Two-stage Hybrid Flow Shop with Parallel Batching Machines: an Application on Hospital Sterilization Plant

Andrea Rossi, Alessio Puppato and Michele Lanzetta

Department of Mechanical, Nuclear and Production Engineering

University of Pisa, Via Diotisalvi 2, 56122 Pisa, Italy

email: lanzetta@unipi.it; tel.: +39 050 2218122; fax: +39 050 2218140

Abstract

The model of a two-stage hybrid (or flexible) flow shop, with sequence-independent uniform setup times, parallel batching machines and parallel batches has been analyzed with the purpose of reducing the number of tardy jobs and the makespan in a sterilization plant. Jobs are processed in parallel batches by multiple identical parallel machines. Manual operations preceding each of the two stages have been dealt with as machine setup with standardized times and are sequence-independent. A mixed integer model is proposed. Two heuristics have been tested on real benchmark data from an existing sterilization plant: constrained size of parallel batches and fixed time slots. Computation experiments performed on combinations of machines and operators numbers suggest balancing the two stages by assigning operators proportionally to the setup time requirements.

Keywords: Scheduling; Hybrid Flow Shop; Parallel Batching; Heuristics; Healthcare.

1. Introduction

This paper is derived by a concrete case. The primary purpose was to improve the effectiveness and the efficiency of an existing hospital sterilization plant, from the continuous push of increasing safety, quality level and reducing the high costs of health services. The problem of assigning jobs to machines to make a better use of resources is called scheduling and is being extensively studied in manufacturing, logistics, computer sciences etc.

The first purpose of this work was to model the existing plant by analogy to manufacturing. The proposed model is a two-stage hybrid (or flexible) flow shop, with sequence-independent uniform setup times, parallel batching machines and parallel batches.

This apparently new model is applicable to similar problem, like continuous casting (steel-making), and coating (heat and galvanic treatments, painting). Investigating this scheduling problem is also important as it affects the logistics targets with due-date reliability/no tardy jobs and small makespan but also high capacity utilization and low inventory levels.

The effect of two proposed heuristics, namely constrained size of parallel batches and fixed time slots, has been experimentally investigated on both real and simulated data, with the scheduling criteria of reducing delayed jobs and the total completion time.

Figure 1: The sterilization plant at the AOUP hospital in Pisa

about here

The sterilization plant under study is shown in Figure 1. Surgical kits (jobs) achieve a standardized sterility assurance level by a machine-washing and steam sterilization cycle. The overall process consists of the following stages: (i) **washing**, including check-in, manual rinsing and mechanical washing, (ii) **sterilization**, including packing, steam sterilization, and finally return to operating units within job deadline, which is determined by the surgical planning. Manual operations preceding washing and sterilization have been dealt with as machine setup. Setup times depend on the surgical kit and operator's skill. Setup times are *batch sequence-independent* i.e. they only depend on the current batch to be processed and not on the previous one (Allahwerdi et al., 2008). The *setup speed* depends on the actual number of operators on the two stages. A fixed number of operators is assigned to each of the two stages at the beginning of each shift, which also represents the rolling horizon of the scheduling system under study.

All jobs, which are delivered at the sterilization plant (*release date*) at prefixed interval of times (*time windows*), have the same routing through the two stages. After setup, jobs are processed by one of the *identical parallel machines* at each stage. The two stages include respectively three washers and four sterilizers. The machine time within a stage is the same for all jobs.

Each machine has a finite capacity, i.e. it is able to process one or more jobs simultaneously, so they are processed in batches. There are two types of batch productions, namely, *serial batches* and *parallel batches*. In serial batches, jobs of the same batch are processed sequentially, while in parallel batches they are processed simultaneously (Hopp and Spearman, 2000).

The problem of *parallel batching machines* in a flow shop system for processing parallel batches has been introduced by Bellager and Oulamara (2009).

The following performance indices are considered: minimizing the number of tardy jobs and the makespan.

This paper models the AOUP sterilization plant as a hybrid (or flexible) flow shop environment where washing and sterilization are two stages of a flow shop system and the surgical kits are the jobs.

Tardy jobs (surgical kits) will cause surgery rescheduling, with heavy medical and economic consequences, hence, represent the most important scheduling target. Life threatening kits are always in stock, they are a small percentage of jobs and generally receive higher priority when going into the system.

The makespan is also considered in this study, because a lower makespan means less idle time, higher machine utilization and efficiency; consequently a more profitable use of resources.

A flow shop environment is similar to a job shop with unidirectional flow through production stages. A hybrid flow shop is a flow shop with at least one stage with more than one machine.

Scheduling problems can be described by a triplet $\alpha|\beta|\gamma$ according to the notation of Graham et al. (1979) where field α denotes the system layout and the production flow type, field β indicates the operation characteristics and field γ denotes the adopted performance indices.

The current problem can be formulated as:

$$FP2B(m_1, m_2)|p\text{-}batch, ST_{si,b}|(\Sigma U_i, \mathrm{C}_{\max})$$

where a two-stage hybrid flow shop *FP2* with m_1 and m_2 parallel batching machines per stage (*B*) processes parallel batches (*p-batch*) with batch sequence-independent setup times ($ST_{si,b}$) in order to minimize the number of tardy jobs ΣU_i and the makespan $\mathrm{C}_{\max}$.

The case of only one stage with capacity and speed of all machines equal to one can be reduced to $P||\mathrm{C}_{\max}$ which is NP-hard according to Garey and Johnson (1978). Therefore the time complexity function of the $FP2B(m_1, m_2)|p\text{-}batch, ST_{si,b}|(\Sigma U_i, \mathrm{C}_{\max})$ problem is NP-hard. Finding an optimum in a reasonable time is unlikely, hence heuristics should be employed.

