



HAL
open science

A semi-online algorithm for optimizing the pre-disinfection duration of medical devices in a hospital sterilization service

Onur Ozturk, Maria Di Mascolo, Marie-Laure Espinouse, Alexia Guoin

► To cite this version:

Onur Ozturk, Maria Di Mascolo, Marie-Laure Espinouse, Alexia Guoin. A semi-online algorithm for optimizing the pre-disinfection duration of medical devices in a hospital sterilization service. MOSIM 2012 - 9th International Conference of Modeling, Optimization and Simulation, Jun 2012, Bordeaux, France. 10 p. hal-00730531

HAL Id: hal-00730531

<https://hal.science/hal-00730531>

Submitted on 10 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A SEMI-ONLINE ALGORITHM FOR OPTIMIZING THE PRE-DISINFECTION DURATION OF MEDICAL DEVICES IN A HOSPITAL STERILIZATION SERVICE

Onur OZTURK, Maria DI MASCOLO, Marie-
Laure ESPINOUSE

Grenoble-INP / UJF-Grenoble 1 / CNRS,
G-SCOP UMR5272 Grenoble, F-38031,
France
name.surname@g-scop.grenoble-inp.fr,

Alexia GOUIN

Grenoble-INP / UJF-Grenoble 1 / CNRS,
GIPSA-lab UMR 5216 St. Martin
D'Hères, F-38402, France
Alexia.Gouin@gipsa-lab.grenoble-inp.fr

ABSTRACT: After utilization in operating theaters, medical devices (MDs) are sent to the sterilization service. The sterilization process is made up of various steps. After pre-disinfection, different sets of MDs used for different surgical operations may be washed together, without exceeding the washer capacity in the washing step. Ideal pre-disinfection time is approximately 20 minutes, since longer pre-disinfection times may corrode MDs. Our aim is thus to minimize the pre-disinfection time exceeding the ideal pre-disinfection time. Hence, the decisions for batching the MD sets and launching washing cycles are crucial in minimizing the waiting time of MDs in the washing step. In this paper, we model the washing step as a batch scheduling problem, where MD sets are denoted as jobs with different sizes and different release dates, but with equal processing times for washing. Although MD arriving times to the sterilization service may be estimated in advance regarding the operating room scheduling, it is not always possible to have the exact information about their arrival within a day. Thus, we develop a semi-online algorithm for the loading of washers. After testing its efficiency for the pre-disinfection criterion, we develop a simulation model in order to test the impact of this optimization on the whole sterilization process.

KEYWORDS: Hospital sterilization service, batch scheduling, semi-online algorithm, simulation.

1 INTRODUCTION

Hospital sterilization services aim at minimizing all infectious risks due to the reutilization of medical devices in surgical operations. A medical device is an instrument, apparatus, appliance, or any other article, which is used for medical purposes on patients, in diagnosis, therapy or surgery. We are interested in medical devices which are used in surgeries, and which have to be sterilized after utilization.

All MDs provided for a surgical operation must be sterilized. It is more appropriate to refer to these instruments as reusable medical devices (or RMDs) as they are re-used after sterilization. The sterilization process is regulated by some quality standards (see Standard AFNOR for French quality standards concerning RMD sterilization). Sterilization is a cyclic process (Fig.1) consisting of several steps. After utilization in operating theaters, RMDs are sent to the sterilization service where they pass through the following steps: pre-disinfection, rinsing and washing, verification, packing, sterilization, storage and reutilization in operating theaters.

After utilization for a surgical operation, RMDs are directly placed in a substance, allowing pre-disinfection, and are transferred to the sterilization service. There, they are firstly rinsed and washed in automatic washers.

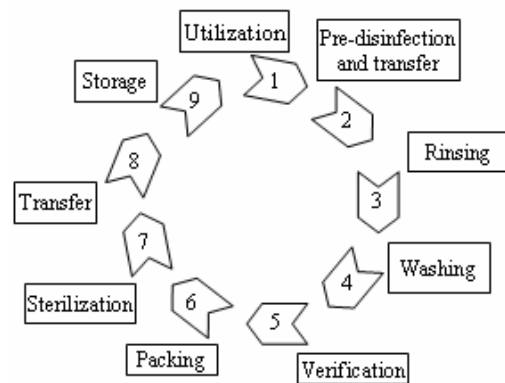


Figure 1: Sterilization cycle

Rinsing is performed either manually or automatically in automatic washers. After washing, RMDs are verified and packed into appropriate boxes. All items must be packed individually or grouped into boxes prior to sterilization. They are then sterilized in so-called “autoclaves”, transferred to operating theaters and stored before reutilization.

In the sterilization services we investigated, manual rinsing is the norm, while automatic washers always rinse prior to washing. One reason for this double rinsing is that it allows RMDs to wait to be washed without any risk of corrosion due to the pre-disinfection liquid. As

the washing step is usually a bottleneck of the sterilization process, the RMDs may have to wait a considerable time before being washed (e.g. more than 30 minutes). Consequently, rinsing the RMDs manually as they arrive removes any pre-disinfection liquid they may contain, thus meaning they can wait to be washed without risk of corrosion. Note that the corrosion effect of pre-disinfection can shorten the life of RMDs. In our investigations, we saw that the managers of sterilization services define an ideal pre-disinfection time which is equal to 20 minutes, while a minimum of 15 minutes is imperative and 50 minutes is often considered as the upper limit. Sterilization service managers consider that RMDs are subject to corrosion from the beginning to the end of pre-disinfection. While how quickly RMDs corrode depends on the pre-disinfection substance, it is clear that the pre-disinfection liquid becomes more penetrative over time.

In fact, if RMD waiting time before automatic washing is sufficiently short, ideal pre-disinfection time can be ensured only with automatic rinsing. Consequently, manual rinsing operators could be transferred to other workstations (for example to the packing station, which is always manual). Considering that the ideal pre-disinfection time is 20 minutes, we define the “pre-disinfection excess time” of an RMD as the difference between its pre-disinfection time and the ideal pre-disinfection time. Note that RMD pre-disinfection excess time equals zero if it is less than (or equal to) the ideal time.

