M.Sc. in Health Systems Thesis

Optimization and Simulation of the Medical Device Sterilization in Hospitals

Azita Jafarbigloo

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
M.Sc. in Health Systems

Telfer School of Management
University of Ottawa

Abstract

There is no doubt Medical Devices have a crucial role in hospital processes such as surgeries and therapeutic procedures. Medical devices available in hospitals are of two types; reusable and non-resalable medical devices. Reusable medical devices are washed and sterilized after each use. The process of sterilizing medical devices is performed in the sterilization department. Each medical device travels through a cycle each time it is utilized. It is explicit that any part of the sterilization cycle that delays the process can cause serious problems for hospitals' performance. The washing step of the sterilization process has been a bottleneck in the system. Thus, optimization approaches can be highly advantageous to improve this bottleneck. The data of the medical devices are usually unknown prior to the scheduling process since the finishing time of the surgeries are not known in advance. Thus, there is no information available on the ready time of medical devices to be sterilized. Due to this factor, to develop applicable solutions, it is critical to consider this problem as an online problem and develop online scheduling methods. In this thesis, we take advantage of mathematical programming and heuristic algorithms to solve both the offline and online settings of the problem. We model the washing step of the sterilization cycle as a scheduling problem. Batch scheduling and bin packing, two well-known optimization approaches, are used for this purpose. Medical devices are batched together first and then scheduled on machines to reduce the total washing time of all medical devices. First, a mathematical model for the offline problem is provided and tested to solve the problem. Then a series of heuristic algorithms are developed using the batch scheduling approach for solving both offline and online problems. Moreover, a special case with divisible job sizes and equal release dates is studied. It was proved that for the strongly divisible sequence the First Fit Increasing algorithm finds the optimal solution, also for the weakly divisible sequence a Dynamic Programming algorithm is developed. Finally, we couple optimization with simulation to test the impact of the optimization of the washing step on the entire sterilization system. Moreover, since the next step of the sterilization cycle, the sterilization step, is very similar to the washing step, we also implement the developed heuristics in this step to evaluate its performance and improve it further. The results show that as long as the washing step is optimized it does not differ which algorithm is used in the sterilization step, thus, the optimization of this step is not necessary.

Table of Contents

List of Tables

List of Figures

List of Symbols

j	$Index\ of\ jobs$
k	$Index\ of\ batches$
m	$Index\ of\ machines$
s_j	$Size\ of\ job\ j$
r_j	$Release\ date\ of\ job\ j$
N	$Number\ of\ jobs$
M	$Number\ of\ machines$
C	$Machine\ capacity$ (represents both machine and batch capacity)
p	$Job\ processing\ time$
nb	$Lower\ bound\ on\ the\ number\ of\ batches$
x_{jkm}	$A\ binary\ decision\ variable, equals\ to\ 1\ if\ job\ j\ is\ executed\ in\ batch\ k\ and\ on\ machine\ m,\ 0\ otherwise$
b_{km}	$A\ binary\ decision\ variable, equals\ to\ 1\ if\ batch\ k\ is\ created\ on\ machine\ m, a\ 0\ otherwise$
S_{km}	$Ready\ time\ of\ batch\ k\ on\ machine\ m$
C_{max}	$Makespan$
L	$List\ of\ all\ the\ jobs$
I	$List\ of\ all\ the\ created\ batches$

br_k	$Batch\ k\ ready\ time$
mr_m	$Machine\ m\ ready\ time$
$p - batch$	$Parallel\ batching$

Harmonic Algorithm

i	$Index\ of\ items$
k	$Index\ of\ groups$
L'	$List\ of\ all\ items$
a_i	$size\ of\ item\ i$
I_k	$Interval\ of\ group\ k$
$I_k - pieces$	$Indicates\ a\ piece\ that\ blongs\ to\ interval\ k$
$I_k - bin$	$Indicates\ a\ bin\ that\ only\ pack\ I_k - pieces$
M	$An\ integer$

Chapter 1

Introduction

This thesis investigates the problem of loading the medical devices in the washing step of the sterilization department of hospitals. The sterilization services are used in hospitals to sterilize reusable medical devices. Reusable medical devices are utilized in different departments of the hospitals, such as surgeries. These medical devices need to go through a specific process to get sterilized and ready for reuse. The washing step is usually the bottleneck of the system and causes an increase in sterilization time of the medical devices (Di Mascolo and Gouin, 2013).

We propose a solution for the long waiting time of the RMDs in the washing step by defining the problem as a batch scheduling problem and developing algorithms to find near-optimal solutions for batch creation. We also propose a simulation approach for the sterilization cycle to show the performance of optimization algorithms in the cycle.

1.1. Reusable Medical Devices (RMDs)

Medical devices can be one of two types; reusable medical devices or non-reusable medical devices. Non-reusable or single-use medical devices are such as Hypodermic needles, syringes, applicators, bandages and wraps, face masks, etc. Reusable medical devices include surgical instruments such as clamps and forceps, endoscopes, and their accessories such as graspers and scissors. Only, the reusable medical devices are sent to the sterilization department to be washed and sterilized. From now on whenever we talk about medical devices we are just talking about reusable medical devices (RMDs).

1.2. Sterilization Process

In general, the sterilization process contains different subprocesses; pre-disinfection, rinsing, washing, verification, packaging, and sterilization. The medical devices that go through these steps will be ready to be reused at the end, therefore the sterilization processes form a cyclic process. These steps are explained in this part.

***Pre-disinfection or Soaking*:** In this step, the RMDs will be soaked (immersed) in a pre-disinfectant liquid. Therefore, the pre-disinfection step will prevent the drying of the stains and facilitate the stain removal process during cleaning. After pre-disinfection, RMDs are transferred to the sterilization department. The minimum pre-disinfection period is 15 minutes while 50 minutes is generally considered as the upper limit, and 20 minutes is the ideal duration (Di Mascolo and Gouin, 2013).

Rinsing*:* To avoid the risk of interference between the pre-disinfection products and washing products the RMDs must be rinsed before going to the washing step. Rinsing can be carried out either automatically with washers or manually in operating rooms before going to the sterilization department. In the case of automatic rinsing, the RMDs wait-time before entering the washers becomes important to avoid exceeding the upper limit of pre-disinfection time and damaging RMDs.

Washing*:* Washing step is a combination of a chemical and a mechanical action. In this step all the stains will be completely cleaned in the washers using a chemical cleaner then the washed RMDs will be dried. The washers are filled with baskets that contain the RMDs. The capacity of the washers can vary, from 4 to 12 DIN baskets (washers' capacities are described with the unit DIN). Normally, RMDs have the same duration of washing.

Verification*:* The RMDs will be verified to be in acceptable condition, not broken or damaged and, washed thoroughly.

Packaging*:* RMDs are packaged in containers or boxes at the packaging station. The packaging ensures the sterility of the RMDs until reuse. They prevent the permeating of any microorganisms, but they allow the penetration of the sterilization chemicals in the sterilization step.

Sterilization*:* this step is done using a machine called "Autoclaves". Autoclaves apply water vapor to DMRs conditioned. The goal is to eliminate all germs or contaminants by putting the object at a temperature of 134 ° C plus an overpressure of 2.2 bars over the entire surface to be sterilized. Warming and overpressure are applied several times in a sterilization cycle. The duration from one cycle in an autoclave can vary from one autoclave to another (from 1 to 2 hours) (Di Mascolo and Gouin, 2013).

Transfer*:* Once the RMDs are sterilized, they will be transferred to the operating rooms using cabinets or closed carts. These carts and cabinets are also sterilized and cleaned before each use.

Storage*:* Finally, the RMDs will be stored away from light and moisture to keep them clean and ready for reuse.

In the next two sections, the batch scheduling and its relationship with the sterilization problem are explained.

1.3. Washing Step as a Batch Scheduling Problem

The batch scheduling problem is a famous problem in the Operations Research literature. The problem studied in this thesis consists of non-identical jobs with different sizes and arrival times and multiple machines working in parallel with the same capacity and processing time. Multiple jobs can be batched together and scheduled on machines as long as the summation of the job sizes in a batch does not exceed the machine capacity. This problem will be explained more in detail in the next chapters.

The washing step can be modeled as a batch scheduling problem, where RMDs are jobs with different release times and different sizes. The washing machines are machines that perform the tasks. The objective of this problem is to improve the efficiency of the washing step and eliminate the bottleneck while maintaining the desired quality of sterilization. For this purpose, we propose an optimization model for the washing step and provide batch scheduling algorithms to solve this problem. As the objective function, we can notice that there are different objective functions compatible with solving this problem, such as:

Minimizing the total washing time for each RMD
($Min \sum C_j$ *, where* C_j *is the completion time of job* j),

Minimizing the total washing time of the step (makespan, Minimizing C_{max}),

Minimizing the number of batches created in the washing step.

The remainder of this paper describes how to use batch scheduling for optimizing the washing step of the sterilization process. The following chapter reviews previous related work on batch scheduling and sterilization process to identify potential gaps and gain a better understanding of the problem. In chapter 3, we present the problem description and the assumptions, and in chapter 4, we describe our methodology for solving the problem. Chapter 5 contains the simulation model for the sterilization cycle. And finally, in chapter 6, we will provide a conclusion and future works.

Chapter 2

Literature Review

Healthcare-related problems have been getting lots of attention from researchers these days, this also applies to the application of Operations Research in the healthcare context. Before starting the modeling and developing a solution for our problem it is necessary to review the literature and understand the gaps and gain a better idea of the researches conducted in the literature for solving similar problems.

The application of batch scheduling and simulation approaches, that we use to model the washing step of the sterilization process, was studied in several research papers to solve similar problems. The following sections focus on studying the batch scheduling and simulation approaches in the literature. However, the medical device sterilization scheduling problem has received limited attention in the literature, and there are just a few papers on this topic, they are also described in the next section.

2.1. Batch Scheduling on Parallel Machines

Generally, batch scheduling problems are classified into two groups based on the batching type: serial batching (s-batch) and parallel batching (p-batch). Serial batching is when all the jobs in the same batch are processed one by one and the processing is completed with the execution of the last job in the batch. A batch scheduling is called parallel batching when the execution of all jobs happens at the same time on one machine. The processing time of the batch will be equal to the processing time of the job with the longest processing time in the batch. The problem that we will be solving here is a parallel batching (p-batch) problem.

In the parallel batching problems with different job sizes, each job should be assigned just to one batch and the size of the batch will be defined with the summation of the job sizes in that batch, also, the batch size should not exceed the batch capacity. The processing time of the batch is equal to the longest processing time of the jobs in the batch (Potts and Kovalyov, 2000).

We can classify the batch scheduling with non-identical job sizes and single or parallel machines into two groups: offline batch scheduling and, online batch scheduling.

The rest of the literature review is organized based on the above classification and the literature in each group is discussed furthermore.

2.1.1. Offline Batch Scheduling

To the best of our knowledge, (Uzsoy, 1994) was the first to study parallel batching on a single machine with identical release dates and considering different job sizes. They studied the minimization of the total completion time and makespan. They proved that these problems are strongly NP-hard. They proposed a branch and bound for minimizing the total completion time and heuristics for both objectives, and designed heuristics that decreased the computation time and rapidly find near-optimal solutions. Later (Ghazvini and Dupont, 1998) considered jobs with non-identical sizes on a single machine, to minimize the mean-flow time (total completion time) of jobs. They provided various heuristics based on the bin packing problem. The first heuristic was modified best-fit which was inspired by the best-fit algorithm in the bin packing literature. Then they proposed a heuristic based on the best-fit algorithm and finally, they developed an iterative, parametric heuristic that outperformed the previous existing algorithms.

(Kempf, 1998) studied the problem of minimizing total completion time of jobs and makespan minimization with job families. They developed an integer programming model and heuristics to solve the problem of single batch processing machine. Their heuristic was inspired by (Uzsoy, 1994) Batch First Fit (BFF) heuristic and called Random Batch First Fit, it also considered machine capacity. (Azizoglu and Webster, 2000) proposed a branch and bound approach for the batch processing problem with arbitrary job processing times, job weights and job sizes. Their objective was to minimize total weighted completion time. Their approach was able to find the optimal solution for up to 25 job instances in a short time. They also later presented another branch and bound approach for the incompatible job families with the same objective (Azizoglu and Webster, 2001).

(Dupont and Dhaenens-Flipo, 2002) solved the problem with the objective of minimization of makespan. They proposed a branch and bound and used it as a heuristic for the large-scale instances. (Melouk, Damodaran and Chang, 2004) used the simulated annealing (SA) approach. They compared their result to the CPLEX results and proved that their results outperform the CPLEX for all instances. In (Rafiee Parsa, Karimi and Husseinzadeh Kashan, 2010), the objective was minimizing the makespan. They formulated the problem using Dantzig-Wolfe decomposition as a set partitioning problem. They developed a heuristic that provided the lower bound using the column generation method. Then they presented a branch and price algorithm that combines the branch and bound method and column generation technique and obtains the optimal solution.

(Chang, Damodaran and Melouk, 2004) and (Kashan, Karimi and Jenabi, 2008) presented heuristics for the identical parallel batch processing problem to minimize the makespan. (Chang, Damodaran and Melouk, 2004) proposed a simulated annealing approach for the identical parallel batch processing problem and (Kashan, Karimi and Jenabi, 2008) proposed a hybrid genetic heuristic, they also compared their result to (Chang, Damodaran and Melouk, 2004) and showed that their approach can obtain near-optimal solution in reasonable time and outperforms the SA algorithm.

(Chung, Tai and Pearn, 2009) was the first to study the parallel machine with unequal job release dates and non-identical job sizes. Their objective was to minimize the total completion time. They proposed a mixed-integer linear programming (MILP) model and three heuristic algorithms to reduce the computation time for solving the large-scale problem. Their proposed heuristic is based on the DELAY algorithm (Lee and Uzsoy, 1999) and it consists of two phases, phase one is devoted to creating the batches and adding the jobs to them and phase two is for scheduling the created batches on the machines.

(Husseinzadeh Kashan, Karimi and Jolai, 2010) solved the multi-objective problem. They considered minimization of the bicriteria of makespan and maximum tardiness as their objective and developed a meta-heuristic approach (genetic algorithm (GA)) for solving the problem. (Lee and Lee, 2013) studied the problem of batch scheduling on a single machine and jobs with non-identical sizes to minimize the maximum completion time. Their heuristics were developed by iterative decomposition of a mixed-integer programming model inspired by (Ghazvini and Dupont, 1998) and (Chen, Du and Huang, 2011).