2. Literature

A survey of scheduling literature on hybrid flow shop environments is available from Ribas et al. (2010). According to Gupta (1988), the two-stage hybrid flow shop scheduling problem *FP2* is NP-hard, even in the simplest case, with only 1 machine on the first stage and 2 machines on the second stage. Due to NP-hardness, two solution types were proposed: branch and bound algorithms and heuristic approaches. Narasimhan and Panwalkar (1984) considered a real-life two-stage hybrid flow shop with 1 machine at stage one and 2 machines at stage two. The cumulative minimum deviation (CMD) rule was suggested for reducing the sum of machine idle time and in-process job waiting time. Later, Narasimhan and Mangiameli (1987) extend the CMD rule with five criteria. In their $FP2B(m_1, m_2)|p\text{-}batch|C_{\max}$ problem the material is processed continuously at stage one, consisting of multiple and identical machines, and then batch processed on the multiple repetitive machines at stage two. Gupta and Tunc (1991) proposed two heuristics to find a minimum makespan schedule for the case of only 1 machine at stage one. The lower bounds on the makespan were also discussed. Deal and Hunsucker (1991) studied the $FP2(m, m)||C_{\max}$ problem with identical number of machines at the two stages. A lower bound calculation for the makespan was introduced and employed to evaluate the performance of three job sequencing rules in conjunction with a FIFO (first-in, first-out) manner. Gupta et al. (2002) considered hybrid flow shop scheduling with controllable processing times and *agreeable* release and due dates (i.e. $r_i \leq r_o \Rightarrow d_i \leq d_o$). They proposed constructive algorithms using job insertion techniques and iterative algorithms based on local search. Kyparisis and Koulamas (2006) surveyed scheduling literature in heuristics for worst-case ratio and suggested a new heuristic for minimizing makespan, which gives a worst-case performance guarantee when the speed of parallel machines in a given stage vary significantly and provided a definition of *uniform* machines for hybrid flow shop (i.e. parallel machines with different speed). Lin and Liao (2003) proposed a heuristic to minimize the weighted maximal tardiness in a real two-stage hybrid flow shop environment with sequence-dependent setup time in the first stage. Lee and Kim (2004) suggested a branch and bound algorithm for the two-stage hybrid flow shop with parallel machines only at the first stage with the objective of minimizing the total tardiness. The objective of minimizing the number of tardy jobs was considered by Gupta and Tunc (1998) who suggested several heuristic algorithms for two-stage hybrid flow shops with parallel machines only at the last stage. They also designed a procedure for producing neighborhoods that generate better solutions. Choi and Lee (2007) considered a two-stage hybrid flow shop with one or more parallel machines at both stages, and suggested a branch and bound algorithm that minimizes the number of tardy jobs. As branch and bound algorithms are time-consuming in practical applications (usually large-sized), the same authors suggested a two-phase heuristic algorithm (2009).

In the sterilization plant, each machine is able to process two or more jobs simultaneously (*batch*). We consider a batch scheduling problem where identical parallel batching machines are available in a flow shop system for processing parallel batches.

In addition to the mentioned precursory work of Narasimhan and Mangiameli, recent works deal with the scheduling problem in a two-stage hybrid flow shop with parallel batching machines, but most include the limitation of parallel batching machines at the last stage only. Bellager and Oulamara (2009) considered the $FP2B(m_1, m_2)|p\text{-}batch(\mathrm{II})|C_{\max}$ problem with a number of parallel batching machines at the second (II) stage only. They provided various lower bounds for heuristics and worst-case solution. Inversely, Luo et al. (2011) considered $FP2B(3, 1)|p\text{-}batch|C_{\max}$ with three parallel machines in the first stage and one machine in the second stage with sequence-dependent setup times and improved manual schedule by heuristics. Liu et al. (2010) minimized the maximum completion time in a hybrid flow shop $FPkB(m_1, \ldots, m_k)||C_{\max}$ from polypropylene batch industries by hybrid particle swarm optimization.

Amin-Naseri and Beheshti-Nia (2009) considered the $FP3B(m_1, m_2, m_3)|p\text{-}batch|C_{\max}$ problem. They proposed a mixed integer programming and three heuristics inspired by the Johnson's rule and a heuristic based on the NEH algorithm for parallel machines and the theory of constrains. They developed a genetic algorithm with a three dimensional structure of chromosomes (jobs, stages and machines in a stage), which outperforms all the heuristics.

Under certain conditions on batches formation, Kim et al. (1997) reduced the hybrid flow shop with parallel machines to a standard flow shop with identical parallel machines at each stage and applied the Johnson's rule. Critical conditions for real scheduling problems are: 1) sizes of transfer batches are common multiples of the number of machines at two stages and given and 2) production lot sizes of all jobs are multiples of their transfer batch sizes and known. However the heuristic used to reduce to the standard flow shop for transfer batches between stages can be used to achieve an upper bound for the makespan. Once released to the shop, orders are processed at the machining centers in earliest due date order.

The works mentioned considered negligible or sequence-dependent setup times. The batch sequence-independent setup times can be derived from the group technology assumption (Huang and Li, 1998, and Quadt and Kuhn, 2007). Jobs belonging to the same product can be grouped in batches and a single setup per product is performed.

Huang and Li (1998) considered the $FP2B(1, m_2)|ST_{si,b}|C_{\max}$ problem where the first stage consists of only 1 machine and the second stage consists of uniform parallel machines and the objective function of minimizing the makespan. They presented two heuristics and derived a model to determine the trade-off between costs and speeds of the machines at the second stage. Quadt and Kuhn (2007) considered a hybrid flow shop $FPkB(m_1, \ldots, m_k)|s\text{-}batch|(SC_{sd}, n^{-1}\Sigma f)$ with setup costs when changing product type or otherwise neglected. Parallel machines processed serial batches at each stage. Job process times were assumed identical at each stage. When all jobs are available at the time origin, there are no setup times and process times can be assumed identical for all jobs, the batch formation becomes a bin-packing problem: setting up batches that saturate the machines capacity at a given stage (Kim et al., 1997). As bin-packing problem is NP-hard, Quadt and Kuhn (2007) approached the problem by two genetic algorithms with a novel representation scheme based on a product sequence instead of a job sequence. Other representation schemes are based on disjunctive graphs (Rossi and Dini, 2007), which can be used generate constructive solutions by metaheuristics algorithms (Rossi and Lanzetta, 2012).