Our main aim in this study is thus to minimize the mean pre-disinfection excess time of RMD sets during the washing step, to ensure the best possible RMD pre-disinfection times. One aim of this goal is to consider the advisability of removing manual rinsing.

The remainder of this paper is organized as follows. In section 2, we describe the problem of loading automatic washers, and show how this problem can be treated as a batch scheduling problem. In section 3, we provide a literature review on batch scheduling problems. In section 4, a semi-online algorithm is presented for our batch scheduling problem. Section 5 is dedicated to computational tests. Our aim in that section is also to evaluate the impact of our optimization method on a complete sterilization service.

2 PROBLEM DESCRIPTION

In a typical hospital, several surgical operations may be performed in the course of the day. All RMDs used in a surgical operation constitute the RMD set for this operation. As you may suppose, there can be a large number of different RMD sets in a hospital. Moreover, the number of different types of RMDs is generally very great, and for a typical hospital, there may be hundreds of RMD references. Because each surgical operation may require different numbers and types of RMDs, RMD sets

may be of different sizes. For different reasons (surgery start times and durations, pre-disinfection procedure, etc.), RMD set arrival times to the sterilization service are different within the same day. Even though operating room scheduling may be used to estimate in advance the arrival times of RMD sets, there may be many elements influencing their arrival (ex: duration of surgeries, disrespect to surgery beginnings, human effect for manual step like pre-disinfection or transport of RMDs). Thus, it is not always possible to estimate the arrival time of RMD sets with exactitude of 100%. We assume in our work that the arrival of RMD sets is known, although, some (or most) of the RMD sets will be arriving before or later than the estimated arriving times. Our solution approach is thus a semi-online algorithm, which is going to make different decisions according to the violation of the estimated RMD arriving times.

Concerning the automatic washers, they can be described as identical batching machines. Moreover, washing durations (including automatic rinsing) are also identical for all RMD sets in any automatic washer. It is possible to put more than one RMD set into a washer, as long as its capacity is not reached. The decisions to take are then: which RMD sets should be placed together in order to constitute a batch for washing, and when to launch a washing cycle. Note that in the washing step, RMD sets are not usually allowed to be split among several washers due to organizational and traceability reasons. If split, it then takes a long time to reassemble the boxes of the RMD sets identically in the subsequent steps. Besides, splitting may cause some mistakes in set reassembling.

According to a questionnaire held among several hospitals of Rhône-Alpes region in France (EESS, 2007), pre-disinfection duration is an important criterion for the performance of sterilization services. In our scheduling problem, we consider there is no manual rinsing and RMD are ready for washing once they arrive to the sterilization service. In case good pre-disinfection durations are guaranteed with our semi-online algorithm, there is going to be no need for a manual rinsing.

2.1 Identification with a batch scheduling problem

In our scheduling problem, RMD sets are denoted as jobs and automatic washers as parallel batching machines. We make the following assumptions:

- There are N jobs to be processed. The release date and the size of a given job j are denoted by r_j and w_j , respectively. The pre-disinfection starting time of job j is denoted by t_j . Since washing times are the same for all RMD sets, job processing times are the same for all jobs and are denoted by p .
- All machines have the same capacity B , and the size of a job cannot be greater than machine capacity.
- Several jobs can be batched together, complying with the machine capacity constraint.

- Once processing for a batch is started, it cannot be interrupted (i.e. preemption is not allowed)
- We are not allowed to split a job into several batches.

Inspired by Graham's notation [Graham *et al.*, 1979], we propose the following notation for our problem: $P / p\text{-batch}, r_j, p_j = p, w_j, B / (1/N)*\sum f_j$. In this notation, P stands for identical parallel machines, $p\text{-batch}$ for parallel batching; r_j and w_j denote job release dates and sizes, respectively, $p_j = p$ stands for equal processing times, and B for machine capacity. Finally, $(1/N)*\sum f_j$ refers to the objective function. This function penalizes excessive waiting times before washing. More precisely, pre-disinfection times are penalized for every minute exceeding 20 minutes. Thus, the formula giving f_j is the following: "washing starting time for job j - pre-disinfection starting time for job j - 20 minutes". Negative values of f_j will refer to 0.

3 LITERATURE REVIEW

The batch scheduling literature is really vast. Thus, we focus here on batch scheduling literature considering different job sizes. We start by articles studying offline batch scheduling (i.e. all data is known in advance).

In parallel batching problems with different job sizes, the sum of job sizes that are put into a batch should not exceed machines capacity. Each job is assigned to just one batch. The processing time of a batch is given by the longest processing time of jobs that are put into that batch. In table 1, we give a brief classification of the literature dealing with parallel batch scheduling problems considering different job sizes. According to release dates and number of machines, we divide the batch scheduling literature into 4 groups: first group considers equal release dates and single machine, second group considers equal release dates and parallel machines, third group considers job release dates and single machine, finally, fourth group represents jobs with release dates and parallel machines.

The first column of table 1 represents the group number. References are cited in the second column. The third column shows the solution method proposed by articles. Finally, last column shows the objective function. Note that in all these articles, different job processing times are considered. Clearly, the articles cited in the fourth group are more interesting for us, since we consider different release dates and parallel machines. Hence, we focus on the fourth group and give some more details for these articles below.

All articles in the fourth group consider makespan minimization. Chung *et al.* (2009) are the first ones who consider the problem with different job sizes, release dates and processing times taking into account parallel machines. They develop first a MILP model which is capable of solving instances containing 10 jobs in a reasonable amount of time.