(Damodaran *et al.*, 2011) considers parallel batch scheduling problem and propose a metaheuristic algorithm to minimize the makespan. They showed that their algorithm can find the optimal solution for 10 job problem instances and for large-scale problems they can find near-optimal solutions better than (Chung, Tai and Pearn, 2009). (Damodaran and Velez-Gallego, 2010) presented a constructive heuristic to solve the parallel batch scheduling problem with different release times. Their heuristic is an extension of the successive knapsack problem for solving $1|batch|Cmax$ (based on Graham's notation (Graham 1979)), proposed by (Dupont and Ghazvini, 1998). Their results were close to the proposed meta-heuristic approach in (Damodaran *et al.*, 2011) and it outperformed the result of (Chung, Tai and Pearn, 2009). (Chen, Du and Huang, 2010) presented two lower bounds to evaluate the performance of the approximation algorithms. Then an ERT-LPT (earliest ready time- longest processing time) heuristic rule was presented for assigning batches to parallel machines. They also, proposed two meta-heuristic algorithms, a genetic algorithm (GA) and an ant colony optimization (ACO), using ERT-LPT with the objective of minimizing the makespan. They compared their result with the result of heuristics developed by (Chung, Tai and Pearn, 2009) and showed that their result outperforms the heuristic approach. (Wang and Chou 2010) considered machines with different capacities. They provided a mixed-integer programming (MIP) model and meta-heuristic based on simulated annealing (SA) and genetic algorithm (GA). They used the instances of (Chung, Tai and Pearn, 2009) and used their experimental result as their benchmarking results and showed that their algorithm has a better performance.

(Abedi *et al.*, 2015) proposed a bi-objective-mixed integer linear programming to solve the problem with capacity limits. Their objective was to minimize the makespan and total weighted earliness and tardiness of jobs. After developing the bi-objective model, they developed an Ɛ-constraint method to solve it. (Ozturk, Begen and Zaric, 2017) studied the parallel batching problem for jobs with different processing times, release dates, and unit sizes. The objective of the problem was to minimize makespan. They proposed a branch and bound algorithm, for solving the parallel machine problem instances.

(Ham, Fowler and Cakici, 2017) addressed the parallel batch-scheduling problem which involves the constraints of different job release times, non-identical job sizes, and incompatible job families which can be represented as $P_m|r_j|\sum w_j C_j$ based on Graham's notation (Graham 1979). Mixed-integer programming and constraint programming models were proposed and tested on a set of common problem instances from the literature. (Soleimani *et al.*, 2020) solved the problem of scheduling of unrelated parallel machines considering the effects of start-time-related deterioration, position-related learning, and sequence-related setup times, to minimize the mean weighted tardiness and power consumption. Three metaheuristic algorithms, Genetic Algorithm (GA), Cat Swarm Optimization (CSO), and Interactive Artificial Bee Colony (IABC), were proposed to obtain quality solutions within acceptable computation time. (Ozturk, 2019) considered the problem of hospital sterilization services for the cases of internal and external sterilization services and they minimized two objectives: makespan and flow time of the washing step. They provided a mixed-integer linear programming model and a dynamic programming model.

2.1.2. Online Batch Scheduling

The following papers investigate the online batch scheduling problem, in these studies online implies that jobs arrive over time and jobs' characteristics are unknown before their arrival. Online scheduling on parallel-batch machines has been studied widely in the literature. Online algorithms can be categorized into two types; delay algorithms, in which the algorithm always waits until some conditions are satisfied in time, even if jobs are available or machines are idle, and, greedy algorithms, which immediately assigns the jobs to the idle machines if any exists, and does not wait for any time conditions (Tian, Fu and Yuan, 2014).

(Zhang, Cai and Wong, 2001) and (Deng, Poon and Zhang, 2003) were the first to research online scheduling on parallel-batch machines problem, they studied online scheduling on a single batch machine to minimize the makespan. They provided the best possible delay algorithm with a competitive ratio of $(\sqrt{5}) + 1)/2$. (Lee and Uzsoy, 1999) presented a greedy algorithm for the unbounded problem which (Liu and Yu, 2000) proved to be 2-competitive. Later, (Fu *et al.*, 2007) studied the problem of online parallel-batch scheduling with jobs in a batch being allowed to restart. They considered jobs with non-identical processing times and interrupting a batch is allowed (when a batch is interrupted all jobs in the batch become unbatched and their process starts over). They proved that there is no online algorithm with a competitive ratio less than $(5 - \sqrt{5})/2$ for this problem. (Zhang *et al.*, 2006) studied the problem with the minimizing the total completion time objective. They provided three approximation schemes with worst-case ratios of 4, 2, and 3/2.

(Li *et al.*, 2005) considered the single batching machine scheduling problem where jobs have different release times and non-identical job sizes. Their objective was the minimization makespan. They presented an approximation algorithm with a worst-case ratio of 2 + Ɛ where Ɛ > 0. (Nong, Ng and Cheng, 2008) proposed a polynomial-time approximation scheme for identical job sizes and approximation algorithm with a worst-case ratio of 5/2 for non-identical job sizes to minimize the makespan. (Lu, Feng and Li, 2010) considered the problem of scheduling with release date and rejection on a single parallel batching machine. Their objective was to minimize the makespan of the accepted jobs and total penalty of the rejected jobs. They proposed a polynomial-time approximation scheme for the split jobs and a 2-approximation algorithm for the identical job release dates, and they presented an approximation algorithm for the general problem with a worst-case ratio of 2 + Ɛ where Ɛ > 0.

(Fang, Lu and Liu, 2011) considered online batch scheduling problem on parallel machines with delivery times. The objective of this study was to minimize the time by which all jobs are delivered. They consider both bounded and unbounded batch machines when job processing times are equal and presented a general online algorithm with a competitive ratio of 2 when the number of machines equals to 2 or 3, and $1.5 + o(1)$ for more than 3 machines. (Cheng *et al.*, 2012) considered the scheduling problem of parallel batch processing machines with non-identical job sizes. The models of minimizing makespan and total completion time were given using the mixed-integer programming method. Then a polynomial-time algorithm was proposed, and the worst-case ratios were proved to be 2 and (8/3–2/3 m) respectively.

(Ma *et al.*, 2014) investigated the online bounded batch scheduling problem with parallel machines to minimize the total weighted completion time. The job processing times are not equal and the processing time of a batch equals the processing time of the longest job in the batch. They presented $4(1 + \varepsilon)$-competitive online algorithms. (Liu and Lu, 2014) studied the problem of online unbounded batch scheduling with m identical machines subject to release dates and delivery times. The objective was to minimize the time that all jobs have been delivered. They first studied a simplification of the general model where processing time and delivery time of the jobs are agreeable. Then they provided an online algorithm for the general problem with competitive ratio of $1 + 2/\lfloor\sqrt{m}\rfloor$.

(Ozturk *et al.*, 2012a) studied the problem of hospital sterilization services and proposed a 2-approximation algorithm for solving the problem of parallel batch processing machines and jobs with non-identical sizes, different release dates, and equal processing times with the objective of minimizing the makespan. They also proposed a MILP model that outperformed the previous models from related literature. (Selvarajah, Steiner and Zhang, 2013) tried to minimize the sum of weighted flow times and delivery cost. They proposed an approximation algorithm for the problem with identical weights and an evolutionary metaheuristics algorithm for the general case.

(Li and Li, 2015) investigated the online batch scheduling problem with equal-length jobs on two identical batch machines, where each machine can process up to b jobs at the same time. Their goal was to maximize the number of early jobs. They provide a greedy online algorithm with a competitive ratio of $1/(b+1)$. (Fang and Lu, 2016) studied online parallel-batch scheduling problem on a single unbounded machine. Their objective was the minimization of total weighted job completion time and they presented an algorithm with a competitive ration of $\sqrt{5}+1$ for the general case, and when all jobs have the same processing time, they were able to provide an optimal online algorithm. Later, (Chen *et al.*, 2016) considered the problem of scheduling on parallel identical machines with late work criterion and a common due date. They studied both offline and online problems and provided algorithms for both cases and used the offline problem for the analysis of the online problem.

In (Arroyo and Leung, 2017b) the objective was to minimize the makespan. They presented several heuristics based on first-fit and best-fit earliest job ready time rules. They also presented a mixed-integer programming model for the problem and a lower bound to evaluate the quality of the heuristics. They also proposed a meta-heuristic based on an iterated greedy (IG) algorithm in (Arroyo and Leung, 2017a). (Li, 2017) also solved the problem to minimize the makespan. First, they proposed a fast 5-approximation algorithm for the case with equal release times, and for the case of unequal release times, they obtained an algorithm with an approximation ratio close to 2.

(Li and Chai, 2018) presented online scheduling problem on identical parallel-batch machines to minimize maximum weighted completion time. The processing time of the jobs in this problem are identical and jobs arrive over time and batch capacity is bounded. They present an online algorithm with a competitive ratio of $(\sqrt{5}+1)/2$. (Chai, Li and Zhu, 2019) studied the online scheduling problem on a bounded parallel-batch machine to minimize the maximum weighted flow time. They presented a deterministic online algorithm with a competitive ratio $(\sqrt{4w+1}+1)/2$ when $w \in [1,2]$ and not greater than w when $w \in (2,+\infty)$. (Liu, Lu and Li, 2020) considered online scheduling of an unbounded parallel-batch machine to minimize the time by which all jobs are delivered. They allowed a limited batch restart for the running batches, where limited means that if a running batch contains restarted a job it cannot restart again. They also considered that all jobs have agreeable processing times and delivery times and provided an online algorithm with a competitive ratio of $3/2$.

2.2. Coupling optimization with simulation

Simulation has been used broadly to help the optimization methods and for adding validity to the methodology. Some cases of coupling simulation and optimization approaches from literature are provided below.

(Arakawa, Fuyuki and Inoue, 2003) studied the elimination of tardy jobs in a job shop production schedule, an optimization-oriented simulation-based scheduling method that incorporates capacity adjustment function was proposed. (Dias *et al.*, 2018) proposed a framework for the integration of scheduling and model predictive control (MPC), which is applicable to industrial-size problems involving fast-changing market conditions. Their framework consists of identifying scheduling-relevant process variables, building dynamic models to capture their evolution, and integrating scheduling and MPC by, first solving a simulation-optimization problem to define the optimal schedule and, then tracking the schedule in a closed-loop using the MPC controller.

In (Doleschal, Klemmt and Weigert, 2011) a selected part of the process with high practical impact was investigated. This was a scheduling problem in the semiconductor frontend oxidation and diffusion area. The optimization objective was the total weighted tardiness. The methods dispatching, simulation-based optimization, mixed integer programming (MIP), and variable neighborhood search (VNS) were compared. (Freitag and Hildebrandt, 2016) described the use of simulation-based multi-objective optimization (multi-objective Genetic Programming) as a hyper-heuristic to automatically develop improved dispatching rules.

(Cabrera *et al.*, 2011) presented an agent-based modeling and simulation to design a decision support system for the operation of healthcare emergency departments. Their main objective in the simulation model was to optimize the performance of Emergency departments by minimizing patient waiting time and maximizing patient throughput.

(Zhang *et al.*, 2012) integrated demographic and survival analysis, discrete event simulation, and optimization approaches to solve the problem of capacity planning in Long-term care. (Chen *et al.*, 2015) used a simulation-optimization approach for allocation of healthcare services. They implemented four types of patients' arrival as appointment scheduling policies for scheduling patients. They considered patients' waiting time, idle time of resources and total cost as their main objective functions and provided optimal and near-optimal solutions. (Lucidi *et al.*, 2016) presented a simulation-based optimization approach for resource planning in hospital wards. They used a discrete-event simulation model to find out the efficient setting and then they use a derivative-free optimization algorithm to modify the setting obtained from the simulation.

Chapter 3

Problem Description

In this section, we define the problem of the washing step of the sterilization cycle.

As stated earlier, the washing step of the sterilization cycle is acknowledged to be a bottleneck in the system that results in a long wait-time for RMDs to become sterilized and ready for reuse. By improving the performance of the bottleneck, we can reduce the wait-time. It is crucial to mention that due to tracking issues, it is not desirable to separate the RMDs in one set to wash them in different machines. Doing so can create problems for tracing the RMDs in a set and also the process of putting them back. These can lead to an extra delay in the sterilization process. The RMD sets can arrive at the department at any time of the day, and all of them have the same washing process duration.

The RMD sets in the washing step are considered as jobs in the batch scheduling problem. Jobs have different sizes and release times, which indicates the size of the RMD sets and the time that they enter the washing step. Jobs are batched together based on different rules in the algorithms. All batches have the same fixed capacity and, the summation of job sizes in one batch cannot exceed the batch capacity. A batch ready time equals the greatest job release time among all the jobs in the batch. Washing machines are described as machines in the batch scheduling problem. Machines are identical, i.e., their processing time and capacity are the same. When batches are ready, they are scheduled on machines, and when a machine is ready, it starts processing, i.e., washing RMDs. After a machine finishes its process, it becomes available for new batches and the processed jobs (RMDs) leave the washing step and proceed to the next step of the sterilization cycle.

The following assumptions and notations are used for the rest of this thesis.

There exists a set of N jobs, $J = \{1, 2, \dots, N\}$, to be processed on M identical parallel batch processing machines, $m \in \{1, 2, \dots, M\}$. For each job $j \in J$ the job size (s_j), job release date (r_j) are known in advance for the offline cases, and known at the time of arrival for the online cases, all jobs have the same job processing time (p). Job release date is the time that job arrives at the washing step and becomes ready for washing, which is the earliest time that the job can start processing. Jobs have unidentical sizes and release dates.

Each machine m has a maximum capacity C and processing time p. All machines are identical, and their processing time is equal to the job processing time. Each machine can process multiple jobs in batches as long as the total size of all the jobs on the machine does not exceed the machine capacity (C).

Inspired by Graham's notation (Graham 1979), the following notation is used for this problem: $P|p - batch, r_j, p_j = p, s_j, C|C_{max}$. P refers to identical parallel machines, *p-batch* indicates parallel batching, $p_j = p$ shows identical processing time, r_j and s_j demonstrates job release time and job size, C is for machine (batch) capacity and C_{max} refers to the makespan.

Assumptions:

- Splitting jobs into different batches is not allowed.
- Machines cannot be interrupted during the process.
- Machine capacity is equal to batch capacity.
- Total job sizes on a batch should not exceed machine capacity.
- All jobs have the same processing time p.

Chapter 4

Solution Methods

This section discusses the methodological framework for solving the problem. First, a mixed-integer linear programming model to solve the offline batch scheduling problem is presented, then two offline heuristic algorithms for finding near-optimal solutions in a shorter computational time are discussed. And finally, a series of heuristic algorithms for online batch scheduling are developed.

4.1. Mixed-Integer Linear Programming Model

(Chung, Tai and Pearn, 2009) formulated an MILP model for the problem with unequal processing time, $P \mid p - batch, rj, pj, wj, B \mid Cmax$, and later inspired by their work (Ozturk *et al.*, 2012b) created an MILP model for the same problem with equal processing times, $P \mid p - batch, rj, pj = p, wj, B \mid Cmax$. (Ozturk *et al.*, 2012b) model was developed for the sterilization problem. Therefore, it is completely applicable to the problem addressed in this paper. The below explanation is the description of the MILP model.