Xuan and Tang (2007) took into account a hybrid flow shop $FP3B(m_1, m_2, m_3)|s\text{-}batch(\text{I-II}), ST_{si,b}|C_{\max}$ for steelmaking, continuous casting and refining. They considered batch sequence-independent setup times on parallel batching machines at the last stage that process serial batches, and discrete parallel machines at the others stages. A machine incurs setup when switching between two batches. For this reason, a sequence-independent setup time is always considered and it is separated from the processing time of the batch. A setup is anticipatory, meaning that the setup of the next batch can start as soon as a machine becomes free to process the batch (Allahwerdi et al., 2008). They used a Lagrangian relaxation algorithm with capacity constraints to approach the problem. The relaxed problem was decomposed into batch-level subproblems, each for a specific batch.

It seems that the two-stage hybrid flow shop examined in this paper, with parallel batching machines at each stage for processing parallel batches with sequence-independent setup times, has not been dealt with in the literature. In addition, a definition of *uniform setup* inspired from that originally proposed by Kyparisis and Koulamas (2006) is applied to setup instead of machines.

3. Problem formulation

Formally, there is a set of surgery kits (or jobs) $i=\{1,2,..,N\}$ for processing in batches on a two-stage flow shop with m_j identical parallel machines included at stage j. Each machine h at stage j has a capacity u_j, a processing time p_j, a setup speed v_j and a batch size b_j identical for all the parallel machines of the stage that is a fixed percentage of the machine capacity $(\delta \cdot u_j)$, $\delta \in [0,1]$. Each job i is available from a release date r_i onwards, has a setup time s_{ij} on a machine of stage j, a size z_i, a priority w_i, and a due date d_i before which the job is expected to complete.

The setup time of a parallel batch depends on the number of operators at each stage. It is the sum of the setup times of the jobs it contains in the case of a single operator.

The impact of speed v_j is that the stage j can carry out v_j units of setup in one time unit. $[s_{ij} / v_j]$ is the actual setup time of job i at stage j. The sum of speeds is a constant in the sterilization plant; the goal is to find the optimal relative difference of speed between the first and the latter stage in order to minimize the number of tardy jobs and the makespan.

To form a batch, surgery kits are placed on metal trays (up to 5 levels for washers) and in containers (up to 2 for sterilizers). The machine capacity at each stage is a multiple of the job sizes. If a larger surgery kit is present, one tray can be taken off to make enough room.

At each stage all jobs in the same batch are processed after setup. In parallel batch processing, the completion time of a job coincides with the belonging batch completion time, which is equal to the processing time p_j.

We make the following assumptions:

- each job i can be processed by at most one machine for each stage;
- no jobs have agreeable release and due dates (jobs with earlier release date do not necessarily have and earlier due date);
- jobs routing are unidirectional but not identical because few jobs (shown in Figure 1) cannot be machine-washed (*deleted job*).
- each job size is lower than the machine capacity and many jobs can be batched together respecting the machine capacity constraint;
- the machine capacity is a least common multiple of the job sizes;
- the setup time of parallel batches depends on the number of operators present at each stage;
- no preemption is allowed (operations will be uninterrupted);
- if the priority of job i is higher than the priority of job o, then job i must be completed before job o;
- loading and unloading times are included in the standardized setup times.

Notation

j	stage index, j=1,...,A, where A is equal to 2
h	machine index, h=1,...,M_j, where M_j is the total number of parallel machine at stage j
i	job index, i=1,...,N, where N is the total number of jobs

b batch index, b=1,...,B, where B is the total number of batches

k service sterilization operator index, k=1,.., v_1, v_1+1,.., V where V is the total number of operators and v_1 is the number of operators at stage 1 (also representing the setup speed at stage 1)

u_j machine capacity at stage j (i.e. all the machines at stage j have the same capacity)

r_i release date of job i; also used to update the job available time in the system

d_i due date of the job i

z_i size of job i

w_i priority of job i at stage 1

$s_{i\,j}$ setup time of job i at stage j

v_j units of setup carried out in one time unit

a_k release date of operator k (included for completeness, 0 in current problem)

p_j processing time of parallel machines at stage j

$C_{i\,j}$ completion time of job i at stage j

$C_{j\,h\,b}$ completion time of batch b of machine h at stage j

L_i lateness of job i (at the last stage A), $L_i = C_{i\,A} - d_i$.

U_i completion status of job i represented by U_i =1 if L_i> 0, 0 otherwise.

$BigM$ a large number → +∞

Decision variables

$X_{i\,j\,h}$ 1, if job i is assigned to machine h in stage j

0, otherwise

$Y_{k\,j}$ 1, if operator k is assigned to stage j

0, otherwise

$Z_{i\,j\,h\,b}$ 1, if job i is assigned to batch b of machine h at stage j

0, otherwise

4. The mixed integer model

A mixed integer problem formulation follows.

The objective functions are: minimizing the number of tardy jobs and the makespan:

Objective function 1: $\text{Min}\sum_{i=1}^{N} U_i$

Objective function 2: $\text{Min } \text{C}_{\max}$

Subject to the constraints:

$$\sum_{h=1}^{M_j} X_{ijh} = 1 \qquad i = 1,..,N \quad j = 1,..,A \qquad (1)$$

$$\sum_{j=1}^{A}\sum_{b=1}^{B} Z_{ijhb} = 1 \qquad i = 1,..,N \quad h = 1,..,M_j \qquad (2)$$

$$\sum_{i=1}^{N} z_i \cdot Z_{ijhb} \leq u_j \qquad h = 1,..,M_j,\ j = 1,..,A \quad b = 1,..,B \qquad (3)$$

$$\sum_{j=1}^{A}\sum_{k=1}^{V} Y_{kj} = v_1 + \sum_{j=2}^{A}\sum_{k=v_1+1}^{V} Y_{kj} = V \qquad (4)$$

$$C_{ij} \geq C_{jhb} - BigM(1 - Z_{ijhb}) \qquad h = 1,..,M_j,\ j = 1,..,A \quad i = 1,..,N \quad b = 1,..,B \qquad (5)$$