Objectives: Makespan: 1; total completion time: 2; total weighted comp. time: 3

Group	Ref.	Algorithm	Obj
1	Uzsoy (1994)	Heuristics, B&B	1, 2
	Jolai Ghazvini and Dupont (1998)	Heuristics	2
	Dupont and Jolai Ghazvini (1998)	Heuristics	1
	Kempf <i>et al.</i> (1998)	MILP, heuristics	1, 2
	Azizoglu and Webster (2001)	B&B	3
	Zhang <i>et al.</i> (2001)	Approx. algo.	1
	Dupont and Dhakens-Flipo (2002)	B&B	1
	Kashan <i>et al.</i> (2006)	Genetic algorithms	1
	Zhang <i>et al.</i> (2007)	Approximation algorithms	2
	Kashan <i>et al.</i> (2009)	Approximation algorithms	1
	Parsa <i>et al.</i> (2010)	Branch and price algo.	1
	Kashan <i>et al.</i> (2010)	Genetic algorithms	1
2	Chang <i>et al.</i> (2004)	Simulated annealing	1
	Kashan <i>et al.</i> (2008)	Genetic algorithm	1
3	Li <i>et al.</i> (2005)	Approximation algorithm	1
	Nong <i>et al.</i> (2008)	Approximation algorithm	1
	Lu <i>et al.</i> (2010)	Approximation algorithm	1
4	Chung <i>et al.</i> (2009)	MILP, heuristics	1
	Damodaran <i>et al.</i> (2011)	meta-heuristic	1
	Damodaran and Velez Gallego (2010)	heuristics	1
	Chen <i>et al.</i> (2010)	Genetic algo, ant colony heuristic	1
	Wang and Chou (2010)	Genetic algo, simulated annealing	1

Table 1: A review for batch scheduling problems with different job sizes

However, with the increasing number of jobs, the performance of the MILP model decreases and thus, authors develop also two heuristic approaches in order to find quick and efficient solutions. In both of these heuristics, first batches are formed, and then they are affected to machines. The batch creation procedure is common in both heuristics and is inspired from the DELAY algorithm proposed by Lee and Uzsoy (1999). It uses two parameters: a , for determining the time window in which jobs are batched, b , for determining the fullness of batches. They test their heuristic with different combinations of a and b . Because of these two parameters, the heuristic of Chung *et al.* (2009) is only pseudo-polynomial. Damodaran and Velez Gallego (2010) develop also a heuristic approach. This heuristic operates

by, first, finding jobs to be executed in the same batch by solving a 0-1 knapsack problem. After forming all batches, they are affected to machines using a heuristic approach. They report that their heuristic outperforms the ones given by Chung *et al.* (2009). Even though that heuristic finds a solution in a small amount of time (less than 10 seconds according to numerical tests of Damodaran and Velez Gallego (2010)), it is also pseudo-polynomial since it uses a pseudo-polynomial dynamic programming algorithm for the resolution of the knapsack problem. Damodaran *et al.* (2011) develop a meta-heuristic called Greedy Randomized Adaptive Search (GRASP). They report that the GRASP approach guarantees the optimal solution for small instances and is more effective than the heuristics proposed by Chung *et al.* (2009). Chen *et al.* (2010) develop a genetic algorithm and an ant colony optimization. For the batch assignment procedure, they propose a heuristic (ERT-LPT: earliest ready time-longest processing time) which is used in both meta-heuristics. For computational experiments, they develop another heuristic considering the batch creation procedure proposed by Dupont and Jolai Ghazvini (1998) where ERT-LPT is applied afterwards. Their results indicate that both meta-heuristics outperform the heuristic approach. Wang and Chou (2010) consider machines with different capacities. They develop a genetic and a simulated annealing algorithm, and test their algorithms on the instances defined by Chung *et al.* (2009). It is reported that the proposed meta-heuristics are more efficient than the heuristics of Chung *et al.* (2009).

All articles cited above suppose that all data, i.e. job sizes, release dates, processing times and number of machines, is known in advance. Hence, those are offline approaches for the scheduling problem. According to our knowledge, there is only one work considering an online scheduling problem with different job sizes, release dates and processing times (Yongqiang and Enyu, 2005). They work on makespan minimization in presence of a single machine and give an online algorithm with an asymptotic competitive ratio of 22/9.

In our work, we have partial information about job arrival times. While some jobs arrivals are as expected, some other job arrivals may be before or after the expected arrival time. Thus, our study is a semi-online approach since, some information (but not all) is known in advance. Concerning the objective function, it is an extension of the *total completion time*. If we consider that all pre-disinfection starting times and arrival times are equal, then our objective function is reduced to total completion time.

Note that the problem we treat is strongly NP-hard. For the offline version of our problem, we developed a MILP model and a heuristic, called *TIH*, whose performance is quite good for the pre-disinfection time optimization (Ozturk *et al.*, 2011). We integrate here that heuristic in a semi-online optimization model for the pre-disinfection

time optimization. Below, we first remind the offline heuristic, and then explain how we derive a semi-online heuristic from that first one. Then, we test the efficiency of the semi-online heuristic on instances inspired from a real case. In the last section, we present a simulation model using ARENA. Our aim is to evaluate if the optimization of pre-disinfection times with the semi-online algorithm has a good impact on the entire sterilization service.

4 SOLUTION APPROACH

Before giving the semi-online algorithm, let us first remind the offline algorithm: this heuristic operates by first deciding a time window $[0, t]$ after which, within each time window, a knapsack problem is solved where all jobs have the same weight (or importance) but possibly different sizes. For resolution of the knapsack problem, we use a procedure derived from the “first fit heuristic”, which is one of the classical bin-packing algorithms. However, we stop the first fit resolution procedure when only one batch is created. This way, a single batch is created, and assigned to the first available machine. The procedure for creation of a batch in the given time horizon is as follows:

first fit procedure (FFP)

- 1- Sort jobs in non-decreasing order of release dates in a list L
- 2- Open a batch
- 3- Starting from the first element, run through the list L : if the job fits the batch, put it in the batch, else, continue with the next element of L

Concerning determination of the time window, its length, t , is defined by $\max(r'_k)$; first machine availability among all machines) where r'_k is the k^{th} earliest job arrival time out of the unassigned jobs. Parameter k varies from 1 to N , i.e. the number of jobs. The start of the time window is 0. Thus, the time window is $[0, t]$ where t changes during execution of the heuristic. t is then the upper limit of the time window and also an instant in the problem. The time complexity of the algorithm is $O(N^3 \log N)$. For more information, we refer the reader to Ozturk *et al.* (2011).