List of indices:

$j = 1, \dots, N\ jobs$

$k = 1, \dots, N\ batches$

$m = 1, \dots\ M\ machines$

Parameters:

s_j $size\ of\ job\ j$

r_j $release\ date\ of\ job\ j$

N $number\ of\ jobs$

M $number\ of\ machines$

C $machine\ capacity$ (batch capacity)

p $job\ processing\ time$

nb $lower\ bound\ on\ the\ number\ of\ batches \left(nb = \left\lceil \sum_{j=1}^{N} s_j / C \right\rceil\right)$

Decision variables:

$x_{jkm} = 1$ *if job* j *is executed in batch* k *and on machine* m, 0 *otherwise*

$b_{km} = 1$ *if batch* k *is created on machine* m, 0 *otherwise*

S_{km} *ready time of batch* k *on machine* m

C_{max} *makespan*

Mathematical formulation:

$$Minimize\ C_{max},$$

Subject to

$$\sum_{k=1}^{N}\sum_{m=1}^{M} x_{jkm} = 1, \quad j = 1, \dots, N, \tag{1}$$

$$\sum_{j=1}^{N} s_j * x_{jkm} \leq C * b_{km}, \quad k = 1, \dots, N, m = 1, \dots, M, \tag{2}$$

$$\sum_{m=1}^{M} b_{km} \leq 1, \quad k = 1, \dots, N, \tag{3}$$

$$S_{km} \geq x_{jkm} * r_j, \quad j = 1, \dots, N, k = 1, \dots, N, m = 1, \dots, M, \tag{4}$$

$$S_{km} \geq S_{k-1,m} + p * b_{k-1,m}, \quad k = 2, \dots, N, m = 1, \dots, M, \tag{5}$$

$$C_{max} \geq S_{Nm} + p * b_{Nm}, \quad m = 1, \dots, M, \tag{6}$$

$$b_{k,(k\ mod\ M)+1} \geq 1, \quad k = 1, \dots, nb, \tag{7}$$

$$b_{km} = 0, \quad k = nb + 1, \dots, N, \forall m \neq (k\ mod\ M) + 1, \tag{8}$$

$$x_{jkm} \in \{0,1\}, \quad b_{km} \in \{0,1\}, \quad S_{km} \geq 0, \quad C_{max} \geq 0.$$

The objective function is minimizing the makespan. Constraints set (1) ensures that all the jobs are assigned to batches and machines. Constraint set (2) is the capacity constraint. Constraint set (3) makes sure that each batch is assigned to at most one machine. Constraint set (4) sets the ready time of the batch, based on the job with the largest release time in the batch. Constraint set (5) ensures that if there is more than one batch scheduled on one machine, there is a time difference (p) between the processing of batches, this set of constraints also automatically assigns ready times to the batches $1, \dots, N$. Constraint set (6) sets the C_{max} value. It is known that at least nb batches will be created. Constraint set (7) assigns the first nb batches on the machines consecutively, $k \ mod \ M$ determines which machines should the batch be scheduled on. For $nb + 1$ batches, we cannot be certain that these batches will be created, thus, $b_{k,(k \ mod \ M)+1}$ can be 1 or 0. Since batches can be assigned to at most one machine the binary variable b_{km}, for $m' \neq (k \ mode \ M) + 1$ must be zero. Therefore, Constraint set (8) sets the binary variables that should be zero.

Since this MILP model was developed for the similar problem, it can be used in this thesis for obtaining the optimal solution and analyzing the performance of the heuristic algorithms.

4.2. Heuristic Algorithms

In this section, we start with developing and testing heuristic algorithms for the offline batch scheduling problem, then we extend our research to the online problem and propose online algorithms. In an offline problem, the job information is known in advance, i.e., job sizes and their release dates are known. Moreover, the problem is called online when no information on jobs is provided, and job sizes and release dates are determined when a job arrives at the system. Therefore, we cannot use the approaches that schedule based on the job sizes (decreasing or increasing order) for the online problem.

4.2.1. Offline batch scheduling

The offline algorithms are addressed in this section, offline means that job sizes and release dates are known in advance. We propose two algorithms based on the First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) to solve the offline problem. Best fit and first fit algorithms have been developed for solving the bin packing problem (Johnson, 1973), however, it has been shown that a modification of these algorithms can also be used in batch scheduling.

For all the algorithms proposed in this section, the two-phase approach is used. The first phase consists of a heuristic for adding the jobs to the batches and the second phase consist of scheduling the batches on machines. From now on, we refer to the first phase as the batching phase and the second phase as the scheduling phase.

4.2.1.1. First Fit Decreasing Algorithm

The general First Fit Decreasing (FFD) and First Fit (FF) algorithms work based on putting each item in list L to the first open bin that can accommodate the item, if there is no bin available a new bin will be created and puts the item in the new bin. The only difference between these two algorithms is that the first fit decreasing first sorts the items in L in a decreasing order. (György Dósa 2007) proved that the approximation ratio of FFD algorithm is $11/9$ $(FFD(L) \leq 11/9 \ OPT(L) + 6/9)$. The approximation ratio of an algorithm is the ratio between the result obtained by the algorithm and the optimal solution.

A modified version of the first fit decreasing is used for the first phase of the batch scheduling problem, batching phase.

The proposed algorithm is as follows:

First Fit Decreasing Algorithm

Phase 1:

1. Sort the jobs in L in increasing order of job release times, break ties by prioritizing the larger job sizes.
2. For each Job j in L:
   2.1. Put the job j in the first open batch that can accommodate the job and update the batch ready time.
   2.2. If there is no available batch, create a new one and add the job to the batch. Update the batch ready time.
   End for

Phase 2:

1. Sort all the created batches in an increasing order of their ready time. List I contains the all the batches.
2. For each batch in I:
   2.1. Put the batch on the first available machine and update the machines ready time.
   2.2. If the machine is full start the processing and update the machines finish time.
   2.3. If there is no machine available, schedule the batch on the first machine that finishes its processing and update its ready time.
   End for

4.2.1.2. Best Fit Decreasing Algorithm

Like the FFD algorithm the Best Fit Decreasing (BFD) and Best Fit (BF) algorithms are different in sorting the jobs based on their sizes. The general BF algorithm places each item in the bin that has the smallest remaining capacity that can fit the item, otherwise, opens a new bin and places the item in it.

A modified version of the Best Fit Decreasing is used for the first phase of the batch scheduling problem, batching phase.

The proposed algorithm is as follows:

Best Fit Decreasing Algorithm

Phase 1:

1. Sort the jobs in L in increasing order of job release times, break ties by prioritizing the larger job sizes.
2. For each Job j in L:
   - 2.1. Put the job j in an open batch with the smallest remaining capacity, that can accommodate the job and update the batch ready time.
   - 2.2. If there is no available batch, create a new one and add the job to the batch. Update the batch ready time.

   End for

Phase 2:

1. Sort all the created batches in an increasing order of their ready time. List I contains the all the batches.
2. For each batch in I:
   - 2.1. Put the batch on the first available machine and update the machines ready time.
   - 2.2. If the machine is full start the processing and update the machines finish time.
   - 2.3. If there is no machine available, schedule the batch on the first machine that finishes its processing and update its ready time.

   End for

4.2.2. Online batch scheduling

The first two online algorithms are similar to the offline algorithms, First Fit (FF) and Best Fit (BF), the only difference is that in these algorithms we do not sort the jobs based on their sizes. Finally, a Harmonic algorithm is used to develop a better online algorithm with lower approximation ratio.

Both FF and BF algorithm has been proven to have an approximation ratio of 1.7 in the bin packing problem ($FF(L) \leq \lfloor 1.7\ OPT \rfloor$ and $BF(L) \leq \lfloor 1.7\ OPT \rfloor$) (György Dósa 2014).

The Online algorithms are proposed only in one phase, since, in contrast to the offline algorithms, these algorithms should add jobs in batches whenever a job arrives and schedule batches on machines when the batch is full, or the batch scheduling rule was satisfied. Therefore, it cannot be separated in two phases.

Various batch scheduling rules can be considered for scheduling a batch on a machine, some of them can be as follows:

- *Batch capacity:* schedule a batch on a machine when the batch is full.
- *Time constraint:* schedule a batch on a machine after time T has passed from the time that batch was created.
- *Remaining batch size:* schedule a batch on a machine if the remaining batch capacity is less than q.
- *Job arrivals:* schedule a batch on a machine if y jobs have arrived at the system after the batch was created.

4.2.2.1. First Fit Algorithm

First Fit Algorithm

For each job j in L:

1. Put job j in the first batch that can accommodate the job and update the batch ready time.
2. If there is no available batch, create a new one and add the job to the batch. Update the batch ready time.
3. If the batch is full (or the batch scheduling rule is met):
   - 3.1. Schedule the batch on the first available machine.
   - 3.2. Update the machine ready time.
   - 3.3. If the machine is full start the processing and update machine finish time.
   - 3.4. If there is no machine available, schedule the batch on the first machine that finishes its processing and update its ready time.

End For

4.2.2.2. Best Fit Algorithm

Best Fit Algorithm

For each Job j in L:

1. Put the job j in an open batch with the smallest remaining capacity, that can accommodate the job and update the batch ready time.
2. If there is no available batch, create a new one and add the job to the batch. Update the batch ready time.
3. If the batch is full (or the batch scheduling rule is met):
   - 3.1. Schedule the batch on the first available machine.
   - 3.2. Update the machine ready time.
   - 3.3. If the machine is full start the processing and update machine finish time.
   - 3.4. If there is no machine available, schedule the batch on the first machine that finishes its processing and update its ready time.

End For

4.2.2.3. Harmonic Algorithm

The last algorithm is inspired by the Harmonic algorithm which is generally used for bin packing problems. The following is a brief description of the Harmonic algorithm and some modification for application in the batch scheduling problem.

Harmonic$_M$ and Refined Harmonic algorithms were first developed by (Lee, 1985) with a worst-case performance ratio of 1.691 and 1.636 respectively. Harmonic algorithm performs based on the idea of grouping the jobs based on their size and adding each job to its own group. The interval of groups is defined, and each job is added to the group if its size belongs to that interval. Then, instead of considering each job's size, just the job's group is considered for scheduling purposes. Each job can only be added to a batch of its own group, thus, all the jobs in one batch belong to the same group.

Grouping in Harmonic$_M$ bin packing algorithm is as follows. List $L' = \{a_1, a_2, a_3, \dots, a_n\}$ where $a_i \in (0,1]$ for all i, each item is classified according to the Harmonic$_M$ partitioning: $(0,1] = \cup_{k=1}^{M} I_k$ where $I_k = (\frac{1}{k+1}, \frac{1}{k}]$, $1 \le k < M-1$ and $I_M = (0, \frac{1}{M}]$ where M is an integer. a_i is called an $I_k - piece$ if $a_i \in I_k$, $1 \le k \le M$. Similarly, the bins will be partitioned to M categories and each $I_k - bin$ is designed to pack only $I_k - pieces$. Each $I_k - bin$ can only pack k $I_k - pieces$.

It is clear from the grouping intervals that Harmonic$_M$ partitions jobs with sizes bigger than ½ in the same group. Since these jobs create the first interval ($I_1 = (\frac{1}{2}, 1]$), there is always just one job in a batch. For jobs withs sizes close to ½ or slightly bigger this approach seems too wasteful, as a solution in Refined Harmonic new $M+2$ intervals based on Harmonic$_M$ intervals are developed with the goal to add the small $I_1 - pieces$ to small $I_2 - pieces$ in a way that their sizes never exceed 1.

Refined Harmonic partitioning is as follows. $(0,1] = \cup_{k=1}^{M} I_k \cup J_a \cup J_b$ where $J_1 = (1-\Delta, 1]$, $J_a = \left(\frac{1}{2}, 1-\Delta\right]$, $J_2 = \left(\Delta, \frac{1}{2}\right]$, $J_b = \left(\frac{1}{3}, \Delta\right]$, $J_k = (\frac{1}{k+1}, \frac{1}{k}]$, $3 \le k < M-1$ and $J_M = (0, \frac{1}{M}]$ where M is an integer. The packing strategy for $J_k - pieces$ for $1 \le k \le M$ are the same as Harmonic$_M$, but $J_a - pieces$ and $J_b - pieces$ will be mixed together. Figure 1 represents the intervals of the Refined Harmonic and how Δ is placed so $J_a - pieces$ and $J_b - pieces$ can be mixed.

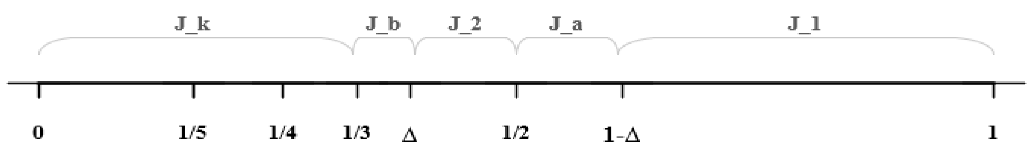

Figure 1 Intervals in Refined Harmonic

In Harmonic$_M$ $I_k - pieces$ were batched in $I_k - bins$, Refined Harmonic works the same for $J_k - pieces$, however for $I_a - pieces$ and $I_b - pieces$, can form different bins. There are 4 possibilities for these pieces.

1) two $J_b - pieces$ in a $J_{bb} - bin$,
2) one $J_a - piece$ in a $J_a - bin$,
3) one $J_a - piece$ and one $J_b - piece$ in a $J_{ab} - bin$,
4) only one $J_b - piece$ in a $J_b - bin$.

A $J_b - bin$ will change to a $J_{bb} - bin$ or a $J_{ab} - bin$ if a $J_b - piece$ or a $J_a - piece$ is added to the bin. The goal is to create $J_{ab} - bins$ or $J_{bb} - bins$ whenever possible.

We use a modification of Refined Harmonic to develop a new online algorithm. The partitioning is based on Refined Harmonic, however rules for assigning a job to a bin are different in the proposed algorithm.

$J_k - pieces$ for $1 \leq k < M - 1$ are packed in $J_k - bins$.

$J_a - pieces$ are packed in $J_a - bins$ if there is no $J_b - bin$ available, otherwise they are added to the $J_b - bin$ and the bin changes to a $J_{ab} - bin$.

$J_b - pieces$ are packed in $J_b - bins$ if there are no $J_b - bin$ or $J_a - bin$ available, otherwise if there is a $J_a - bin$ available it should be added to that and the bin changes to a $J_{ab} - bin$. If not, the job is added to the $J_b - bin$ and the bin changes to $J_{bb} - bin$.

$J_M - pieces$ are added to any available bin based on Next Fit algorithm[1], and if no such a bin exists a new $J_1 - bin$ is created and the job is added to the bin. Since $J_M - pieces$ are the smallest jobs, instead of creating a new type of bin for them they can be added to the remaining capacity of other bins to reduce the number of bins and wait-time of the jobs that are already in bins for the same type of jobs to be added. Given all the above descriptions about the Harmonic algorithms, the proposed algorithm is as follows.

[1] Next Fit algorithm adds the job to the next available bin that fits the jobs and if there is no bin available it opens a new bin.