$$C_{jhb} \geq C_{i(j-1)} + s_{ij} + p_j - BigM(1 - Z_{ijhb}) \qquad h = 1,..,M_j,\ j = 2,..,A \quad i = 1,..,N \quad b = 1,..,B \qquad (6)$$

$$C_{jhb} \geq \max\left\{C_{i(j-1)} + s_{ij}, C_{jh(b-1)}\right\} + p_j - BigM(1 - Z_{ijhb}) \qquad h = 1,..,M_j,\ j = 2,..,A \quad i = 1,..,N \quad b = 2,..,B \qquad (7)$$

$$C_{ij} \geq \max\left\{C_{i(j-1)} + s_{ij}, C_{oj}\right\} + p_j - BigM(2 - Z_{ijhb} - Z_{ojh(b-1)}) \qquad h = 1,..,M_j,\ j = 2,..,A \quad i = 1,..,N \quad b = 2,..,B \qquad (8)$$

$$C_{i(j+1)} \cdot Z_{ijhb} \geq \min_{i=1,...,N}\left[C_{ij} \cdot Z_{ijhb}\right] + \left[\frac{\sum_{i=1}^{N} s_{i(j+1)} \cdot Z_{ijhb}}{v_j}\right] + p_j - \\ - BigM(1 - Z_{ijhb}) \qquad h = 1,..,M_j,\ j = 1,..,A-1 \quad i = 1,..,N \quad b = 1,..,B \qquad (9)$$

Constraint (1) assures that each job is only assigned to one machine for each stage. Constraint (2) guarantees that each job is assigned exactly to one batch for each stage. Constraint (3) assures that the number of jobs included in a parallel batch does not exceed the capacity of the assigned machine. Constraint (4) describes the relationship between setup speed at each stage and number of operators. The assignment of operators to the two stages by the parameter v_1 is a degree of freedom and is evaluated for optimality in computation experiments, ones their total number V is fixed.

Constraint (5) assures that the completion time of a job cannot be lower than the completion time of the belonging batch. Constraint (6) describes the relation between job completion time and completion time of the belonging batch at the subsequent stage. Similarly to (6), the relation in constraint (7) is between two subsequent batches of the same machine. Again in constraint (8) the relation is between two jobs of subsequent batches on the same machine. In constraint (9) the job completion time is higher than the early completion time at the previous stage plus the sum of the processing time and the *actual batch setup time* defined as

$$\left[(\sum_{i=1}^{N} s_{i\,j} \cdot Z_{i\,j\,h\,b}) / v_j \right], \quad j = 1,...,A \qquad (10)$$

5. The proposed heuristic algorithms

The developed scheduling method is reported in Listing 1 and performs batch forming considering the job priority and satisfying all other constraints (1) to (9).

Listing 1: Pseudo code with the scheduling algorithm implemented

```
1.  Arrange jobs in a list $L^1$ according to descending priority $w_i$ and, in case of ties ($w_i = w_o$) arrange jobs according to increasing due dates $d_i$ and $d_o$.
    Set stage $j=1$, batch index $b$=0, completion time $C_{i0}= r_i$ for all jobs and time slot index T=0
2.  Select the highest priority job $i^*$ from $L^1$
3.  Assign the first job $i^*$ to the first available operator $k^* \in \{1,.., v_1\}$ if $h=1$, $k^* \in \{ v_1+1,.., V\}$ otherwise
4.  Evaluate the completion time of the setup phase for job $i^*$ increasing the available time of operator $k^*$: $r_{i^*} \leftarrow \max\ \{C_{i^*j}, a_{k^*}\} + s_{i^*j}$
5.  Insert job $i^*$ in a list $L^2$ according to $r_{i^*}$ and update its completion time: $C_{i^*j} = r_{i^*}$
6.  Remove job $i^*$ from $L^1$
7.  If $L^1$ is empty go to step 8, else go to step 2
8.  Set the binary digit close_batch$_h$ to 0 for each machine $h$ = 1,.., $M_j$
9.  Apply rule FCFS (first-come, first-served) by the selection of job $i^*$ from $L^2$
```

```
10.   Parallel batch forming: for each machine $h*$ at stage $j$ the batch of jobs ($b$+$h*$) is formed
11.   If all the batches are closed, update the availability time of jobs: $C_{i\,j} \leftarrow C_{i\,j\,+}\,p_j$
12.   If $L^2$ is empty go to step 14 else go to 13
13.   Set $b \leftarrow b+ M_j$ and go to step 8
14.   $j \leftarrow j+1$
15.   If $j$=2 go to step 2
16.   Evaluate $\sum_{I=1}^{N} U_i = \{|i| \mid C_{i\,2} - d_i > 0\}$ and $C_{max} = \max_{i=1,..,N} C_{i\,2}$
```

Based on the specific constraints of this case study we formulate a heuristics for the mixed integer model in order to allocate operators at the two stages.

Two heuristic algorithms H^{❶} and H^{❷} are proposed in order to test the impact of different batch forming criteria on performance.

According to these two heuristics, batches are closed also without completion respectively at fixed times or before a given capacity threshold is reached. A pictorial view of the different cases available for combinations of H and δ (in grayed boxes) is reported in Figure 2.

Figure 2: Batch closing as a function of elapsed time and batch size in different conditions for the proposed heuristics

about here

The main difference between the two proposed heuristics is how they approach the constrained size of parallel batches and fixed time slots, respectively.

With heuristic H^{❶} machines on the two stages start when parallel batches reach a fixed fraction δ of the machine capacity u_j. Heuristic H^{❷} acts like a system clock, which determines the batch closing independently on the batch size.

From Figure 2 it can be noticed that for H^{❷} and δ<1 two cases are possible: on the left, the batch size $\delta{\cdot}u_j$ is reached before the time slot has elapsed; on the right, the batch size $\delta{\cdot}u_j$ is not reached when the time slot has elapsed.