4.1 Semi-online algorithm: *TIH*_{CORRECTIVE}

We develop a semi-online version of heuristic *TIH*. This heuristic supposes that all data is known in advance. With these data, a first schedule is calculated by executing heuristic *TIH*. Then, each time a job arrives to the sterilization service, it checks if the arrival of that job is on time. If it is the case, the first calculated schedule stays valid. Otherwise, i.e. if a job arrives in advance or later than the expected arrival time, the data supposing that arrival times are known is updated. More clearly, if a job arrives earlier than expected, the arrival time of that job is updated, and then *TIH* is re-executed in order to get a new schedule for washing.

Time Interval Heuristic (TIH)

- 1 Sort jobs in non-decreasing order of arrival times r^j : L_0 , and set $L_l = L_0$
- 2 Set the initial solution, sol_{in} , for the mean pre-disinfection excess time, equal to a large number. For k from 1 to N , set $l = k$
 - 3.1 While L_l is not empty,
 - 3.1.1 If the number of elements in L_l is smaller than l , set $l =$ number of elements in L_l
 - 3.1.2 Define the length t of the time window as $t = \max$ (arrival time of the l^{th} element of L_l ; first machine availability out of all machines)
 - 3.1.3 Apply *FFP* on jobs whose arrival times are shorter than t and erase batched jobs from L_l
 - 3.1.4 Among the batched jobs, find the job whose pre disinfection start time is the latest: pre_{max} . Set $t' = \max (pre_{max} + 15$; first machine availability out of all machines; greatest job release date in the batch)
 - 3.1.5 Once t' is reached, launch a washing cycle with the batch formed by *FFP*, and calculate the new availability of the machine on which the batch is processed
 - End while
 - 3.2 Let sol_{mpt} be the obtained mean pre-disinfection excess time
 - 3.3 If $sol_{in} > sol_{mpt}$, set $sol_{in} = sol_{mpt}$
 - 3.4 Set $L_l = L_0$
- End for
- 4 Set the final solution, sol_{final} , equal to sol_{in} .

TIH_{CORRECTIVE}

- 1 Execute *TIH* on data contained by $L_{PROVISION}$ in order to get a first pre-assignment of jobs to batches. For each batch b , set $start_b$ as the expected processing start time.
- 2 When a job arrives,
 - 2.1 If that job is earlier than its expected arrival, update its arrival time in $L_{PROVISION}$ and re-execute *TIH* with the new updated data of $L_{PROVISION}$ in order to determine new batches.
- 3 When a job is supposed to have arrived,
 - 3.1 If it has not arrived yet, update its arrival so that it arrives with the succeeding job. Re-execute *TIH* in order to determine new batches.
- 4 Each time a new $start_b$ is reached, re-execute *FFP* in order to create batch b and assign it to the first available machine. Update the new availability of the machine.

If a job has still not arrived although its arrival time has passed, it is supposed that the late job will arrive with the next job. Thus, the arrival time of the late job is updated in this way, and *TIH* is re-executed to get a new schedule. For the first execution of *TIH*, we have got a list, $L_{PROVISION}$, containing the expected arrival times. Each time a job is early or late, its arrival time is updated in that list. The different steps of the algorithm are given in the next column:

In fact in *TIH_{CORRECTIVE}*, *TIH* is used to make a pre-assignment of jobs to batches. Because after each execution of a batch on a machine, *TIH* knows which machine is available in which instant. Each time a starting time is reached for a batch; *FFP* determines the jobs to be executed in the same batch.

4.2 Implementation of TIH_{CORRECTIVE}

We designed a simulation model for the implementation of *TIH_{CORRECTIVE}*. The model is built on ARENA. At first step, the model represents only the washing step of a sterilization service, in order to test the behavior of *TIH_{CORRECTIVE}* for pre-disinfection times. Let us briefly talk about the simulation model.

The washing step is composed of a washing stock and of washers, which are easily represented using appropriate modules of ARENA. For each RMD set/job, an entity is created for the representation of job arrival. The creation of an entity means that a job enters to the washing stock. Thus, we have a first type of data for real arrival times for RMD sets. However, *TIH_{CORRECTIVE}* works with a second kind of data where expected arrival times are kept. In the simulation model, the main aim of this second kind of data is to see if a job is late (In case a job is not late but early, it will enter the washing stock in the simulation model before its expected arrival time. Thus, there is no more need to keep an expected arrival time for that job. But if it is late, the model needs to have the information about the expected arrival time since a job may be late several times after each update of its expected arrival time.). Thus, in the simulation model, we have an artificial clock that checks the expected arrival times. According to expected job arrival, that clock tells *TIH_{CORRECTIVE}* if an entity should have arrived, or not, to the washing stock. Then, *TIH_{CORRECTIVE}* checks the washing stock and updates the expected arrival time of the late job. All these control operations for the execution of *TIH_{CORRECTIVE}* require an information flow which we managed with the VBA module of ARENA. Thus, all steps of the algorithm are coded in VBA which is responsible for operations like updating arrival times, deciding the jobs to be batched together and processing of batches on machines. Below, we show a small figure representing the washing step.

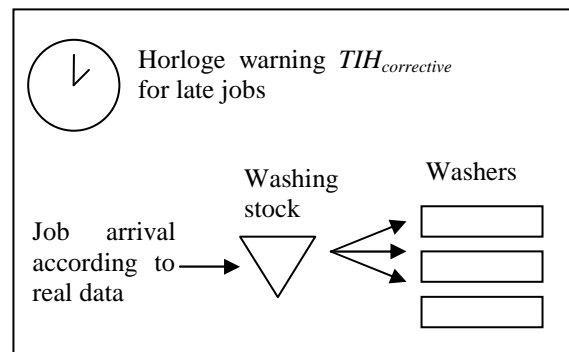


Figure 2: Functioning of the washing step

5 COMPUTATIONAL RESULTS

In this section first, we test the performance of $TIH_{CORRECTIVE}$ for pre-disinfection times. Afterwards, we complete the simulation model with the rest of the sterilization steps (verification, packing, sterilization) in order to see the impact of $TIH_{CORRECTIVE}$ on the whole system. But first, let us explain the test instances.