Refined Harmonic Algorithm

For each Job j in L:

1. Identify the job's class based on its size.
2. Sort all existing batches based on decreasing order of their wait-time, in case of tie sort based on decreasing order of full capacity.
3. If job is a $J_k - pieces$:
   a. If a batch of the same class $J_k - bin$ exists AND job fits in the batch:
      i. Add the job to the batch, and update batch size and time.
   b. If there is no available $J_k - bin$:
      i. Create a new $J_k - bin$ and add the job to the batch, and update batch size and time.
4. If job is a $J_a - pieces$:
   a. If a batch of the class $J_b - bin$ exists AND job fits in the batch:
      i. Add the job to the batch, and update batch size and time.
      ii. Change the batch type to $J_{ab} - bin$.
   b. If there is no available batch:
      i. Create a new $J_a - bin$ and add the job to the batch, and update batch size and time.
5. If job is a $J_b - pieces$:
   a. If a batch of the class $J_a - bin$ exists AND job fits in the batch:
      i. Add the job to the batch, and update batch size and time.
      ii. Change the batch type to $J_{ab} - bin$.
   b. Else if a batch of the class $J_b - bin$ exists AND job fits in the batch:
      i. Add the job to the batch, and update batch size and time.
      ii. Change the batch type to $J_{bb} - bin$.
   c. If there is no available batch:
      i. Create a new $J_b - bin$ and add the job to the batch, and update batch size and time.
6. If job is a $J_M - pieces$:
   a. Use Next Fit to add the job to an existing batch.
   b. If no batch exists:
      i. Create a new $J_1 - bin$ and add the job to the batch, and update batch size and time

End For

It is crucial to first determine the right M and Δ for the algorithm. These parameters should be determined based on the problem description and job sizes in RMD sets scheduling. Since job sizes follow a discrete uniform distribution between 1 to 36, $s_j \sim U[1,36]$ we know that for list $L' = \{a_1, a_2, a_3, \ldots, a_n\}$ where $a_i \in (0,1]$ for all i. We can have $a_i = \frac{s_j}{36}$, therefore for each job i we calculate a_i to determine the class of the job. Additionally, we saw that we have $M + 2$ intervals for the Refined Harmonic Algorithm with $M > 2$. Considering different options for M we can see that based on the job size, it is only logical that $M \in [3,10]$, so there is at least one job in each interval. However, for the cases where M equals 6 to 10 there exists some intervals with only one job. For example, if $M = 8$, I_7, I_6 and I_5 will only have one job in them. Moreover, if M equals 3 or 4 the last interval, I_M, will include the majority of jobs. Therefore, M is chosen to be 5.

After determining M we need to determine Δ to create the intervals. We know that $\Delta \in (\frac{1}{3}, \frac{1}{2})$, therefore $1 - \Delta \in (\frac{1}{2}, \frac{2}{3})$. Considering $M = 5$ applying the same approach used for determining M, we can see that Δ should be a fraction of $\frac{1}{36}$ to create meaningful I_a and I_b intervals. Δ should be placed between I_2 and I_3 in a way that intervals I_a and I_b both include the maximum number of job sizes. We know that $\Delta \in (\frac{1}{3}, \frac{1}{2})$ or in other words $\Delta \in (\frac{12}{36}, \frac{18}{36}]$ which means we have 6 choices for Δ, it's clear that other fractions between these choices do not change the results, therefore, since we should obtain maximum number of jobs in I_a and I_b we can say that Δ should be $\frac{15}{36}$, to have 3 jobs in I_a and 3 jobs in I_b. It is possible to choose other fraction for Δ however it will result in having fewer jobs in one of the intervals and more in another, but there will be always a total of 6 jobs in both intervals.

4.3. Preliminary Test Results

4.3.1. Test Instances

In this section, each algorithm is tested with a series of same test instances and the results are compared. The test instances are created based on the data gathered in the literature (Di Mascolo and Gouin, 2013). There are usually 1 to 4 washing machines in the sterilization department and the have capacity of 6 DNI and a washing duration of 60 minutes. Job sizes and job release dates have a discrete uniform distribution. Job sizes are multiples of $\frac{1}{36}$ of machine capacity, therefore, $s_j \sim U[1,36]$. Jobs have interarrival times from 0 to 40 minutes which also has a uniform distribution, $Interarrival\ Time \sim U[0,40]$. 16 simple test instances are generated to test the algorithms. Different sets of jobs are tested in instances, the number of jobs in the instances varies from 10 to 25. The makespan of the mathematical model is used as the optimal solution, and the gap reported in Table 2 for each algorithm is calculated using $(C_{max}(Alg) - C_{max}(MILP)) * 100/C_{max}(MILP)$, where $C_{max}(Alg)$ is the solution of the algorithm and $C_{max}(MIP)$ is the solution of the MILP model.

A computer with core-i5 CPU and 8 GB RAM was used for the computational experiment. The MILP model was solved using CPLEX 12.1 and algorithms were coded and solved with Java 8.1. Table 1 represents the resolution time of the MILP model based on the number of jobs and machines. Table 2 shows the performance of the developed algorithms compared to the optimal solution.

Table 1 Resolution Time for MILP model

No. of jobs	No. of machines	Res. Time (sec)
10	1	0.03 sec
10	2	0.05 sec
10	3	0.11 sec
10	4	0.13 sec
15	1	0.22 sec
15	2	0.23 sec
15	3	0.20 sec
15	4	0.39 sec
20	1	0.08 sec
20	2	0.33 sec
20	3	0.70 sec
20	4	0.95 sec
25	1	0.25 sec
25	2	2.95 sec
25	3	1.58 sec
25	4	1.58 sec

Table 2 Comparison between the MILP model and Algorithms

		Offline		Online		
No. of jobs	No. of machines	%FFD-Gap	%BFD-Gap	%FF-Gap	%BF-Gap	%RH-Gap
10	1	5.06	0.00	35.19	35.19	51.90
15	1	11.13	11.13	62.26	62.26	46.42
20	1	12.13	12.13	34.10	34.10	35.74
25	1	9.62	9.62	25.64	39.36	41.03
10	2	0.00	0.00	0.00	0.00	49.38
15	2	9.17	9.17	33.33	33.33	33.33
20	2	11.82	11.82	38.41	37.27	23.64
25	2	9.04	9.04	38.88	38.88	30.38
10	3	0.00	0.00	0.00	0.00	9.09
15	3	0.00	0.00	0.00	17.48	3.15
20	3	0.00	0.00	0.00	3.31	0.00
25	3	0.00	0.00	29.64	29.64	7.32
10	4	0.00	0.00	0.00	0.00	0.00
15	4	0.00	0.00	0.00	2.48	0.00
20	4	2.18	2.18	4.12	7.51	2.18
25	4	0.00	0.00	0.00	5.06	0.00

4.3.2. Performance of the proposed algorithms

As represented in Table 1 by increasing the number of jobs the resolution time of the MILP model increases. Therefore, for greater job numbers the MILP model is not able to find the optimal solution in a reasonable time. However, the algorithms can find near-optimal solutions in a short time compared to the MILP model. FFD and BFD algorithms have quite better performance in obtaining near-optimal solutions compared to FF and BF algorithms. For some test instances, the BF algorithm shows slightly different results compared to FF, however RH algorithm finds better results in most cases compared to both BF and FF algorithms. Moreover, Table 2 shows how increasing the number of machines will affect the performance of the online algorithms. Therefore, we can see that without implementing any closing rules and just scheduling based on machines and jobs' availability RH algorithm has the best performance in most cases among the online algorithms.

4.4. A Special Case for Offline Batch Scheduling

The previous part of this chapter mainly focused on solving the problem of minimizing the makespan, in the following section, we focus on minimizing the total completion time of the jobs in the same settings.

In this section we focus on a specific problem where the jobs have divisible job sizes. Based on (Coffman, Garey and Johnson, 1987), strongly divisible and weakly divisible sequence are defined as follows.

Item sizes form a divisible sequence when the sequence,

$$S = \{s_1 > s_2 > s_3 > \cdots > s_i > s_{i+1} > \cdots\}$$

is such that for all $i \geq 1$ s_{i+1} exactly divides s_i. A list of jobs L and batch capacity of B are weakly divisible if the job sizes in L form a divisible sequence, and they are strongly divisible if in addition to the divisible sequence the largest job size, s_1 , also divides the batch capacity, B. For example, the following two instances show a strongly divisible and a weakly divisible sequence of jobs respectively.

$$S_1 = \{2,4,8,8,16\}, \qquad B = 32$$

$$S_2 = \{2,4,8,8,16\}, \qquad B = 20$$

Where S_1 and S_2 are sets of job sizes and B is the capacity of the batch.

The first approach for minimizing the total completion time of jobs is heuristic algorithms.

(Coffman, Garey and Johnson, 1987) proved that when the jobs have a strongly divisible sequence FF, FFD and FFI algorithms always produce the optimal number of bins. Considering this theorem, we can prove that with a strongly divisible sequence the FFI algorithm[2] always finds the optimal solution for minimum completion times.

Theorem: FFI is optimal for minimizing $\sum C_j$ when all jobs are available at the same time and job sizes follow a strongly divisible sequence.

Proof: In other words, to minimize the completion times in a bin packing problem we can penalize the jobs based on their bin index which leads to putting more jobs in the lower indexed bins. It is clear that the optimal solution for the *min* $\sum C_j$ problem is to add as many as jobs in the first bin, therefore, sorting the jobs in an increasing order of job sizes is the first step. We know that FFI produces the minimum number of bins when the jobs have a strongly divisible (SD) sequence (Coffman, Garey and Johnson, 1987) , thus, there is no better solution that puts more jobs in the lower indexed bins and creates the same number of bins simultaneously, and we can conclude that FFI algorithm is optimal for *min* $\sum C_j$ with a SD sequence. □

[2] FFI algorithm works the same as First Fit algorithm with the additional step of first sorting the jobs in an increasing order of sizes.

However, we cannot make the same conclusion for the WD sequence, since the assumption of FFI creating the minimum number of bins is not correct anymore.

For solving a more general problem based on the above problem where we have a fixed number of job sizes and the number of jobs of each size is known we present a dynamic programming algorithm to minimize the total completion times.

Suppose we have r different values for the job sizes, and we can have many jobs with the same sizes. Therefore, we have $\{ s_i : 1 < i < r\}$.

$L[x_1, x_2, \ldots, x_r]$ is a list of jobs to be batched in a form that x_i represents the number of jobs with size s_i for $1 \leq i \leq r$. B and M represent the batch capacity and the number of machines respectively.

Let $f(x_1, x_2, \ldots, x_r, b)$ be the dynamic programming function for $\sum_{j=1}^{n} C_j$ to pack x_i items of size s_i respectively in the bin with index b, where n is the total number of jobs in list L. The dynamic programming algorithm is as follows.

Dynamic Programming Algorithm

```
Sort list L based on increasing order of sizes.
Initialize: for all entities f(x_1, x_2, ..., x_r, b) = ∞;
f(0) = 0;
for x_1 ← 0 to n_1 do
 | for x_2 ← 0 to n_2 do
 |  | ...
 |  | for x_r ← 0 to n_r do
 |  |  | for y_1 ← 0 to x_1 do
 |  |  |  | for y_2 ← 0 to x_2 do
 |  |  |  |  | ...
 |  |  |  |  | for y_r ← 0 to x_r do
 |  |  |  |  |  | for b ← 0 to n do
 |  |  |  |  |  |  | if Σ_{i=1}^{r} ((x_i − y_i) * s_i) ≤ B
 |  |  |  |  |  |  |  | f(x_1, x_2, ..., x_r, b) = min { f(x_1, x_2, ..., x_r, b), f(y_1, y_2, ..., y_r, b − 1) + (Σ_{i=1}^{r} (x_i − y_i)) * p * ⌈b/M⌉}
 |  |  |  |  |  |  | End if
 |  |  |  |  |  | End for
 |  |  |  |  | End for
 |  |  |  |  | ...
 |  |  |  | End for
 |  |  | End for
 |  | End for
 |  | ...
 | End for
End for
```

The complexity of this algorithm with r distinct sizes and n_i jobs for each size when $1 \leq i \leq r$ is $\prod_{i=1}^{r} \frac{(n_i+1)(n_i+2)}{2}$ which is in order of $O(n^{2r})$. The complexity reaches its maximum when the jobs are divided equally among all distinct sizes. For instance, if we have 10 jobs to be scheduled with 3 distinct sizes, the complexity is maximum when the three size groups include 3,3,4 jobs.

Chapter 5

Simulation

In this chapter, the sterilization cycle is simulated to validate the result of the heuristic algorithms. As we have seen in chapter 1 the sterilization cycle includes the following steps, pre-disinfection, rinsing, washing, verification, packaging, and sterilization.

So far, we have developed optimization algorithms to minimize the makespan of the washing step, however these algorithms do not consider the effect of the scheduling methods on the whole system. The goal of this chapter is to demonstrate how these algorithms impact the whole system.

5.1. Simulation Models

The sterilization cycle is simulated as a stochastic discrete event simulation model. In other words, the interarrival time of the RMDs and their sizes are considered to be stochastic and the events, such as the start of the washing process, are considered to occur at discrete times. Anylogic software is used as the simulation tool. Since the sterilization cycle includes different processes the Process Modeling Library (PML) in Anylogic is used to build the model. Each step in the sterilization cycle is simulated as an individual process.

A simplified model of the sterilization cycle is simulated and tested to compare the results. This model consists of three main steps; washing, conditioning, and sterilization. We can simplify the system to this point since the other steps do not have a significant impact on RMDs flow time or $Cmax$. Figure 2 represents the simplified model and Figure 3 is the actual simulation model used for each algorithm in AnyLogic, each step of the simulation model is captured in this figure. The description of these steps are as follows.