The two columns show a direct relationship among heuristics: on the left, it can be observed that for $\delta < 1$ and the same batch size ($\delta{\cdot}u_j$) the batch formed by H^{❷} is equal to the one formed by H^{❶}; on the right, it can be observed that the batch formed by H❷ for $\delta < 1$ is equal to the one formed by H❷ for $\delta=1$.

As the batch closing mechanism affects the lot sizing, the first heuristic H❶ uses a batch-sizing criterion based on the fixed percentage limit δ of the machine capacity in fulfilling constraint (3):

$$\sum_{i=1}^{N} z_i \cdot Z_{i\,j\,h\,b} \le \delta \cdot u_j \qquad \begin{array}{l} h = 1,..,M_j,\ j = 1,..,A \\ b = 1,..,B \\ \delta \in [0,1] \end{array} \qquad (11)$$

In heuristic H❶, the steps 10. and 11. in Listing 1 are replaced by Listing 2.

Listing 2: Pseudo code for heuristic H❶

```
a.      If the set $M = \{h \mid \text{close\_batch}_h = 0\}$ is not empty:
a.1         Select a machine $h^* \in M$
a.2         Apply the close batch rule for machine $h^*$:
            $z_{i^*} \cdot Z_{i^*\,j\,h^*\,(b+h^*)} + \sum_{i=1}^{N} z_i \cdot Z_{i\,j\,h^*\,(b+h^*)} > \delta \cdot u_j$
a.3         If the batch must be closed, set close_batch$_h$ to 1 and evaluate $C_{i\,j} \leftarrow C_{i\,j} +$
            $p_j$ for all jobs of batch and go to step a.1
a.4         Else insert job $i^*$ in batch $(b + h^*)$ of a machine $h^* \in M$, i.e set $Z_{i^*\,j\,h^*\,(b+h^*)}$
            $= 1$ and remove job $i^*$ from $L^2$
```

Heuristic H❷ is inspired by *time slots* of jobs availability. Time slot is defined as

$$\{\mathrm{T} \cdot p_j \mid \mathrm{T} \in \aleph\}, j=1,...,A \qquad (12)$$

To reduce the machine idle time due to setup, jobs available after a time slot of duration equal to the machine processing time p_j are included in the next batch. In this case, the steps 10. and 11. in Listing 1 are replaced by Listing 3. For the FCFS rule, the first job that violates the last condition at step a. implies the close of all opened batches. When a batch is closed, the machine immediately starts operation.

Listing 3: Pseudo code for heuristic H❷

```
a.    If the set $M = \{h \mid \text{close\_batch}_j = 0\}$ is not empty and the availability times $r_{i*}$ of job $i*$ verifies $r_{i*} \leq T \cdot p_j$:
a.1       Select a machine $h^* \in M$
a.2       Apply the close batch rule for machine $h^*$: $z_{i^*} \cdot Z_{i^*\ j\ h^*\ (b+h^*)} + \sum_{i=1}^{N} z_i \cdot Z_{i\ j\ h^*\ (b+h^*)} > \delta \cdot u_j$
a.3       If the batch must be closed, set close_batch$_h$ to 1 and evaluate $C_{i\,j} \leftarrow C_{ij} + p_j$ for all job of the batch and go to step a.1
a.4       Else insert job $i*$ in the batch $(b + h*)$ of a machine $h^* \in M$, i.e set $Z_{i^* jh^* (b + h^*)} = 1$ and remove job $i*$ from $L^2$
a.5       If $L^2$ is empty go to step 14. else go to step 9.
b.    Case of exit loop a.
b.1       M is empty: go to step 12.
b.2.      $r_{i*} > T \cdot p_j$: set $T \leftarrow T+1$, set close_batch$_h$ to 1 for all the opened batches and evaluate $C_{i\,j} \leftarrow C_{i\,j} + p_j$ for all jobs of these batches
```

6. Computation experiments

The performance of the proposed heuristics, implemented with Java SE 6 as in Listing 1 to Listing 3, have been tested on real data. The scheduling input parameters of 60 jobs from the sterilization plant at the AOUP hospital in Pisa, Italy on a peak day selected after monitoring the plant for several months have been included as supplementary online material. The release date represents the surgery kit delivery at the sterilization plant in time windows. The operating units listed belong to six surgery departments involved on that particular day.

The manual setup times for the specific jobs have been determined by directly monitoring the sterilization plant for several months. They represent an estimate on a statistical basis of the different operators' skill and have been standardized, with a normal distribution with mean 2 and standard deviation of respectively 12 and 18 on the two stages, and limited in the ranges 1–30 and 1–45.

Standard times seem the only practically viable approach, because it would be very time consuming to collect the performance of all possible operators (currently 20) for all available surgical kits (about 800 including replicated kits). It would also be very difficult to assign a specific job to a specific operator within each stage. Replacing standard times with a probabilistic approach is not expected to significantly affect the results for the averaging effect among the many jobs involved and considering the normal distribution of setup times.

A fixed number of operators is assigned to the two stages at the beginning of each 24 hours (=1440 minutes) period, which also represents the rolling horizon of the scheduling system under study. The operators assignment to the two stages has been exhaustively tested considering their low number.

The lower bound for the makespan for the examined case with the hypothesis of unlimited operators and machine capacity has been estimated as 624.3 minutes using the expression:

$$\max_{i=1,..,N} LB(C_{i2}) \qquad (13)$$

where $LB(C_{i2})$ is the lower bound for the completion of job i (at stage 2) and is evaluated by the following expression:

$$LB(C_{i2}) = r_i + \sum_{j=1}^{2}\left(s_{ij} + p_j\right) \qquad (14)$$

Table 1 Result of simulations for heuristics $\text{H}^{❶}$ and $\text{H}^{❷}$ with two values of batch size and total operators number and assignment. The two stages include respectively M_1=3 and M_2=4 machines. The estimated lower bound of 624.3 min. for $C_{\max}$ is based on (13).