5.1.1 Test instances

The test instances are inspired from real data given by a French private hospital. There are 4 automatic washers in the sterilization service. Washer capacities are the same and equal to 6 DIN (DIN is a standard measurement type for the volume of automatic washers), and washing time (i.e. batch processing time) is 60 minutes. RMD set sizes are multiples of 1/36 of machine capacity. Thus, we estimate job sizes using a uniform distribution: $U[1,36] * \text{machine capacity} / 36$. We observed that inter-arrival times between two RMD sets may take any value between 0 and 40 minutes. Therefore, we sampled job arrival times from a uniform distribution such that two consecutive arriving jobs may have an inter-arrival time equal to X minutes, where $X \sim U[0; 40]$. We denote this type of arrival as a “random RMD arrival”. However, in some other sterilization services, regular collection of RMD sets may take place in operating theaters. In this case, someone is in charge of collecting RMD sets from operating theaters at fixed intervals all day long, thus, RMD sets arrive at the sterilization service regularly. We consider 2 different values for the regular inter-arrival times: 20 minutes and 40 minutes, and assume that the number of jobs released in a collecting tour is sampled from a uniform distribution which is $U[0;2]$ for 20 minutes of regular collecting and $U[1;3]$ for 40 minutes of regular collecting. We thus define 3 instance types, according to RMD set arrivals. Let us refer to them as 1st, 2nd and 3rd instance types for irregular arrivals, 20 minutes of regular collecting, and 40 minutes of regular collecting, respectively. For the start of pre-disinfection times, it is observed that RMD sets arrive at the sterilization service at least 5 minutes and at most 30 minutes after the beginning of their pre-disinfection. Any values between 5 and 30 minutes were equally observed. Hence, the pre-disinfection start time of a job is defined as the difference between its arrival at the sterilization service and “the transfer time (td)” where td follows a uniform distribution $U[5; 30]$. Again inspired from the real data, instances contain 50 jobs.

We use the instructions above to create expected arrival times of RMD sets. In order to create the real arrival times of RMD sets, we make some modifications on job arrival times. For the number of jobs arriving early/late, we suppose three configurations: 1- 10% of jobs may be early/late, 2-50% of jobs may be early/late, 3-100% of jobs may be early/late. Thus, for the second configuration for example, 25 jobs over 50 may arrive earlier or later than the expected arrival time. In order to determine

how earlier/late those jobs are, we have 4 cases: 1- a job can be 0 to 15 minutes early/late, 2- a job can be 0 to 30 minutes early/late, 3- a job can be 0 to 60 minutes early/late, 4- a job can be 0 to 30 minutes late only. In order to calculate real arrival times, we added/subtracted $U[0, max]$ minutes to/from expected arrival times where $max = 15, 30$ or 60 minutes according to the 4 cases above. Note that for each instance type, configuration and case, 10 instances are created containing 50 jobs.

5.1.2 Performance of $TIH_{CORRECTIVE}$ pre-disinfection times

We compare $TIH_{CORRECTIVE}$ to a natural strategy for the loading of washers, which is $FIFO_{online}$, and also to the lower bound value. The lower bound (LB) is obtained by a simple calculation. We suppose that each job is processed on a machine as soon as its arrival without waiting in the washing stock. This way, we get the best pre-disinfection time for each job. Regarding $FIFO_{online}$, that strategy supposes that no information is known in advance about future arrivals of RMD sets. All information about an RMD set is known as soon as it arrives to the sterilization service. The weak point of this strategy is the loading of washers. Batches are created with successive RMS sets. When there is an RMD set that does not enter a batch, batch is closed and assigned to the first available washer. Thus, $FIFO_{online}$ tries to maximize the utilization of capacity usage of washers. However, that causes long RMD set waiting before washing, which increases long pre-disinfection times. In the table below, we compare $FIFO_{online}$ to the LB and $TIH_{CORRECTIVE}$, for the pre-disinfection criterion. For each type of instance, configuration and case, tables 2, 3 and 4 show the maximum and average pre-disinfection excess times. Let us remind that a job has an excess for the pre-disinfection if it is pre-disinfected more than 20 minutes. Pre-disinfection times smaller than 20 minutes are ideal, thus have no penalization for the objective function.

Inst type.	Case	LB	TIH_{COR}		$FIFO_{online}$	
			max	moy.	max	moy.
1	1	0,74	10	1.9	99	23.1
	2	0,8	22	2.3	119	19.2
	3	0,75	24	1.98	82	12
	4	1	40	4	97	11.4
2	1	1,2	14	4.9	75	21.4
	2	1,1	34	5.3	100	18.2
	3	0,95	30	5.1	102	18.1
	4	1,21	22	5.3	59	19.2
3	1	2,8	33	14.9	94	28.1
	2	3,4	34	15.5	106	25.3
	3	3,1	33	16.7	63	32.2
	4	4,2	34	16.5	106	35

Table 2: Performance of $TIH_{CORRECTIVE}$ and $FIFO_{online}$ for pre-disinfection excess times for configuration 1

Inst type.	Case	LB	$TIH_{COR.}$		$FIFO_{online}$	
			max	moy.	max	moy.
1	1	0,6	38	3.5	59	10.9
	2	0,5	32	2.8	69	10.9
	3	0,7	26	2.1	91	15.2
	4	0,6	31	1.2	91	14.5
2	1	1,1	14	4.3	59	17.1
	2	1,2	14	4.3	58	19
	3	2,1	31	6.4	92	20.9
	4	1,9	14	5.9	101	23.1
3	1	4,1	32	13.3	103	29.8
	2	3,5	32	16.2	130	25.6
	3	3,8	33	16.6	80	27.5
	4	4	34	15.1	84	33.7

Table 3: Performance of $TIH_{COR.RECTIVE}$ and $FIFO_{online}$ for pre-disinfection excess times for configuration 2