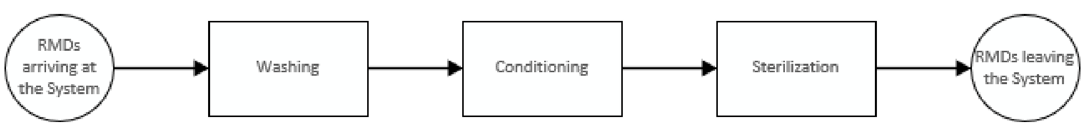

Figure 2 The simplified sterilization processes

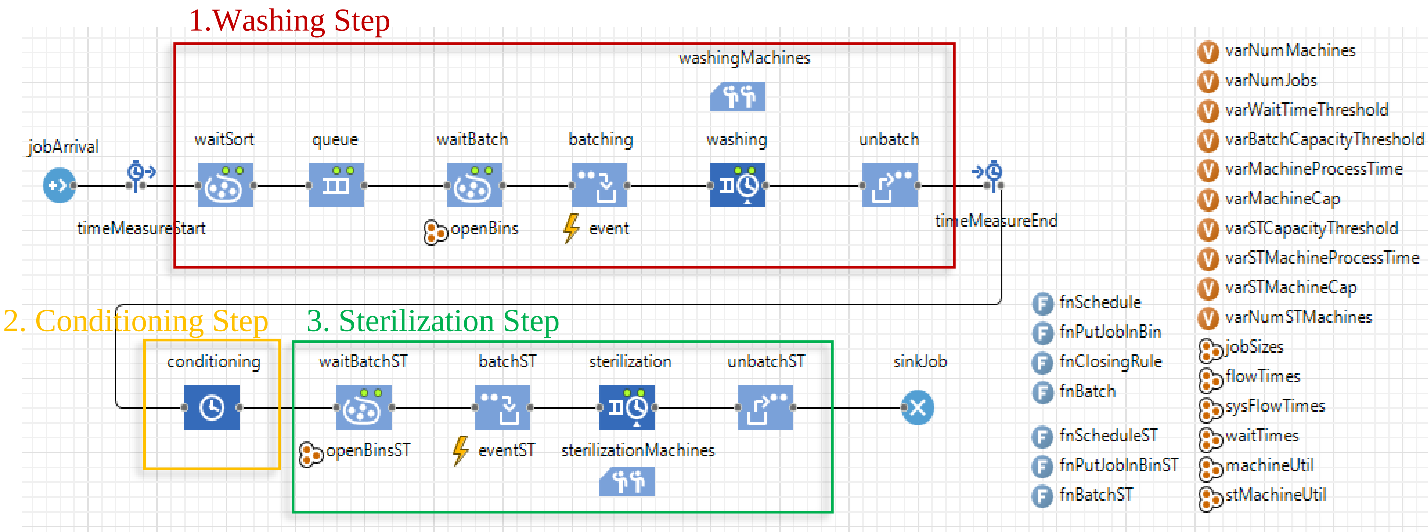

Figure 3 Complete model, Sterilization Cycle

Washing: this step is the bottleneck of the system, and the proposed algorithms will be implemented on this step in the simulation model. In this step, RMDs are washed using specific washing machines. As mentioned before, washing step can have 1 to 4 washing machines with a capacity of 6 DIN. The duration of washing is constant and equal to 60 minutes. The machines cannot be interrupted after they have started the washing process.

Figure 4 represents the washing step flow chart in the simulation model. In this flow chart jobs first arrive in the *jobArrival* bloc based on the inter-arrival data, their size and arrival date are assigned to them at the time of the arrival. Jobs that enter the system when no washing machine is available wait at the *waitSort* bloc and when a washing machine becomes available the jobs in this bloc enter the *queue* and get sorted based on decreasing order of their sizes. At *waitBatch* jobs are batched based on different algorithms and send to *batching* bloc for grouping the jobs. They enter the *washing* bloc which uses washing machine resources and jobs get processed in this step. Finally, they are unbatched (*unbatch*) and leave the washing step (*sinkJob*).

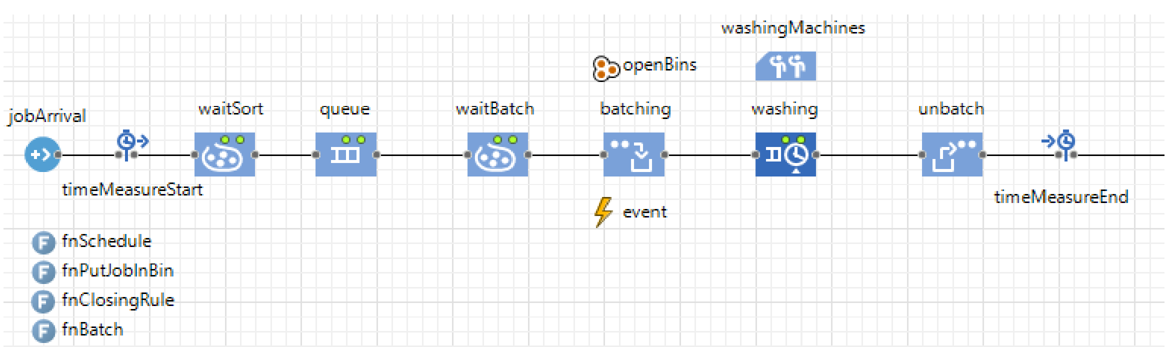

Figure 4 Simulation model, washing step flow chart

Conditioning: this step is a combination of the simplified steps of the sterilization cycle (Figure 5). In this step, RMD sets that are unbatched in the previous step (*unbatch*) get packed and ready to enter the sterilization step (each RMD set is packed separately). The duration of the conditioning step follows a discrete uniform distribution for each RMD set ($cd_j \sim U[4,6]$).

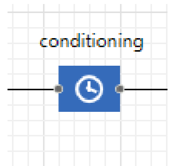

Figure 5 Simulation model, conditioning step flow chart

Sterilization: this step is very similar to the washing step, only the machine capacities are double the washing machines in the washing step, therefore, this step cannot increase the cycle time of the RMDs and create a bottleneck. In this step, RMDs are packed together again to become sterilized by the sterilization machines (autoclaves). The sterilization step has a constant duration of 105 minutes (Di Mascolo and Gouin, 2013). Similar to closing rules in the washing step we have closing rules in this step, to start the machines at certain points. However, because this step is not a bottleneck the main assumption in this step is using FIFO for batching RMDs.

The flow chart of the sterilization step, Figure 6, is similar to the washing step. RMDs enter the *waitBatchST* bloc after finishing conditioning and wait for the autoclaves to become available. Then in the *batchST* bloc they are batched and batches are sent to the *sterilization* bloc to be sterilized by the autoclaves (*sterilizationMachines* represents the autoclaves in the flow chart, we consider that there are 3 autoclaves in this step). Finally, the RMDs are unbatched in the *unbatchST* unit and exit the simulation flow chart (*sinkJob*).

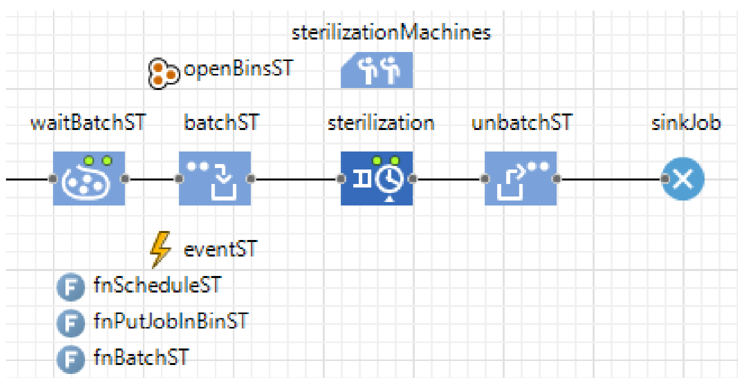

Figure 6 Simulation model, sterilization step flow chart

The above information described the processes that RMDs go through in the simulation model. RMDs are modeled as entities that move in the system. The sizes and interarrival dates of these entities follow the discrete uniform distribution the same as what we had in the heuristics ($s_j \sim U[1,36]$, $Interarrival\ Time \sim U[0,40]$). The sterilization steps are modeled as processes that process each entity based on their duration and pre-defined entering and exiting rules.

Flow chart represented in Figure 3 is used for all the simulation models, however the code in each bloc is changed based on the algorithm used in each model.

To demonstrate that the washing step is the bottleneck, FIFO is used as a batching approach for the first simulation model and considered as the status quo of the system. After simulating the current system, heuristic algorithms proposed in chapter 4 are simulated and their results are analyzed. To demonstrate that the algorithms are online the RMDs' data will be only known at the time of their arrival in each step. Implementation of the algorithms are possible through code blocs in AnyLogic.

Rest of this chapter focuses on implementing different batching algorithms in both washing step and sterilization step. Since the batch scheduling algorithms can also be use in the sterilization step, we develop different models using pairs of algorithms for these two steps. For the washing step the following algorithms are use; Best Fit, First Fit, First Fit with no thresholds, Refined Harmonic, Refined Harmonic with no thresholds and FIFO. All of the previous algorithms except two Refined Harmonics are used for the sterilization step.

5.1.1. FIFO Model

This model is the simplest version of batch scheduling used for RMD set scheduling. RMDs are packed together only based on their arrival time, and sent to the washing machines when there is a machine available and the next job does not fit the batch, in other words, the algorithm waits for the summation of the job sizes to become more than the batch capacity then batches the jobs. (Di Mascolo and Gouin, 2013) showed that in this model the washing step is the bottleneck.

5.1.2. No Wait Model

This model works based on first fit algorithm. It batches every job that arrived at the system in the first open batch that fits. The closing rule in this model is only the availability of the machines (this model does not consider any thresholds).

5.1.3. First Fit Model

First Fit model batches the jobs whenever a washing machine is available, and if there is no washing machine available the jobs will wait in the *waitSort* bloc and they are sorted based on decreasing order of their sizes. This model has two types of thresholds, capacity thresholds and wait time thresholds. Whenever the three closing rules, two thresholds and machine availability, are satisfied the batch is closed and ready for washing.

5.1.4. Best Fit Model

This model works the same as First Fit model only with one improvement on the batching strategy, each job is batched in the batch with the smallest remaining space.

5.1.5. Refined Harmonic

In the Refined Harmonic model is that RMDs are classified at *jobArrival* bloc using a function called *fnClassification* based on the determined intervals (refer to chapter 4 for more information) and the jobs are batched together based on their groups. The rest of the algorithm is the same as Best Fit and First Fit.

5.1.6. No Wait Refined Harmonic

This model is the same as the Refined Harmonic model but without any thresholds.

All of these algorithms can be implemented in the sterilization step with the same setting only the washing machines are replaced with sterilization machines.

5.2. Simulation Results

5.2.1. Test Instances

Input parameters for algorithms include arrival times, job sizes, number of jobs, number of machines, capacity threshold, and wait-time threshold. Capacity threshold is the limit for scheduling a batch based on its full capacity, for example, if the capacity threshold is 90% any batch with a size of more than 90% of the batch capacity is ready to be scheduled on a machine. Wait-time threshold represents the maximum time that a batch waits before being scheduled, if batch wait-time exceeds the wait-time threshold the batch is ready to be scheduled.

Table 3 represents the values for the input parameters. Based on capacity threshold and wait-time threshold values, used in Best Fit, First Fit, Refined Harmonic, 9 input types are created and each input type is tested with 100 test instances. Test instances can have 10, 20, 30, 40, or 50 jobs and 1, 2, 3, or 4 machines, each combination of the number of jobs and number of machines is tested 5 times for each input type. These thresholds are determined intuitively based on the cost of water and energy for each cycle of the machines and the acceptable amount of waiting for RMDs.

Table 3 Input Types

Input Type	Capacity Threshold (% of batch capacity)	Wait-Time Threshold (minutes)
1	70%	10
2	80%	10
3	90%	10
4	70%	20
5	80%	20
6	90%	20
7	70%	30
8	80%	30
9	90%	30

After running all 900 instances on 11 different simulation models for the instances with the same input parameters mean of the objective functions was calculated and used for the analysis. Each simulation run had five main outputs including Cmax, Cycle Cmax, Average Flow Time, Average Cycle Flow Time, and Average Wait-Time. The detailed numerical results of the simulation models are represented in Tables 5 to 9. These tables include the improvement in the washing step and sterilization cycle, when number of jobs equals to 50, compared to FIFO model, one table is provided specifically for each objective. Moreover, the extreme detailed results are found in the Appendix, each objective value is compared to the FIFO model and the improvement in objective values is presented as minutes decreased for each objective in the tables when the number of jobs equals 30, 40, and 50.

Table 4 represents the synthesis of the results of the 11 simulation models. First, the results are grouped based on the number of washing machines, due to the important role of the number of washing machine in the duration of the sterilization cycle and creating a bottleneck, then for each number of machines, the best algorithms including the best pair of wait-time threshold and capacity threshold are presented (wait-time threshold is in minutes and capacity threshold is the percentage of washing machine's capacity). The results with objective values (Cmax, Average Flow Time, Average Wait Time, Cycle Cmax and, Average Cycle Flow Time) of 1% above the minimum of each run were considered as best solutions. Based on the results of the simulation models we can see that for sterilization services with 1 or 2 washing machines implementing algorithms with thresholds can improve our objectives. However, when the number of washing machines increases to 3 or 4 in general the *No-Wait* algorithm or in other words, the First Fit algorithm without any wait-time and capacity thresholds seems to have a better performance. These results are close to our expectation since considering the interarrival time of RMDs, with more washing machines the probability of machines availability when a batch is ready is higher so adding thresholds will only keep the ready batches waiting while machines are available and increases Cmax and flow time compared to the *No-Wait* algorithm.

In addition to representing the algorithms with the best results, Table 4 shows the average decrease for each objective compared to the FIFO model in minutes. This table consists of two main sections, the washing step, and the Sterilization cycle. Three objectives are considered for the washing step including Cmax, Average Flow Time and, Average Wait Time, and two for the sterilization cycle; Cycle Camx and, Average Cycle Flow Time. For each objective, a column is provided with the algorithm finding the best results and another column for the difference between the objective of the FIFO algorithm and the proposed algorithm. Results are categorized based on the Number of Machines (Column 1) and for each row, all the algorithms finding the best results are presented. For instance, for 2 machines FF (10, 0.8), the First Fit algorithm with a wait-time threshold of 10 minutes and a capacity threshold of 80% of machine capacity, can decrease the washing step's Cmax on average 46.85 minutes compared to results of FIFO.