Case no.	Heuristics H (p)	Batch size δ (q)	Operators no. Total V (r)	On stage j=1 v_1 (s)	On stage j=2 V-v_1	$\sum_{i=1}^{N} U_i$	$C_{\max}$	%gap from lower bound for $C_{\max}$
1	1	1	7	5	2	10	867.2	38.9
2			7	2	5	4	682.7	9.4
3			7	4	3	6	747.6	19.8
4			7	3	4	4	695.8	11.5
5			6	4	2	10	867.2	38.9
6			6	2	4	4	707	13.2
7			6	3	3	6	748.9	20.0

8		0.8	7	5	2	10	889.8	42.5
9			7	2	5	4	681.3	9.1
10			7	4	3	4	759.8	21.7
11			7	3	4	4	700.7	12.2
12			6	4	2	10	888.5	42.3
13			6	2	4	4	710.5	13.8
14			6	3	3	4	763.8	22.3
15	2	1	7	5	2	3	867.2	38.9
16			7	2	5	0	676.2	8.3
17			7	4	3	2	745.3	19.4
18			7	3	4	0	691.1	10.7
19			6	4	2	3	867.2	38.9
20			6	2	4	0	700.9	12.3
21			6	3	3	1	751.2	20.3
22		0.8	7	5	2	5	868.3	39.1
23			7	2	5	0	676.2	8.3
24			7	4	3	0	745.3	19.4
25			7	3	4	0	689.5	10.4
26			6	4	2	3	867.2	38.9
27			6	2	4	0	700.4	12.2
28			6	3	3	0	749.7	20.1

The 28 configurations tested are listed in Table 1 along with the two performance indices calculated (the number of tardy jobs and the makespan). The grayed rows in Table 1 list the assignment of 7 operators; at washing (and sterilization) v_1=2 (5), 3 (4), 4 (3) and 5 (2). Similarly for a different setup speed, with one operator less: V=6 and v_1=2, 3 and 4.

7. Results

The configuration parameters can be expressed in compact form by the quartet $\mathrm{H}^{p}|\delta^{q}|V^{r}|v_1^{s}$ with the quotes of p, q, r and s listed in Table 1. The short version $\mathrm{H}^{p}|\delta^{q}||$ indicates each set of seven configurations where the first two columns take the values H=p and δ=q.

Before this work, the loading criterion in the sterilization plant was to full machine capacity without time slots of jobs availability i.e. washers and sterilizers were loading at the maximum capacity as expressed with the configuration $H^{❶}|\delta^{1}||$.

Table 1 shows the results achieved by the heuristic $H^{❶}$ and $H^{❷}$ with batch size δ^{1} – full machine capacity – and $\delta^{0.8}$ on the worst-case data, available as supplementary online material.

It can be noticed that the minimum number of tardy jobs is achieved with $H^{❷}$. In addition, the optimum schedule for tardy jobs has already been reached with 0 tardy jobs in 8 out of 14 cases.

Both with $H^{❶}$ and $H^{❷}$ the worst results are achieved when the assignment of operators among the two stages is strongly unbalanced inversely to the respective setup speeds.

The makespan C_{max} ranges from 676.2 (case $H^{❷}|\delta^{1}|V^{7}|v_{1}^{2}$, no. 16) to 889.8 minutes (case $H^{❶}|\delta^{0.8}|V^{7}|v_{1}^{5}$, no. 8). This shows that a wrong selection of heuristic H, batch size δ, total number of operators V and assignment on the two stages v_{i} causes a total batch processing time increase of 214 minutes (+32%).

The minimum makespan found is only 8.3% above the lower bound defined in (13).

With $\delta^{0.8}$, by removing one operator, the makespan increases of about 24 and 29 minutes with, respectively, the best heuristic (case $H^{❷}|\delta^{0.8}|V^{7}|v_{1}^{2}$, no. 23, makespan 676.2 versus case $H^{❷}|\delta^{0.8}|V^{6}|v_{1}^{2}$, no. 27 makespan 700.4) and the worst heuristic with δ=0.8 (case $H^{❶}|\delta^{0.8}|V^{7}|v_{1}^{2}$, no. 9, makespan 681.8 versus case $H^{❶}|\delta^{0.8}|V^{6}|v_{1}^{2}$, no. 13, makespan 710.5).

In most cases, the minimum makespan is achieved for $H^{❷}|\delta^{0.8}||$ and the worst tested heuristic is $H^{❶}|\delta^{0.8}||$. Full batch sizes (of both $H^{❶}$ and $H^{❷}$) offer an average performance. This shows that δ is a discriminating parameter for the proposed heuristics.

As for the operators assignment, with $H^{❶}|\delta^{1}||$ a wrong operator assignment (case $H^{❶}|\delta^{1}|V^{7}|v_{1}^{2}$ no. 2 versus $H^{❶}|\delta^{1}|V^{7}|v_{1}^{5}$ no. 1) may produce an increase of the number of tardy jobs from 4 to 10 and of the makespan of 184.5 minutes (+27%). Also within the best heuristics $H^{❷}|\delta^{0.8}||$ (case $H^{❷}|\delta^{0.8}|V^{7}|v_{1}^{2}$ no. 23 versus $H^{❷}|\delta^{0.8}|V^{7}|v_{1}^{5}$ no. 22) it provides an increase respectively from 0 to 5 and of 192.1 minutes (+28%).

Within the same heuristics, a reduction of one operator (cases no. 6 and 27 versus cases no. 2 and 23) produces no effect on the number of tardy jobs and an increase of the makespan of only 24.3 (+3.6%) and 13.3 minutes (+2%). This performance decrease is most probably tolerable compared to the relevant economic impact of reducing the operator number.

7.1 Benchmark tests

In the first set of tests, the worst case taken from the most critical day during the plant monitoring time has been considered. Among the three best solutions we have selected the heuristic H❷|$\delta^{0.8}|V^{6}|\nu_{1}^{2}$ no. 27 which includes fewer operators. It achieved no tardy job, a makespan of 700.4 minutes and a %gap of 12.2.

For system validation purposes, we have tested the heuristic on five benchmark problems available as supplementary online material. For each benchmark, the job parameters, namely priority, release and due dates and setup times for the two stages, have been generated using the following criteria:

- setup times have been generated using the above mentioned parameters;
- priority, release date and due date depend on the operating needs and delivery time windows, so the values and proportion from the worst-case available as supplementary online material have been kept fixed and randomly assigned to the 60 kits.