Inst type.	Case	LB	$TIH_{COR.}$		$FIFO_{online}$	
			max	moy.	max	moy.
1	1	0,4	17	3.2	91	11.8
	2	0,4	22	1.8	96	18.8
	3	0,5	27	1.4	59	13.6
	4	0,6	30	2.2	61	11.2
2	1	1,4	20	5.1	78	18.1
	2	1,8	25	4.1	78	21.5
	3	1,3	28	5.8	78	20.2
	4	2,3	24	5.7	77	19
3	1	4,1	34	12.7	79	25.1
	2	4,1	34	12.9	119	31.4
	3	3,9	32	13.1	72	24.6
	4	4	34	13.5	117	33.1

Table 4: Performance of $TIH_{COR.RECTIVE}$ and $FIFO_{online}$ for pre-disinfection excess times for configuration 3

According to numerical tests, $TIH_{CORRECTIVE}$ gives quite good results for pre-disinfection times in an operational point of view. Considering that 20 minutes is the ideal pre-disinfection duration, the mean pre-disinfection times given by $TIH_{CORRECTIVE}$ is between 20-25 minutes for 1st and 2nd types of instances, independently from different configurations and cases. Because the arriving of jobs takes longer in the 3rd type of instance compared to 1st and 2nd types, pre-disinfection times are generally longer than 30 minutes. But again compared to $FIFO_{online}$, $TIH_{CORRECTIVE}$ is more powerful. Another consequence we get from the test results is that application of a $FIFO$ strategy gives rise to excessive job waiting in the washing stock. Hence, pre-disinfection times increase, depending on the waiting at the washing stock.

Until now, we only considered that a job may arrive later or earlier than expected. Thus, all other information, like job sizes and pre-disinfection beginning times were

supposed to be perfectly known in advance. For that case, we have just seen that $TIH_{COR.RECTIVE}$ gives quite similar results for pre-disinfection times almost for each different configuration and different case. However, what would happen if information on job sizes and pre-disinfection beginning times were not sure either? In other words, a job may have a different size and pre-disinfection beginning time than the expected ones. In order to see the performance of $TIH_{CORRECTIVE}$ for that new case, we create 20 instances for each of the 1st, 2nd and 3rd instance types, following the instructions given in the first paragraph of this section. We suppose, for each type, that the first 10 instances are the expected data and the second 10 instances are the real data. This way we have a group of instances representing real and expected data while all (or almost all) data are different on job sizes, arrival times and pre-disinfection beginnings. Thus, the responsibility of $TIH_{CORRECTIVE}$ is changed a little bit. Each time a new job arrives, it checks not only the exactitude of the arrival time, but also if the expected data on the job size and pre-disinfection beginning is the same as the real one. If not, it updates this information with the real data and TIH is re-executed. We give in the table below the performance of $TIH_{CORRECTIVE}$ for the pre-disinfection times where no expected information coincides with the real one (let us call this case configuration 4).

Type Inst.	min	max	moy.
1	0	43	5.6
2	0	55	6.6
3	0	60	12.2

Table 5: Performance of $TIH_{CORRECTIVE}$ for pre-disinfection times for configuration 4

According to numerical results given in the table above, performance of $TIH_{CORRECTIVE}$ for pre-disinfection times is not much influenced even though there is a big difference between real and expected data. Table 3 shows that good pre-disinfection times are obtained with $TIH_{CORRECTIVE}$ even when the real data is completely different than the expected data. Because our main objective is satisfied with $TIH_{CORRECTIVE}$, our second aim is now to see the impact of this optimization on the whole service.

5.1.3 Impact of $TIH_{CORRECTIVE}$ on the performance of a sterilization service

We have seen that good pre-disinfection times are obtained with $TIH_{CORRECTIVE}$. The main result of this observation is that there is no more need for a manual rinsing. RMD waiting is low enough at the washing step, and hence, after a small waiting, they are directly loaded into washers. In anyway, washing cycles start with an automatic rinsing. Thus, manual rinsing is no more necessary and so the application of $TIH_{CORRECTIVE}$ lets us remove the manual rinsing operator to other steps of the sterilization service.

In order to test the impact of $TIH_{CORRECTIVE}$ on the whole service, we completed the simulation model created in the previous section 4. We added the other steps which are verification, packing and sterilization. In the sterilization service we investigated, there are 4 packing posts (verification is included in this step). After packing, 3 autoclaves are present for the sterilization. Because we do not need a manual rinsing anymore, our proposition is to remove this step and to add a fifth working post to the packing step (according to EESS (2007), packing step can be a bottleneck for some sterilization services).

Once RMD sets are washed, they are transferred to verification and packing posts. Verification is fast and is considered as part of packing. After verification in packing step, all RMD belonging to an RMD set are placed into bags and boxes. The number of bags and boxes in an RMD set is independent from the size of that RMD set. While an RMD set of 6 DIN contain 2 big boxes only, another RMD set of size 4.2 DIN may contain 3 small boxes and 2 bags¹. According to the observed data, the number of bags in an RMD set vary uniformly between [0, 6] while the number of boxes vary uniformly between [0, 5].

In the packing step, the composition of bags takes approximately 0.7-0.8 minutes. The composition of boxes takes approximately 20-21 minutes. After packing RMD into boxes and bags, these boxes and bags become an individual entity in our simulation model. The sizes of boxes and bags are calculated respecting the original size of the RMD set in which they are contained till the end of washing. The size of bags is fixed to 0.8 DIN while box sizes are variable. In order to estimate the size of boxes in an RMD set, the total size of bags is subtracted from the size of the RMD set. Then, it is divided to the number of boxes in order to assign a size to each box. Box and bag sizes are important for launching sterilization cycles in autoclaves. As said previously, we have 3 autoclaves. Their sizes are equal to 12 DIN. The strategy for the launching of these machines is that the used capacity should be at least 80%. This way the desired pressure is obtained, steam has a better contact with RMD and thus the hygiene level is ensured.

The simulation is performed for instances tested in section 5.1.2. The average run time of an instance is about 5 seconds. We start by analyzing the waiting times in different stocks. These stocks are at washing, packing and sterilization steps. Because the performance of $TIH_{CORRECTIVE}$ depends mainly on different instance types (i.e. 1st, 2nd and 3rd instances types, cf. section 5.2), we analyze our results according to these instance types. Let us start by waiting times in the washing stock. We show

by A, B, and C the results corresponding to 1st, 2nd and 3rd instances types.