Table 4 Synthesis of the results of Simulation Models

	Washing Step						***Sterilization Cycle***			
Number of Machines	Cmax	Average decrease in Cmax (minutes)	Average Flow Time	Average decrease in Avg Flow Time (minutes)	Average Wait Time	Average decrease in Avg Wait Time (minutes)	Cycle Cmax	Average decrease in Cycle Cmax (minutes)	Average Cycle Flow Time	Average decrease in Avg Cycle Flow Time (minutes)
1	FF (10, 0.9) BF (10, 0.9) No Wait	236.2 236.2 194.85	FF (30, 0.8) BF (30, 0.8) No Wait	169.87 169.87 152.25	FF (30, 0.8) BF (30, 0.8) No Wait	169.87 169.87 152.26	BF-BF (10, 0.9) BF-No Wait (10, 0.9) BF-FIFO (10, 0.9) FF-BF (10, 0.9) FF-No Wait (10, 0.9) FF-FIFO (10, 0.9) No Wait-No Wait No Wait-FIFO	236.20 232.60 236.20 236.20 235.27 236.20 141.27 141.27	FF-FF (20, 0.7) BF-BF (20, 0.7) BF-No Wait (20, 0.9) FF- No Wait (20, 0.9) No Wait- No Wait	167.60 167.60 172.27 172.73 133.43
2	FF (10, 0.8) BF (10, 0.8) No Wait	46.85 46.85 38.25	FF (10, 0.9) BF (10, 0.9) No Wait	26.58 26.58 30.77	FF (10, 0.9) BF (10, 0.9) No Wait	24.31 24.31 34.55	BF-FIFO (10, 0.8) FF-FIFO (10, 0.8) No Wait-FIFO	47.75 47.75 35.12	FF-FF (20, 0.7) BF-BF (20, 0.7) BF-No Wait (20, 0.9) FF- No Wait (20, 0.9) No Wait-No Wait	26.03 26.03 34.00 26.05 32.38
3	No Wait	8.96	No Wait	24.2	No Wait	24.2	No Wait- FIFO	11.40	No Wait-No Wait Wait RH -No Wait	22.37 21.25
4	No Wait No Wait RH	1.64 1.64	No Wait No Wait RH	26.28 26.25	No Wait No Wait RH	26.28 26.25	No Wait- FIFO	10.68	No Wait-No Wait No Wait RH -No Wait	23.21 23.69

Table 5 Numerical results, comparison of proposed algorithms and FIFO model for Cmax (50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
Type 1 (10, 0.7)	**1**	249.8	249.8	233.2	30.2	1.4
	2	61	61	63.2	-16	-26.8
	3	2.8	2.8	-0.6	-1	-8.8
	4	-4	-4	-0.4	-4	-0.4
Type 2 (10, 0.8)	**1**	308.2	308.2	268	62.8	48.4
	2	99.2	99.2	80.6	-50.8	-18.4
	3	2.6	2.6	-3.6	-11.8	-11.6
	4	-5.8	-5.8	-6.6	-7.6	-4.4
Type 3 (10,0.9)	**1**	293.2	293.2	278.8	37	22.6
	2	29.8	29.8	26.4	-36	-43
	3	-10.8	-10.8	-6.4	-13.8	-8.4
	4	-7.4	-7.4	0.6	-7.4	0.6
Type 4 (20, 0.7)	**1**	267.8	267.8	261.8	97	91
	2	57.8	57.8	66.2	-26.2	-29.6
	3	-1	-1	4.2	-8.6	0.4
	4	-12.6	-12.6	-0.8	-7.2	0.2
Type 5 (20, 0.8)	**1**	307.2	307.2	299.8	87.6	80.2
	2	73.4	73.4	76.2	-1.2	-25.6
	3	-4.8	-4.8	4.6	-9.2	-1.4
	4	-7	-7	-2.2	-13	-2.2
Type 6 (20, 0.9)	**1**	302	302	279.8	80.8	48
	2	39.4	39.4	53.4	-12.4	-22
	3	-3.6	-3.6	10.4	-11	-3.2
	4	-10.8	-10.8	-3.2	-9.6	-3.2
Type 7 (30, 0.7)	**1**	345	345	304.6	125.4	109.4
	2	50.2	50.2	46.4	-9.4	-21.6
	3	-6.8	-6.8	7.4	-9.8	1.8
	4	-5.6	-5.6	-8.2	-17.8	-8.2
Type 8 (30, 0.8)	**1**	324.2	324.2	293.4	116.8	73.8
	2	37.8	37.8	29	-28.8	-28.6
	3	4.6	4.6	7.6	-25.6	-19.8
	4	-2.4	-2.4	0.6	-12.6	0.6
Type 9 (30, 0.9)	**1**	252.2	252.2	243.6	55	72.8
	2	60	60	46.2	-23.6	-5.8
	3	-18.6	-18.6	-5	-20.6	-4.8
	4	-23.4	-23.4	2	-24.6	0.4

Table 6 Numerical results, comparison of proposed algorithms and FIFO model for in Average Flow Time (50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
Type 1 (10, 0.7)	**1**	213.37	213.37	205.31	102.1	87.7
	2	36	36	36.46	8	-0.2
	3	21.54	21.54	20.06	16.34	13.34
	4	20.24	20.24	25.12	19.79	24.64
Type 2 (10, 0.8)	**1**	236.71	236.71	222.86	132.83	133.56
	2	64.99	64.99	56.96	-9.18	7.85
	3	19.06	19.06	21.38	15.96	15.04
	4	22.6	22.6	28.07	21.84	27.84
Type 3 (10,0.9)	**1**	197.66	197.66	200.34	111.52	99.56
	2	35.66	35.66	36.32	2.91	4.83
	3	18.87	18.87	24.58	16.89	23.68
	4	19.7	19.7	26.04	18.97	25.92
Type 4 (20, 0.7)	**1**	223.39	223.39	224.96	165.56	157.37
	2	44.6	44.6	51.21	-5.84	4.78
	3	16.3	16.3	23.76	13.31	21.33
	4	14.95	14.95	23.67	12.6	22.84
Type 5 (20, 0.8)	**1**	210.82	210.82	212.69	131.28	125.83
	2	38	38	39.64	0.76	1.75
	3	15.33	15.33	23.55	11.6	20.12
	4	13.97	13.97	25.34	11.37	25.06
Type 6 (20, 0.9)	**1**	214.31	214.31	208.22	123.9	104.76
	2	24.12	24.12	27.8	-2.65	0.58
	3	14.28	14.28	24.96	8.86	19.24
	4	9.5	9.5	21.98	7.52	21.84
Type 7 (30, 0.7)	**1**	253.26	253.26	242.62	159.08	153.32
	2	34.29	34.29	39.22	4.68	10.92
	3	13.44	13.44	20.77	8.82	17.66
	4	10.78	10.78	20.72	6.82	20.22
Type 8 (30, 0.8)	**1**	239.72	239.72	236.98	161.64	149.14
	2	19.87	19.87	16.65	-0.78	-4.46
	3	9.73	9.73	20.68	-1.16	12.54
	4	8.88	8.88	21.73	4.44	20.93
Type 9 (30, 0.9)	**1**	203.22	203.22	208.52	111.18	124.1
	2	32.8	32.8	36.65	-3.3	8.25
	3	7.18	7.18	21.83	3.43	20.9
	4	5.76	5.76	23.32	2.92	22.97

Table 7 Numerical results, comparison of proposed algorithms and FIFO model for in Average Wait Time (50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
Type 1 (10, 0.7)	**1**	213.37	213.37	205.31	102.1	87.7
	2	36	36	36.46	8	-0.2
	3	21.54	21.54	20.75	16.34	16.2
	4	20.24	20.24	25.12	19.79	24.64
Type 2 (10, 0.8)	**1**	236.71	236.71	222.86	132.83	133.56
	2	64.99	64.99	56.96	12.9	7.85
	3	19.06	19.06	21.38	15.96	15.04
	4	22.6	22.6	28.07	21.84	27.84
Type 3 (10,0.9)	**1**	197.66	197.66	200.34	111.52	99.56
	2	35.66	35.66	36.32	2.91	4.83
	3	18.87	18.87	24.58	16.89	23.68
	4	19.7	19.7	26.04	18.97	25.92
Type 4 (20, 0.7)	**1**	223.39	223.39	224.96	165.56	157.37
	2	44.6	44.6	51.21	-5.84	4.78
	3	16.3	16.3	23.76	13.31	21.33
	4	15.19	15.19	23.67	12.6	22.84
Type 5 (20, 0.8)	**1**	210.82	210.82	212.69	131.28	125.83
	2	38	38	39.64	0.76	1.75
	3	15.33	15.33	23.55	11.6	20.12
	4	14.14	14.14	25.34	11.54	25.06
Type 6 (20, 0.9)	**1**	214.31	214.31	208.22	123.9	104.76
	2	24.12	24.12	27.8	-2.65	0.58
	3	14.28	14.28	24.96	8.86	19.24
	4	9.58	9.58	21.98	7.52	21.84
Type 7 (30, 0.7)	**1**	253.26	253.26	242.62	159.08	153.32
	2	34.29	34.29	39.22	4.68	10.92
	3	13.68	13.68	20.77	8.91	17.66
	4	10.87	10.87	20.75	6.91	20.28
Type 8 (30, 0.8)	**1**	239.72	239.72	236.98	161.64	149.14
	2	19.87	19.87	16.65	-0.78	-4.46
	3	9.73	9.73	20.68	-1.16	12.54
	4	9.04	9.04	21.73	4.57	20.93
Type 9 (30, 0.9)	**1**	203.22	203.22	208.52	111.18	124.1
	2	32.8	32.8	36.65	-3.3	8.25
	3	7.32	7.32	21.83	3.57	20.9
	4	5.76	5.76	23.32	2.92	22.97

Table 8 Numerical results, comparison of proposed algorithms and FIFO-FIFO model for Cycle Cmax (50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
Type 1 (10, 0.7)	**1**	250.6	250.6	250.2	250.6	250.6	250	234	234	30	1
	2	42.4	61.8	45.8	42.4	61.8	43.2	63.6	41.2	-35.4	-47.4
	3	-16.6	9.2	-7	-16.6	9.2	-16.4	4.2	-16.2	-21.6	-29.2
	4	4.8	7	4.8	4.8	7	5.4	19	9	5.6	9.6
Type 2 (10, 0.8)	**1**	309	309	308.6	309	309	307.4	268.8	268.8	62	48
	2	82.4	101	88.4	82.4	101	88.4	82.8	65.6	-60.2	-40.8
	3	3.4	24	3.8	3.4	24	4.2	19.6	7.8	-4.8	-4.2
	4	-17	11	-11.8	-17	11	-16.4	6.6	-10.8	-16.4	-10.8
Type 3 (10,0.9)	**1**	293.2	293.2	292.6	293.2	293.2	292.4	279.2	279.2	36.6	22.2
	2	19	28.8	18.2	19	28.8	6	25	10.2	-62.2	-60.2
	3	-37.6	-10.4	-36.4	-37.6	-10.4	-36.2	-11	-27.6	-44.8	-37.6
	4	-20.2	-3.2	-28.2	-20.2	-3.2	-28.8	5.2	-19	-28.6	-18.6
Type 4 (20, 0.7)	**1**	269.4	269.4	268.4	269.4	269.4	268.4	263.4	263.4	98	92
	2	40.6	56.6	39	40.6	56.6	39.8	66.2	51.8	-37.2	-41.6
	3	-16.8	-1	-16.2	-16.8	-1	-15.2	4.6	-14.6	-37.2	-18.6
	4	-16.8	-5	-11.2	-16.8	-5	-13.8	-1	-7.4	-15.6	-4.4
Type 5 (20, 0.8)	**1**	307.6	307.6	306.8	307.6	307.6	307.2	300.6	300.6	87.4	80.2
	2	55	73	51	55	73	52	76.4	53.2	-13	-31.8
	3	-5.4	6.6	-1	-5.4	6.6	-2	14	0.2	-16.8	-4.6
	4	-19.4	-9.6	-23.8	-19.4	-9.6	-20.6	-1	-16.8	-23.2	-16.8
Type 6 (20, 0.9)	**1**	302.4	302.4	301.8	302.4	302.4	302.4	280.2	280.2	80.4	47.8
	2	22.6	38.2	21.8	22.6	38.2	22.6	49.6	34.6	-36	-43.4
	3	-14.4	-7.4	-15.2	-14.4	-7.4	-11	4	2	-18.2	-14.8
	4	-5.6	7.6	-6.2	-5.6	7.6	-5.6	8.2	-10.4	-2.8	-10.2
Type 7 (30, 0.7)	**1**	345.4	345.4	344.4	345.4	345.4	345	305	305	125.2	109.4
	2	32	47.4	32.4	32	47.4	32.2	42	23.8	-17.4	-41.4
	3	-10	2.8	-0.8	-10	2.8	0.8	15.4	-4.6	-8.6	-10.4
	4	-2.4	0.8	-3.6	-2.4	0.8	2.6	4.2	-7.6	-17.2	-6
Type 8 (30, 0.8)	**1**	325	325	324	325	325	325	294.6	294.6	117.8	74
	2	20.2	37	18	20.2	37	17.2	24.2	12.8	-45.6	-39.6
	3	6	9	3.8	6	9	8.2	16	11.8	-21.8	-20.6
	4	13.2	7.6	7.8	13.2	7.6	2.2	17.6	5.6	-6	-1
Type 9 (30, 0.9)	**1**	252.2	252.2	251.4	252.2	252.2	252.4	244	244	54.6	71.8
	2	47.8	60.4	36.4	47.8	60.4	47.4	47	26.6	-32.2	-20.2
	3	-20.4	-4.2	-18.2	-20.4	-4.2	-17.8	8.4	-16.4	-20.4	-16
	4	-37.2	-22.8	-29.2	-37.2	-22.8	-35	4	-11.2	-38.2	-10.8

Table 9 Numerical results, comparison of proposed algorithms and FIFO-FIFO model for Average Cycle Flow Time (50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
Type 1 (10, 0.7)	**1**	215.76	213.34	214.2	215.76	213.34	215.64	204.83	207.51	104.45	90.42
	2	36.05	34.38	36.52	36.05	34.38	35.81	36.36	36.37	8.26	-2.37
	3	19.73	16.46	19.42	19.73	16.46	19.24	14.69	18.85	14.12	12.87
	4	15.96	12.68	16.56	15.96	12.68	16.11	18.69	21.11	16.71	20.51
Type 2 (10, 0.8)	**1**	237.51	235.31	238.76	237.51	235.31	237.39	221.47	223.93	133.66	134.92
	2	64.36	64.77	64.65	64.36	64.77	64.36	57.33	56.51	-7.56	6.75
	3	15.93	10.21	15.9	15.93	10.21	14.41	12.63	16.97	13.05	10.36
	4	16.95	15.73	16.74	16.95	15.73	15.98	20.5	21.72	15.27	21.6
Type 3 (10,0.9)	**1**	199.2	196	199.92	199.2	196	199.82	199.72	201.92	113.45	101.37
	2	33.81	33.06	35.02	33.81	33.06	33.32	32.71	34	0.73	2.63
	3	17.96	12.8	17.54	17.96	12.8	17.91	17.28	22.74	15.64	21.1
	4	15.52	12.38	15.19	15.52	12.38	15.84	17.37	21.9	14.12	21.67
Type 4 (20, 0.7)	**1**	224.3	221.95	222.6	224.3	221.95	223.13	224.05	226.04	166.65	158.96
	2	41.8	41.77	41.86	41.8	41.77	42.21	49.12	49.01	-7.82	1.78
	3	15.22	8.09	14.19	15.22	8.09	14.81	13.18	19.44	10	15.71
	4	14.05	12.37	14.72	14.05	12.37	14.36	16.73	19.14	10.4	19.1
Type 5 (20, 0.8)	**1**	213.3	210.76	211.47	213.3	210.76	211.87	212.32	214.98	133.4	127.32
	2	35.98	34.64	36.36	35.98	34.64	36.34	35.38	38.41	-0.09	0.89
	3	15.87	11.33	15.63	15.87	11.33	15.48	18.62	20.75	8.8	17.46
	4	10.33	7.01	11.05	10.33	7.01	10.96	16.69	19.85	7.38	20.77
Type 6 (20, 0.9)	**1**	217.35	215.1	217.18	217.35	215.1	216.93	208.91	211.4	126.5	108.19
	2	24.25	22.4	23.9	24.25	22.4	24.13	25.27	26.71	-2.69	-0.93
	3	11.57	10.89	12.86	11.57	10.89	11.58	18.44	21.34	5.72	15.78
	4	8.55	7.59	8.85	8.55	7.59	8.78	16.49	19.62	4.43	19.63
Type 7 (30, 0.7)	**1**	255.64	252.98	254.47	255.64	252.98	254.84	241.85	244.15	160.6	155.16
	2	33.88	32.27	33.62	33.88	32.27	34.07	36.12	37.57	4.25	9.93
	3	13.36	12.07	13.06	13.36	12.07	14.38	13.62	18.14	6.75	13.99
	4	11.52	8.66	10.73	11.52	8.66	13.14	14.39	18.64	7.25	18.18
Type 8 (30, 0.8)	**1**	242.08	239.86	242.98	242.08	239.86	242.95	237.96	240.78	164.98	151.68
	2	21.13	17.29	18.94	21.13	17.29	19.13	11.79	14.06	-1.34	-6.14
	3	9.23	8.56	7.6	9.23	8.56	6.46	14.2	15.6	-6.11	7.45
	4	10.6	9.17	10.31	10.6	9.17	9.27	17.7	19.18	5.5	19.76
Type 9 (30, 0.9)	**1**	204.4	201.9	206.27	204.4	201.9	206	207.79	210.8	113.27	125.94
	2	33.22	32.11	32.74	33.22	32.11	32.94	33.76	35.29	-3.91	7.64
	3	7.8	6.44	8.64	7.8	6.44	6.45	13.52	16.26	-0.65	16.72
	4	6.08	4.28	7.3	6.08	4.28	5.49	17.31	20.81	1.5	20.75