Table 2 Tardy jobs and makespan and respective lower bounds for the randomly generated benchmarks available as supplementary online material for heuristic H❷|$\delta^{0.8}|V^{6}|\nu_{1}^{2}$.

Benckmark n.	Lower bound for $\sum_{i=1}^{N} U_i$	Lower bound for C_{max}	$\sum_{i=1}^{N} U_i$	C_{max}	%gap from lower bound for C_{max}
1	7	620.6	7	689.0	9.9
2	6	633.5	7	709.2	10.7
3	6	607.9	6	671.2	9.4
4	5	629.0	6	719.1	12.5
5	4	633.5	5	676.2	6.3

The number of tardy jobs and the makespan for the five benchmarks are listed in Table 2. The lower bound for tardy jobs, which was always 0 in the worst-case dataset available as supplementary online material, is also listed. Benchmarks include jobs i with $LB(C_{i2}) > d_i$, which would not be acceptable in the real case.

The lower bound of the number of tardy jobs for each benchmark is evaluated by the sum of the completion status $U_i = 1$ for all the jobs i ($i=1,\ldots,n$)

$$\sum_{i=1}^{60} LB(U_i) = \sum_{i=1}^{60} \left\lfloor \frac{LB(L_i)}{\max[LB(C_{i2}), d_i]} + 1 \right\rfloor \qquad (15)$$

achieved when the lower bound of the lateness function defined by

$$LB(L_i) = LB(C_{i2}) - d_i \qquad (16)$$

is $LB(L_i) > 0$.

The lower bound for the makespan of the five benchmarks, calculated as in (13), is also listed in Table 2.

It can be observed that benchmarks are well established, the lower bound for $C_{\max}$ (between 607.9 and 633.5) being very close to the original dataset.

The number of tardy jobs in three out of five benchmarks (n. 2, n. 4 and n. 5) exceeds by 1 unit the lower bound. For the remaining benchmarks (n. 1 and n. 3) no job overruns the lower bound. The % gap from the lower bound of the makespan ranges between 6.3% and 12.5%.

8. Discussion

From the input data of the worst-case examined it can be observed that the system is overloaded at the start of the observation period and that job delivery is delayed. Consequently the fragmentation of batches by dispatching rules such as constrained size of parallel batches δ and fixed time slots $\{T \cdot p_j | T \in \aleph\}$ is beneficial. For instance, the combination of time slots and constrained size of parallel batches ($H^{❷}|\delta^{0.8}$) may reduce the batch size further (below 0.8), thus balancing setup among stages.

From the observation of the best production plans, the machine loading is only 20÷30% towards the beginning and the end of production plans, with intermediate values as high as δ (1 and 0.8). Consequently there is not an optimum fixed δ but as shown by simulations, operating with δ lower than full capacity is advised.

In the examined case, $H^{❷}$ is the dominating heuristic both for minimizing the number of tardy jobs and the makespan. This shows that the time slot of jobs availability seems the discriminating parameter in addition to δ.

It can be noticed that the optimum scheduling with no tardy jobs is achieved with $H^{❷}$, while the minimum achieved with $H^{❶}$ is 4.

The makespan minima shown in Table 1 occur when the actual batch setup times (defined in (10)) are similar, i.e. the operators are assigned to washing and sterilization proportionally to the respective setup times.

Both performance indices are achieved when the subdivision of operators among the stages is strongly unbalanced ($V^7|v_1{}^2$, $V^7|v_1{}^3$ and $V^6|v_1{}^2$).

Computation experiments suggest balancing the two stages by assigning operators proportionally to the setup time requirements and machine capacity. This way the plant can be considered as a continuous flow line with given cycle time and synchronized stages.

A reduction of one operator produces no effect on the number of tardy jobs. The slight increase of the makespan is most probably tolerable compared to the relevant economic impact of reducing the operator number. It is advised to switch operators in order to balance the actual setup speed among the stages.

The operator assignment seems a dominating parameter for the system performance making this more a layout design than a scheduling problem.

As for the correlation between the two objective functions, it can be noticed that the number of tardy jobs is higher for H❶, although the makespan for the two heuristics are in the same range. In addition, a direct correlation between tardy jobs and makespan within each heuristic is observed. Consequently, the minimization of tardy jobs is also an efficiency criterion.

The proposed system provides combinations of the two performance indices versus operator number, assignment, batch size and time slot. The number of tardy jobs can be used for a production volume at full capacity in order to reduce penalty for due dates overrun. The makespan must be used in case of delays of delivery to make up for idle times (the minimum C_{max} achieved in simulations is less than half of the observation period).

9. Conclusions

A scheduling system has been examined in order to simulate various production scenarios, different assignments and amounts of resources with the purpose of reducing the number of tardy jobs and the makespan. The proposed model is a two-stage hybrid flow shop with sequence-independent uniform setup times, parallel batching machines and parallel batches. To the best of our knowledge, the proposed configuration has not been dealt with in the literature and is available in many manufacturing (and other) processes, with concurrent machines in multiple stages processing several products in batches, and non negligible setup times.

The basic idea which can be exported to similar problems is that a better scheduling may be achieved with a constant flow, by

A mixed integer model has been proposed and two heuristics have been implemented and tested with two variations: constrained size of the parallel batches and fixed time slots.

Computation experiments on real data for a worst case have shown that the proposed heuristics are able to prevent tardy jobs and achieve a makespan that is about half of the scheduling horizon, providing significant economical benefits.

A preliminary set of tests has pointed out the relevant parameters (descending) influencing tardy jobs and makespan in the examined (worst) case: heuristic, balancing and number of operators and capacity, with a stronger interaction between heuristic and balancing or capacity.

The best parameter combination has been validated on five randomly generated benchmarks conservatively derived from the worst case examined.