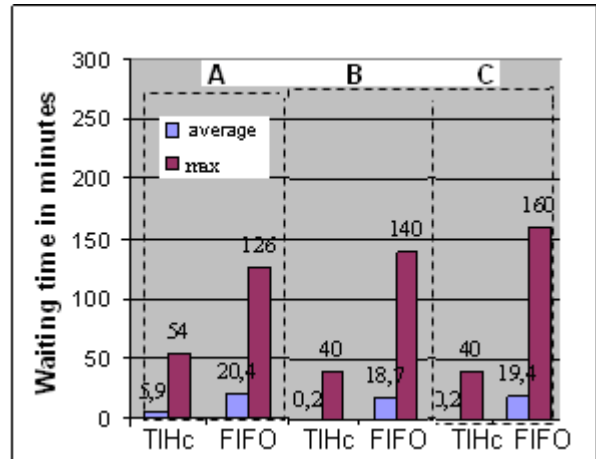


Figure 3: Maximum and mean waiting times in the washing stock

According to figure 3, $TIH_{CORRECTIVE}$ is able to minimize the waiting of RMD in the washing step. We see also that waiting times are smaller for 2nd and 3rd types of instances. This observation is due to arrival procedure of RMD sets. Because in these instances RMD sets arrive in big quantities (not one by one as for 1st instance type), it is possible to benefit better from the washer capacity.

Because packing of bags takes little time, waiting of entities in the packing stock for bags is small both for $TIH_{CORRECTIVE}$ and $FIFO_{online}$. However, it is the right opposite for the waiting times in the stock for packing of boxes. Figure 4 gives us mean and maximum waiting times at this stock, again for 3 instance types showed with A, B and C.

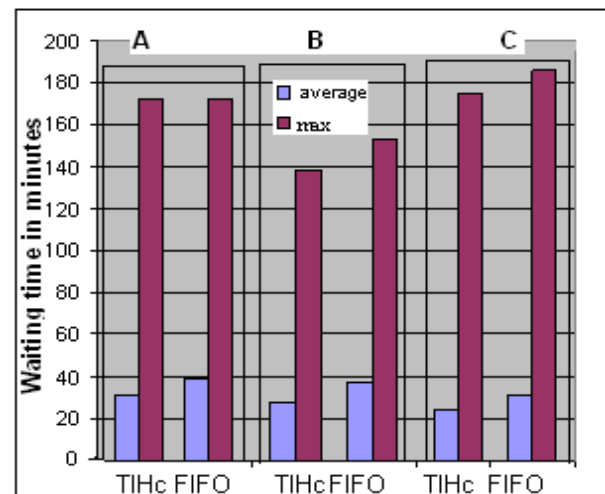


Figure 4: Maximum and mean waiting times in the packing stock for boxes

¹ Note that RMD sets are not standard. They are composed of boxes and bags according to needs of surgeons. While a surgeon may need little RMD, another surgeon may need many RMD for the same surgery.

The difference between $TIH_{CORRECTIVE}$ and $FIFO_{online}$ is less significant at the packing stock of boxes compared to waiting times in the washing stock. This is because packing of boxes takes a long time. But again, the differ-

ence between these two strategies is 10 to 15 minutes which results to an average minimization of 20-25 minutes for the time passed in the system considered with the gain at the washing stock.

RMD waiting at the sterilization stock is the same both for $TIH_{CORRECTIVE}$ and $FIFO_{online}$. The average RMD waiting is about 30 minutes.

According to results given above, if $TIH_{CORRECTIVE}$ is able to minimize a little bit the time passed in the sterilization service, then, it should also be able to increase a certain number of RMD sterilized per day. For that purpose, we tested for each instance type, the average number of boxes and bags prepared and sterilized at the end of the day. In table 4, we show these quantities for different instance types.

	Inst. type 1	Inst. type 2	Inst. Type 3
$TIH_{cor.}$	186	178	162
$FIFO_{on.}$	173	169	153

Table 6: Average box and bag numbers sterilized at the end of a day

According to the results given in table 4 for the criterion of the number of boxes and bags sterilized, $TIH_{CORRECTIVE}$ is again better than $FIFO_{online}$.

6 CONCLUSION

In this study, we modeled the washing step of a sterilization service as a batch scheduling problem. Our aim is to minimize the mean pre-disinfection excess time of RMD sets at the washing step, when there is no manual rinsing in the system.

For the objective of optimizing the mean pre-disinfection time of RMD, we developed a MILP model and a heuristic in previous works. These methods supposed that all information about RMD set arrivals, sizes and pre-disinfection beginnings were known in advance. However, in most of the hospitals, there is little information exchange between operating blocs and the sterilization services. Thus, it is not possible to know all information about RMD arrivals in advance. Considering that case, we modified the previously developed heuristic in order to have a semi-online version. We called it $TIH_{CORRECTIVE}$.

$TIH_{CORRECTIVE}$ supposes that all data about RMD arrivals are known in advance. It prepares a first schedule for the washing of RMD sets. If there is an RMD set which is late or which arrives earlier to the sterilization service, it updated the information of that RMD set, and then prepares a new schedule for RMD washing.

Considering that the ideal pre-disinfection time is 20 minutes, 20 to 30 minutes of pre-disinfection is consid-

ered good by managers of sterilization services. In order to test the performance of $TIH_{CORRECTIVE}$ on real life inspired instances, we inserted it in a simulation model built in ARENA. Because the MILP model (proposed in a previous work) is efficient on small instances (ex: containing 10 to 15 jobs), we compared $TIH_{CORRECTIVE}$ to a lower bound value and to a general strategy for RMD washing which is $FIFO_{online}$. According to test results, $TIH_{CORRECTIVE}$ is able to give good pre-disinfection times in most cases. We tested also its impact on the rest of the sterilization process. Simulation results show that it performs better than $FIFO_{online}$ for waiting times in different stocks and the number of RMD sterilized per day.