Chapter 6

Conclusion and Future Work

This research is focused on improving the efficiency of the sterilization department in hospitals by minimizing the makespan in the washing step of the sterilization process. To accomplish this goal, a batch scheduling approach is used, and online and offline algorithms are developed. Two offline heuristics (FFD and BFD) and three online heuristics (FF, BF, RH) are developed and compared with the MILP model. The result of computational experiments has shown that the algorithms can find near-optimal solutions in a reasonable time and the results can be improved by defining more efficient scheduling rules. Finally, different simulation models are developed to add validity to the optimization methods and to simulate the entire sterilization cycle. Different closing rules are considered in the simulation models and tested to find the best solution. Two main closing rules considered in these models are the wait-time threshold and capacity threshold. Wait-time threshold is a closing rule based on the waiting time of each RMD in the washing step and Capacity Threshold closes the batches when an acceptable percentage of the batch is full, it is clear that these closing rules can be initiated only if a machine is available. The proposed algorithms are implemented in both the washing step and sterilization step of the models and different combinations of these algorithms are tested. The final results represent the best combination of algorithms and closing rules based on the number of machines.

The results are categorized based on the number of existing washing machines in the washing step, therefore, managerial decisions can be made for hospitals with different sizes. The results represent that for sterilization services with 1 or 2 washing machines it is more beneficial to use the Best Fit and First Fit heuristics with wait-time and capacity thresholds, however for 3 or 4 washing machines it is better to only use simple First Fit without any thresholds, the Refined Harmonic algorithm without thresholds also demonstrate good performance in the case of 4 washing machines.

Moreover, we solved a special case of this problem with divisible job sizes and proposed a dynamic programming model to find optimal solutions for minimizing the total completion time.

The result of this research demonstrates how simple heuristics can improve the duration of the sterilization cycle and the wait-time of the RMDs in the washing step. One of the main factors that is considered in developing these heuristics is the ability to be implemented easily by the staff in the sterilization services. Therefore, sterilization departments can increase their performance and lower their cycle time by using the heuristics with proper thresholds based on the number of washing machines available in their department.

Future work can include considering different closing rules for the models. Also, a combination of Refined Harmonic algorithm and a simple algorithm like First Fit can be developed to test when the RH algorithm finds better solutions. It is important to notice that in this research we specifically focused on developing algorithms for the washing step and as an extension to our research these algorithms were also tested on the sterilization step, thus, one way to extend this research is to focus on improving the sterilization step after the bottleneck in the washing step is removed and develop new simple and easy to implement heuristics specifically for the sterilization step.

The outcome of this research can be implemented in hospitals' sterilization departments directly to both enhance their performance and reduce the wait-time. Moreover, it can be beneficial in policymaking and as a decision-making tool for managers to improve the system.

References

Abedi, M. *et al.* (2015) 'Bi-objective optimisation for scheduling the identical parallel batch-processing machines with arbitrary job sizes, unequal job release times and capacity limits', *International Journal of Production Research*, 53(6), pp. 1680–1711. doi: 10.1080/00207543.2014.952795.

Arakawa, M., Fuyuki, M. and Inoue, I. (2003) 'An optimization-oriented method for simulation-based job shop scheduling incorporating capacity adjustment function', *International Journal of Production Economics*, 85(3), pp. 359–369. doi: 10.1016/S0925-5273(03)00122-1.

Arroyo, J. E. C. and Leung, J. Y. T. (2017a) 'An effective iterated greedy algorithm for scheduling unrelated parallel batch machines with non-identical capacities and unequal ready times', *Computers and Industrial Engineering*, 105, pp. 84–100. doi: 10.1016/j.cie.2016.12.038.

Arroyo, J. E. C. and Leung, J. Y. T. (2017b) 'Scheduling unrelated parallel batch processing machines with non-identical job sizes and unequal ready times', *Computers and Operations Research*, 78, pp. 117–128. doi: 10.1016/j.cor.2016.08.015.

Azizoglu, M. and Webster, S. (2000) 'Scheduling a batch processing machine with non-identical job sizes', *International Journal of Production Research*, 38(10), pp. 2173–2184. doi: 10.1080/002075400050028034.

Azizoglu, M. and Webster, S. (2001) *Scheduling a batch processing machine with incompatible job families*. Available at: www.elsevier.com/locate/dsw (Accessed: 18 June 2020).

Cabrera, E. *et al.* (2011) 'Optimization of healthcare emergency departments by agent-based simulation', *Procedia Computer Science*, 4, pp. 1880–1889. doi: 10.1016/j.procs.2011.04.204.

Chai, X., Li, W. and Zhu, Y. (2019) 'Online scheduling to minimize maximum weighted flow-time on a bounded parallel-batch machine', *Annals of Operations Research*. doi: 10.1007/s10479-019-03352-6.

Chang, P. Y., Damodaran, P. and Melouk, S. (2004) 'Minimizing makespan on parallel batch processing machines', *International Journal of Production Research*, 42(19), pp. 4211–4220. doi: 10.1080/00207540410001711863.

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), pp. 942–956. doi: 10.1080/0951192X.2010.495137.

Chen, H., Du, B. and Huang, G. Q. (2011) 'Scheduling a batch processing machine with non-identical job sizes: A clustering perspective', *International Journal of Production Research*, 49(19), pp. 5755–5778. doi: 10.1080/00207543.2010.512620.

Chen, P. S. *et al.* (2015) 'Scheduling patients' appointments: Allocation of healthcare service using simulation optimization', *Journal of Healthcare Engineering*, 6(2), pp. 259–280. doi: 10.1260/2040-2295.6.2.259.

Chen, X. *et al.* (2016) 'Scheduling on parallel identical machines with late work criterion: Offline and online cases', *Journal of Scheduling*, 19(6), pp. 729–736. doi: 10.1007/s10951-015-0464-7.

Cheng, B. *et al.* (2012) 'Minimizing makespan and total completion time for parallel batch processing machines with non-identical job sizes', *Applied Mathematical Modelling*, 36(7), pp. 3161–3167. doi: 10.1016/j.apm.2011.09.061.

Chung, S. H., Tai, Y. T. and Pearn, W. L. (2009) 'Minimising makespan on parallel batch processing machines with non-identical ready time and arbitrary job sizes', *International Journal of Production Research*, 47(18), pp. 5109–5128. doi: 10.1080/00207540802010807.

Coffman, E. G., Garey, M. R. and Johnson, D. S. (1987) 'Bin packing with divisible item sizes', *Journal of Complexity*, 3(4), pp. 406–428. doi: 10.1016/0885-064X(87)90009-4.

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), pp. 1119–1128. doi: 10.1007/s00170-009-2457-1.

Damodaran, Purushothaman *et al.* (2011) 'A GRASP approach for makespan minimization on parallel batch processing machines', *J Intell Manuf*, 22, pp. 767–777. doi: 10.1007/s10845-009-0272-z.

Deng, X., Poon, C. K. and Zhang, Y. (2003) 'Approximation Algorithms in Batch Processing', *Journal of Combinatorial Optimization*, 7(3), pp. 247–257. doi: 10.1023/A:1027316504440.

Dias, L. S. *et al.* (2018) 'A simulation-based optimization framework for integrating scheduling and model predictive control, and its application to air separation units', *Computers and Chemical Engineering*, 113, pp. 139–151. doi: 10.1016/j.compchemeng.2018.03.009.

Doleschal, D., Klemmt, A. and Weigert, G. (2011) 'Iterative simulation-based optimization for parallel batch scheduling problems', *Proceedings of the International Spring Seminar on Electronics Technology*, pp. 374–379. doi: 10.1109/ISSE.2011.6053891.

Dupont, L. and Dhaenens-Flipo, C. (2002) *Minimizing the makespan on a batch machine with non-identical job sizes: an exact procedure*, *Computers & Operations Research*.

Dupont, L. and Ghazvini, F. (1998) *Minimizing makespan on a single batch processing machine with non-identical job sizes.*, *Eur J Autom Syst*. Available at: https://pascal-francis.inist.fr/vibad/index.php?action=getRecordDetail&idt=1683483 (Accessed: 18 June 2020).

Epstein, L., Johnson, D. S. and Levin, A. (2018) 'Min-Sum Bin Packing', *Journal of Combinatorial Optimization*, 36(2), pp. 508–531. doi: 10.1007/s10878-018-0310-x.

Fang, Y. and Lu, X. (2016) 'Online parallel-batch scheduling to minimize total weighted completion time on single unbounded machine', *Information Processing Letters*, 116(8), pp. 526–531. doi: 10.1016/j.ipl.2016.03.010.

Fang, Y., Lu, X. and Liu, P. (2011) 'Online batch scheduling on parallel machines with delivery times', *Theoretical Computer Science*, 412(39), pp. 5333–5339. doi: 10.1016/j.tcs.2011.06.011.

Freitag, M. and Hildebrandt, T. (2016) 'Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization', *CIRP Annals* -

Manufacturing Technology, 65(1), pp. 433–436. doi: 10.1016/j.cirp.2016.04.066.

Fu, R. *et al.* (2007) 'Online scheduling in a parallel batch processing system to minimize makespan using restarts', *Theoretical Computer Science*, 374(1–3), pp. 196–202. doi: 10.1016/j.tcs.2006.12.040.

Ghazvini, F. J. and Dupont, L. (1998) *Minimizing mean flow times criteria on a single batch processing machine with non-identical jobs sizes, Int. J. Production Economics*.

Ham, A., Fowler, J. W. and Cakici, E. (2017) 'Constraint Programming Approach for Scheduling Jobs with Release Times, Non-Identical Sizes, and Incompatible Families on Parallel Batching Machines', *IEEE Transactions on Semiconductor Manufacturing*, 30(4), pp. 500–507. doi: 10.1109/TSM.2017.2740340.

Husseinzadeh Kashan, A., Karimi, B. and Jolai, F. (2010) 'An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with non-identical job sizes', *Engineering Applications of Artificial Intelligence*, 23(6), pp. 911–922. doi: 10.1016/j.engappai.2010.01.031.

Johnson, D. S. (1973) 'Near-Optimal Bin Packing Algorithms', *Massachusetts Institute of Technology*, p. 400.

Kashan, A. H., Karimi, B. and Jenabi, M. (2008) 'A hybrid genetic heuristic for scheduling parallel batch processing machines with arbitrary job sizes', *Computers & Operations Research*, 35, pp. 1084–1098. doi: 10.1016/j.cor.2006.07.005.

Kempf, K. G. (1998) *Scheduling a Single Batch Processing Machine with Secondary Resource Constraints, Journal of Manufacturing Systems*.

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), pp. 219–236. doi: 10.1080/002075499192020.

Lee, D. T. (1985) 'A Simple On-Line Bin-Packing Algorithm', 32(3), pp. 562–572.

Lee, Yoon Ho and Lee, Young Hoon (2013) 'Minimising makespan heuristics for scheduling a single batch machine processing machine with non-identical job sizes', *International Journal of Production Research*, 51(12), pp. 3488–3500. doi: 10.1080/00207543.2012.748226.

Li, S. *et al.* (2005) 'Operations Research Letters Minimizing makespan on a single batching machine with release times and non-identical job sizes', *Operations Research Letters*, 33, pp. 157–164. doi: 10.1016/j.orl.2004.04.009.

Li, S. (2017) 'Approximation algorithms for scheduling jobs with release times and arbitrary sizes on batch machines with non-identical capacities', *European Journal of Operational Research*, 263(3), pp. 815–826. doi: 10.1016/j.ejor.2017.06.021.

Li, W. H. and Chai, X. (2018) 'Online Scheduling on Bounded Batch Machines to Minimize the Maximum Weighted Completion Time', *Journal of the Operations Research Society of China*, 6(3), pp. 455–465. doi: 10.1007/s40305-017-0179-x.

Li, W. and Li, S. (2015) 'Online batch scheduling of equal-length jobs on two identical batch machines to maximise the number of early jobs', *International Journal of Systems Science*, 46(4), pp. 652–661. doi: 10.1080/00207721.2013.794904.

Liu, H., Lu, X. and Li, W. (2020) 'A best possible online algorithm for parallel batch scheduling with delivery times and limited restart', *Optimization Letters*. doi: 10.1007/s11590-020-01618-8.

Liu, P. and Lu, X. (2014) 'Online unbounded batch scheduling on parallel machines with delivery times', *Journal of Combinatorial Optimization*, 29(1), pp. 228–236. doi: 10.1007/s10878-014-9706-4.

Liu, Z. and Yu, W. (2000) 'Scheduling one batch processor subject to job release dates', *Discrete Applied Mathematics*, 105(1–3), pp. 129–136. doi: 10.1016/S0166-218X(00)00181-5.

Lu, S., Feng, H. and Li, X. (2010) 'Minimizing the makespan on a single parallel batching machine', *Theoretical Computer Science*, 411(7–9), pp. 1140–1145. doi: 10.1016/j.tcs.2009.12.008.

Lucidi, S. *et al.* (2016) 'A derivative-free approach for a simulation-based optimization problem in healthcare', *Optimization Letters*, 10(2), pp. 219–235. doi: 10.1007/s11590-015-0905-4.

Ma, R. *et al.* (2014) 'Online bounded-batch scheduling to minimize total weighted completion time on parallel machines', *International Journal of Production Economics*, 156, pp. 31–38. doi: 10.1016/j.ijpe.2014.05.012.

Di Mascolo, M. and Gouin, A. (2013) 'A generic simulation model to assess the performance of sterilization services in health establishments', *Health Care Management Science*, 16(1), pp. 45–61. doi: 10.1007/s10729-012-9210-2.