The low algorithm processing time allows (i) switching in real-time the different heuristics and optimization criteria for a given list of surgery kits to be scheduled for sterilization for successive use in operating units; and (ii) assessing what-if scenarios, by adding or removing operators and/or machines.

Further investigations for the actual plant include: testing more heuristics (e.g. priority assignment, which currently is a manual process involving many also not measurable parameters) and their combinations for possible performance improvement; dynamic rescheduling after a given event, e.g. job arrival, job request, or machine/operator failure. Additionally, optimization algorithms, such as metaheuristics, may increase the performance and can be adapted to non deterministic conditions.

10. Acknowledgements

This work is derived from Dr. Alessio Puppato’s internship in mechanical engineering at Azienda Ospedaliero-Universitaria Pisana AOUP. The authors would like to thank Dr. Carlo R. Tomassini, Executive Director, and Dr. Marco Nerattini, Head of Operating Unit Innovazione e Sviluppo e Analisi dei Processi and the technical staff of the Central Sterile Services Department. A special thank to Dr. Silvia Pagliantini from Operating Unit Innovazione e Sviluppo e Analisi dei Processi for assistance in the data collection and analysis.

In depth and detailed analyses from the anonymous referees are also gratefully acknowledged.

11. References

Allahverdi A., Ng C.T., Cheng T.C.E., Kovalyov, M. Y. A survey of scheduling problems with setup times or costs. European Journal of Operational Research. 2008; 187:985–1032.

Amin-Naseri, M.R., Beheshti-Nia, M.A. Hybrid flow shop scheduling with parallel batching. International Journal of Production Economics. 2009; 117:185–196.

Bellager, A., Oulamara, A. Scheduling hybrid flowshop with parallel batching machines and compatibilities. Computers and Operations Research. 2009; 36:1982–1992.

Choi, H.-S., & Lee, D.-H. A branch and bound algorithm for two-stage hybrid flow shops: Minimizing the number of tardy jobs. Journal of the Korean Institute of Industrial Engineers. 2007; 33:213–220.

Choi, H.-S., & Lee, D.-H. Scheduling algorithm to minimize the number of tardy jobs in two-stage hybrid flow shops. Computer & Industrial Engineering. 2009; 56:113–120.

Deal, D.E., Hunsucker, J.L. The two-stage flowshop scheduling problem with M machines at each stage. Journal of Information and Optimization Sciences. 1991 12:407–417.

Garey, M. R., Johnson, D. S. (1979). Computers and intractability: A guide to the theory of NP-completeness. San Francisco: W.H. Freeman and Company.

Graham R.L., Lawler E.R., Lenstra J.K., Rinnooy Kan A.H.G. Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discrete Math 1979;5:287–326.

Gupta, J. N. D., & Tunc, E. A. Minimizing tardy jobs in a two-stage hybrid flowshop. International Journal of Production Research. 1998; 36:2397–2417.

Gupta, J., Kruk, K., Lau, V., Werner, F., Sotskov, Y.N. Heuristics for hybrid flow shop with controllable processing times and assignable due dates. Computers and Operations Research. 2002; 29: 1417–1439.

Gupta, J.N.D., Tunc, E.A. Schedules for a two-stage hybrid flowshop with parallel machines at the second stage. International Journal of Production Research. 1991; 29:1489–1502.

Hopp W.J., Spearman M.L. Factory physics. 2nd ed., Boston: McGraw-Hill International Editions; 2000.

Kim J.S., Kang S.H., Lee S.M. Transfer batch scheduling for a two-stage flowshop with identical parallel machines at each stage. Omega Int J Manag Sci. 1997; 25:547–55.

Kyparisis G.J., Koulamas C. A note on makespan minimization in two-stage flexible flow shop with uniform machines. European Journal of Operation Research. 2006; 175:1321–1327.

Lee, G.-C., & Kim, Y.-D. A branch-and-bound algorithm for a two-stage hybrid flow shop scheduling problem minimizing total tardiness. International Journal of Production Research. 2004; 42: 4731–4743.

Lin, H., Liao, C.. A case study in a two-stage hybrid flow shop with setup time and dedicated machines. International Journal of Production Economics. 2003; 86:133–143.

Liu, B., Wang, L., Liu, Y., Qian, B., Jin, Y.-H., An effective hybrid particle swarm optimization for batch scheduling of polypropylene processes, Computers & Chemical Engineering. 2010, 34, 4:518-528, ISSN 0098-1354, 10.1016/j.compchemeng.2009.12.010.

Luo, H., Huang, G. Q., Feng Zhang, Y., Yun Dai, Q, Hybrid flowshop scheduling with batch-discrete processors and machine maintenance in time windows. International Journal of Production Research, 2011, 49(6):1575-1603. doi:10.1080/00207541003610262.

Narasimhan, S.L., Mangiameli, P.M. A comparison of sequencing rules for a two-stage hybrid flow shop. Decision Sciences. 1987, 18:250–265.

Narasimhan, S.L., Panwalkar, S.S.. Scheduling in a two stage manufacturing process. International Journal of Production Research. 1984; 22:555–564.

Pan, Q.-P., Wang, L., Effective heuristics for the blocking flowshop scheduling problem with makespan minimization, Omega, Volume 40, Issue 2, April 2012:218-229, ISSN 0305-0483, 10.1016j.omega.2011.06.002.

Quadt, D., Kuhn, H. Batch scheduling of jobs with identical process times on flexible flow lines. International Journal of Production Economics. 2007; 105:385–401.

Rossi, A., Lanzetta, M. Featuring Native Metaheuristics for Non-Permutation Flowshop Scheduling. Journal of Intelligent Manufacturing. 2012 (submitted).

Rossi, A., Dini, G. (2007) Flexible job-shop scheduling with routing flexibility and separable setup time using ant colony optimisation method. Robotics and Computer Integrated Manufacturing, 23, 503–16.

Ribas, I., Leinsten, R., Framiñan, M. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. Computers & Operations Research. 2010; 37: 1439–1454.

Xuan, H., Tang, L. Scheduling a hybrid flow shop with batch production at the last stage. Computers and Operations Research. 2007; 34:2718–2733.

4:2718–2733.