For the future work, the results of the simulation model can be enriched by making some more statistical analysis. It is also possible to extend this work by considering purely online configurations, or some stochastic methods. Moreover, the simulation model can be improved by taking into account manpower representing the sterilization service operators.

REFERENCES

- Azizoglu, M. and Webster, S. 2001. Scheduling a batch processing machine with incompatible job families. *Computers and Industrial Engineering*, 39 (3-4), 325-335.
- Chang, Y., Damodaran, P. and Melouk, S., 2004. Minimizing makespan on parallel batch processing machines. *International Journal of Production Research*, 42 (19), 4211-4220.
- Chen, H., Du, B. and Huang, G.Q., 2010. Metaheuristics to minimise makespan on parallel batch processing machines with dynamic job arrivals. *International Journal of Computer Integrated Manufacturing*, 23 (10), 942-956.
- Chung, S.H., Tai, Y.T., and Pearn, W.L., 2009. Minimizing makespan on parallel batch processing machines with non-identical ready time and arbitrary job sizes. *International Journal of Production Research*, 47 (18), 5109-5128.
- Damodaran, P. and Velez Gallego, M.C., 2010. Heuristics for makespan minimization on parallel batch processing machines with unequal job ready times. *International Journal of Advanced Manufacturing Technology*, 49 (9-12), 1119-1128.
- Damodaran, P., Velez-Gallego, M.C. and Maya, J., 2011. A GRASP approach for makespan minimization on parallel batch processing machines. *Journal of Intelligent Manufacturing*, 22 (5), 767-777
- Dupont, L. and Jolai Ghazvini, F., 1998. Minimizing makespan on a single batch processing machine with

- non-identical job sizes. *European Journal of Automation*, 32 (4), 431-440.
- Dupont, L. and Dhaenens-Flipo, C., 2002. Minimizing the makespan on a batch processing machine with non-identical job sizes: an exact procedure. *Computers and Operations Research*, 29 (7), 807-819.
- EESS, 2007, Enquête Electronique sur les Services de Stérilisation.
- Graham, R., Lawler, E., Lenstra, J., and Kan, A., 1979. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey, *Annals of Discrete Mathematics*, 5, 287-326.
- Jolai Ghazvini, F. and Dupont, L. 1998. Minimizing mean ow times criteria on a single batch processing machine with non-identical job sizes. *International Journal of Production Economics*, 55 (3), 273-280.
- Kashan, A.H., Karimi, B., and Jolai, F., 2006. Minimizing Makespan on a Single Batch Processing Machine with Non-identical Job Sizes: A Hybrid Genetic Approach. *Evo-COP'2006*, 135-146.
- Kashan, A.H., Karimi, B., and Jenabi, M., 2008. A hybrid genetic heuristic for scheduling parallel batch processing machines with arbitrary job sizes. *Computers and Operations Research*, 35 (4), 1084-1098.
- Kashan, A.H., Karimi, B., and Fatemi Ghomi, S.T.M., 2009. A note on minimizing makespan on a single batch processing machine with nonidentical job sizes. *Theoretical Computer Science*, 410 (27-29), 2754-2758.
- Kashan, A.H., Karimi, B., and Jolai, F., 2010. An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with nonidentical job sizes. *Engineering Applications of Artificial Intelligence*, 23 (6), 911-922.
- Kempf, K.G., Uzsoy, R. and Wang, C., 1998. Scheduling a single batch processing machine with secondary resource constraints. *Journal of Manufacturing Systems*, 17 (1), 37-51.
- Lee, C.Y., and Uzsoy, R., 1999. Minimizing makespan on a single batch processing machine with dynamic job arrivals. *International Journal of Production Research*, 37 (1), 219-236.
- Li, S., Li, G., Wang, X. and Liu, Q., 2005. Minimizing makespan on a single batching machine with release times and non-identical job sizes. *Operations Research Letters*, 33 (2), 157-164.
- Lu, S., Feng, H. and Li, X., 2010. Minimizing the makespan on a single parallel batching machine. *Theoretical Computer Science*, 411 (7-9), 1140-1145.
- Nong, Q.Q., Ng, C.T. and Cheng T.C.E., 2008. The bounded single-machine parallel-batching scheduling problem with family jobs and release dates to minimize makespan. *Operations Research Letters*, 36 (1), 61-66.
- Ozturk, O., Di Mascolo, M., Gouin, A. and Espinouse M-L, 2011, Optimisation du chargement des laveurs dans un service de sterilization, in *proceedings of 4^{èmes} Journées Doctorales / Journées Nationales MACS*, Marseille, France
- Parsa, N.R., Karimi, B., Lee, and Kashan, A.H., 2010. A branch and price algorithm to minimize makespan on a single batch processing machine with non-identical job sizes. *Computers and Operations Research*, 37 (10), 1720-1730.
- Standard AFNOR FD S98-135. Sterilisation des dispositifs médicaux. Guide pour la maîtrise des traitements appliqués aux dispositifs médicaux réutilisables
- Uzsoy, R., 1994. Scheduling a single batch processing machine with non-identical job sizes. *International Journal of Production Research*, 32 (7), 1615-1635.
- Wang, H. and Chou, F-D. 2010. Solving the parallel batch-processing machines with different release times, job sizes, and capacity limits by metaheuristics. *Expert Systems with Applications: An International Journal*, 37 (2), 1510-1521.
- Yongqiang, S. and Enyu, Y. (2005). On-line problems of minimizing makespan on a single batch processing machine with nonidentical job sizes, *Applied Mathematics -A Journal of Chinese Universities*, 20(3) : 297-304
- Zhang, G., Cai, X., Lee, C.Y. and Wong, C.K., 2001. Minimizing makespan on a single batch processing machine with nonidentical job sizes. *Naval Research Logistics*, 48 (3), 226-240.
- Zhang, X., Zhang, Y., Cao, Z. and Cai, Z., 2007. Approximation Schemes for Scheduling a Batching Machine with Nonidentical Job Size. *Journal of Systems Science and Complexity*, 20 (4), 592-600.