Melouk, S., Damodaran, P. and Chang, P.-Y. (2004) 'Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing', *Int. J. Production Economics*, 87, pp. 141–147. doi: 10.1016/S0925-5273(03)00092-6.

Nong, Q. Q., Ng, C. T. and Cheng, T. C. E. (2008) 'Operations Research Letters The bounded single-machine parallel-batching scheduling problem with family jobs and release dates to minimize makespan', *Operations Research Letters*, 36, pp. 61–66. doi: 10.1016/j.orl.2007.01.007.

Ozturk, O. *et al.* (2012a) 'Makespan minimisation on parallel batch processing machines with non-identical job sizes and release dates', *International Journal of Production Research*, 50(20), pp. 6022–6035. doi: 10.1080/00207543.2011.641358.

Ozturk, O. *et al.* (2012b) 'Makespan minimisation on parallel batch processing machines with non-identical job sizes and release dates', *International Journal of Production Research*, 50(20), pp. 6022–6035. doi: 10.1080/00207543.2011.641358.

Ozturk, O. (2019) 'A bi-criteria optimization model for medical device sterilization', *Annals of Operations Research*. doi: 10.1007/s10479-019-03296-x.

Ozturk, O., Begen, M. A. and Zaric, G. S. (2017) 'A branch and bound algorithm for scheduling unit size jobs on parallel batching machines to minimize makespan', *International Journal of Production Research*, 55(6), pp. 1815–1831. doi: 10.1080/00207543.2016.1253889.

Potts, C. N. and Kovalyov, M. Y. (2000) 'Scheduling with batching: a review', *European Journal* *of Operational Research*. Elsevier Science B.V., pp. 228–249. doi: 10.1016/S0377-2217(99)00153-8.

Rafiee Parsa, N., Karimi, B. and Husseinzadeh Kashan, A. (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), pp. 1720–1730. doi: 10.1016/j.cor.2009.12.007.

Selvarajah, E., Steiner, G. and Zhang, R. (2013) 'Single machine batch scheduling with release times and delivery costs', *J Sched*, 16, pp. 69–79. doi: 10.1007/s10951-011-0255-8.

Soleimani, H. *et al.* (2020) 'Scheduling of unrelated parallel machines considering sequence-related setup time, start time-dependent deterioration, position-dependent learning and power consumption minimization', *Journal of Cleaner Production*, 249, p. 119428. doi: 10.1016/j.jclepro.2019.119428.

Tian, J., Fu, R. and Yuan, J. (2014) 'Online Over Time Scheduling on Parallel-Batch Machines: A Survey', *Journal of the Operations Research Society of China*, 2(4), pp. 445–454. doi: 10.1007/s40305-014-0060-0.

Uzsoy, R. (1994) 'Scheduling a single batch processing machine with non-identical job sizes', *International Journal of Production Research*, 32(7), pp. 1615–1635. doi: 10.1080/00207549408957026.

Wang, H.-M. 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*, 37, pp. 1510–1521. doi: 10.1016/j.eswa.2009.06.070.

Zhang, G., Cai, X. and Wong, C. K. (2001) 'On-line algorithms for minimizing makespan on batch processing machines', *Naval Research Logistics*, 48(3), pp. 241–258. doi: 10.1002/nav.5.

Zhang, X. *et al.* (2006) *APPROXIMATION SCHEMES FOR SCHEDULING A BATCHING MACHINE WITH NONIDENTICAL JOB SIZE **.

Zhang, Y. *et al.* (2012) 'A simulation optimization approach to long-term care capacity planning', *Operations Research*, 60(2), pp. 249–261. doi: 10.1287/opre.1110.1026.

Appendix

Section 1

This section represents the detailed numerical results for the washing step. A table is provided for each objective, i.e., Cmax, Average Flow Time, and Average Wait Time. Each table has 8 columns consisting of input parameters for each test; Input Type (Wait Time Threshold, Capacity Threshold), Number of Jobs, Number of Machines, and 5 algorithms tested on the washing step; Best Fit, First Fit, No wait, Refined Harmonic and, Refined Harmonic No wait. The average test results are represented only for 30, 40, and 50 jobs and 1 to 4 machines. The numerical results are the difference between the objective of the FIFO algorithm and the objective of the proposed algorithm in minutes. For example, for Best Fit in the Cmax table (Table 10) the results are calculated as follows: $Cmax_{FIFO} - Cmax_{BF}$.

Table 10 Numerical results, comparison of proposed algorithms and FIFO model for Cmax (30, 40, 50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
Type 1 (10, 0.7)	30	1	134.4	134.4	107.8	-24.2	-38.6
		2	28.6	28.6	23.6	-12.8	-20.8
		3	-10.8	-10.8	-6	-15.2	-9.2
		4	-4.6	-4.6	0.8	-6.4	0.8
	40	1	203.8	203.8	201.6	20.8	30.8
		2	45	45	46	2.2	-3.8
		3	-12.8	-12.8	-7.6	-12.8	-11.6
		4	-5.4	-5.4	0.2	-7.2	0.2
	50	1	249.8	249.8	233.2	30.2	1.4
		2	61	61	63.2	-16	-26.8
		3	2.8	2.8	-0.6	-1	-8.8
		4	-4	-4	-0.4	-4	-0.4
Type 2 (10, 0.8)	30	1	155	155	114.2	20.8	28.8
		2	29.2	29.2	33.2	-25	-23.8
		3	-2.6	-2.6	-0.6	-12	-9.6
		4	-8.4	-8.4	0.6	-8.4	0.6
	40	1	212.8	212.8	183.8	-19	-23.6
		2	38.6	38.6	43.6	-28	-36
		3	1.8	1.8	1.4	1	0.8
		4	-3	-3	0.6	-3	0.6
	50	1	308.2	308.2	268	62.8	48.4
		2	99.2	99.2	80.6	-50.8	-18.4

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
		3	2.6	2.6	-3.6	-11.8	-11.6
		4	-5.8	-5.8	-6.6	-7.6	-4.4
Type 3 (10,0.9)	**30**	**1**	201.6	201.6	162.8	-5.8	-32.4
		2	37	37	32	-19.6	-22
		3	-8.4	-8.4	-3	-10.2	-5.2
		4	-6.8	-6.8	0.4	-6.8	0.4
	40	**1**	213.8	213.8	175	6.4	16.4
		2	28.8	28.8	21	-26.6	-24
		3	-5	-5	2.6	-6.4	2.6
		4	-0.4	-0.4	1.6	-3	1.6
	50	**1**	293.2	293.2	278.8	37	22.6
		2	29.8	29.8	26.4	-36	-43
		3	-10.8	-10.8	-6.4	-13.8	-8.4
		4	-7.4	-7.4	0.6	-7.4	0.6
Type 4 (20, 0.7)	**30**	**1**	179.8	179.8	149.6	-17	-33.4
		2	25.4	25.4	28	3.2	-1.4
		3	-7.2	-7.2	3.2	-11.8	-3.6
		4	-2.2	-2.2	1.4	-2.4	1.4
	40	**1**	213.8	213.8	189.2	6.4	6.2
		2	40.8	40.8	24.8	-27.8	-30.2
		3	-6.8	-6.8	1.8	-19.4	-6.4
		4	-10.8	-10.8	0.6	-14.6	0.6
	50	**1**	267.8	267.8	261.8	97	91
		2	57.8	57.8	66.2	-26.2	-29.6
		3	-1	-1	4.2	-8.6	0.4
		4	-12.6	-12.6	-0.8	-7.2	0.2
Type 5 (20, 0.8)	**30**	**1**	180.6	180.6	172.2	9.8	1.4
		2	7.6	7.6	2.8	-43.4	-41.2
		3	-10.8	-10.8	-2.2	-18.8	-12.4
		4	-9.2	-9.2	3.2	-12	3.2
	40	**1**	264	264	250.6	80	67.6
		2	47.6	47.6	52.4	-31.4	-35.4
		3	-2.4	-2.4	4.8	-1.2	-3.2
		4	-10.4	-10.4	-2.4	-15.4	-2.4
	50	**1**	307.2	307.2	299.8	87.6	80.2
		2	73.4	73.4	76.2	-1.2	-25.6
		3	-4.8	-4.8	4.6	-9.2	-1.4
		4	-7	-7	-2.2	-13	-2.2

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
Type 6 (20, 0.9)	30	1	104.2	104.2	99.8	-5.6	14.4
		2	28.4	28.4	22.8	-19.6	-14.2
		3	-3.2	-3.2	6.2	-3.2	6.2
		4	-12.6	-12.6	0.4	-12.6	-0.8
	40	1	234.6	234.6	177.4	88.2	55.4
		2	40.2	40.2	38.2	-11	-8
		3	-8.2	-8.2	-3.2	-17.4	-3.2
		4	-13.6	-13.6	-0.4	-18.4	-0.4
	50	1	302	302	279.8	80.8	48
		2	39.4	39.4	53.4	-12.4	-22
		3	-3.6	-3.6	10.4	-11	-3.2
		4	-10.8	-10.8	-3.2	-9.6	-3.2
Type 7 (30, 0.7)	30	1	160.4	160.4	139.8	-37	-55.4
		2	38.4	38.4	27.4	-32.8	-29.4
		3	-9	-9	1	-24.6	-9.2
		4	-17.8	-17.8	-1.6	-17.8	-1.6
	40	1	215.6	215.6	208	6.4	12.8
		2	46.6	46.6	48	-14.2	-7.6
		3	-11.4	-11.4	-1.2	-13.6	-1.2
		4	-21.2	-21.2	-3	-21.2	-3
	50	1	345	345	304.6	125.4	109.4
		2	50.2	50.2	46.4	-9.4	-21.6
		3	-6.8	-6.8	7.4	-9.8	1.8
		4	-5.6	-5.6	-8.2	-17.8	-8.2
Type 8 (30, 0.8)	30	1	118.8	118.8	117.6	-52	-53.2
		2	-4.8	-4.8	-10	-33	-43.6
		3	-0.2	-0.2	2.4	-8.4	-2.8
		4	-15.2	-15.2	0.4	-15.2	0.4
	40	1	260	260	225.8	-10.4	-18.2
		2	23.6	23.6	35	-31.4	-30.8
		3	-5.4	-5.4	2.8	-11	1.8
		4	-13.8	-13.8	-0.4	-16.4	-0.4
	50	1	324.2	324.2	293.4	116.8	73.8
		2	37.8	37.8	29	-28.8	-28.6
		3	4.6	4.6	7.6	-25.6	-19.8
		4	-2.4	-2.4	0.6	-12.6	0.6
Type 9 (30, 0.9)	30	1	188.2	188.2	158.4	89.2	85.2
		2	27.2	27.2	25.6	-39	-31.8

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
		3	-26.2	-26.2	1	-26.2	1
		4	-28.6	-28.6	0.4	-28.6	0.4
	40	**1**	283.4	283.4	258.6	76	87.8
		2	45.8	45.8	53.2	-34.2	-28.6
		3	-21.8	-21.8	-2.8	-14.4	-9.2
		4	-20.6	-20.6	0.6	-23.6	-3.8
	50	**1**	252.2	252.2	243.6	55	72.8
		2	60	60	46.2	-23.6	-5.8
		3	-18.6	-18.6	-5	-20.6	-4.8
		4	-23.4	-23.4	2	-24.6	0.4

Table 11 Numerical results, comparison of proposed algorithms and FIFO model for in Average Flow Time (30, 40, 50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
Type 1 (10, 0.7)	**30**	**1**	94.95	94.95	78.93	37.61	34.6
		2	29.65	29.65	30.65	8.42	4.98
		3	17.47	17.47	18.33	16.3	17.44
		4	20.62	20.62	24.88	19.21	24.27
	40	**1**	177.57	177.57	192.15	96.75	109.8
		2	32.08	32.08	33.53	9.94	7.7
		3	20.41	20.41	23.17	16.69	19.48
		4	19.61	19.61	24.44	18.18	24.12
	50	**1**	213.37	213.37	205.31	102.1	87.7
		2	36	36	36.46	8	-0.2
		3	21.54	21.54	20.06	16.34	13.34
		4	20.24	20.24	25.12	19.79	24.64
Type 2 (10, 0.8)	**30**	**1**	120.09	120.09	114.26	70.07	77.66
		2	23.5	23.5	24.84	3.69	7.02
		3	23.35	23.35	26.83	20.35	23.47
		4	19.65	19.65	24.74	18.35	24
	40	**1**	149.97	149.97	136.22	54.2	54.175
		2	23.87	23.87	26.11	-4.35	-6.18
		3	15.11	15.11	19.18	13.57	16.36
		4	19.65	19.65	24.16	18.18	23.32
	50	**1**	236.71	236.71	222.86	132.83	133.56

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
		2	64.99	64.99	56.96	-9.18	7.85
		3	19.06	19.06	21.38	15.96	15.04
		4	22.6	22.6	28.07	21.84	27.84
Type 3 (10,0.9)	**30**	**1**	122.71	122.71	108.71	44.63	39.17
		2	34.45	34.45	32.33	12.34	13.43
		3	16.56	16.56	20.07	11.23	15.49
		4	18.83	18.83	26.21	18.46	26.21
	40	**1**	167.44	167.44	160.36	93.02	97.84
		2	23.91	23.91	24.03	8.09	13.31
		3	18.35	18.35	19.38	12.49	13.71
		4	19.07	19.07	26.3	18.62	26.3
	50	**1**	197.66	197.66	200.34	111.52	99.56
		2	35.66	35.66	36.32	2.91	4.83
		3	18.87	18.87	24.58	16.89	23.68
		4	19.7	19.7	26.04	18.97	25.92
Type 4 (20, 0.7)	**30**	**1**	114.37	114.37	110.6	38.35	39.03
		2	26.13	26.13	20.99	9.25	6.47
		3	14.96	14.96	21.02	11.91	20.15
		4	17.61	17.61	26.84	15.83	26.84
	40	**1**	152.46	152.46	147.99	71.02	77.23
		2	24.12	24.12	22.35	-4.97	-5.27
		3	17.37	17.37	24.11	10.16	21.57
		4	15.31	15.31	24.14	12.42	23.44
	50	**1**	223.39	223.39	224.96	165.56	157.37
		2	44.6	44.6	51.21	-5.84	4.78
		3	16.3	16.3	23.76	13.31	21.33
		4	14.95	14.95	23.67	12.6	22.84
Type 5 (20, 0.8)	**30**	**1**	142.37	142.37	151.87	71.61	74.19
		2	17.93	17.93	19.88	-7.47	-1.2
		3	14.57	14.57	23.99	9.72	22.12
		4	15.12	15.12	26.33	13.03	26.33
	40	**1**	191.11	191.11	188.08	114.47	118.54
		2	29.17	29.17	34.03	-4	-0.85
		3	17.28	17.28	24.79	13.56	19.48
		4	14.46	14.46	24.36	11.59	23.52
	50	**1**	210.82	210.82	212.69	131.28	125.83
		2	38	38	39.64	0.76	1.75
		3	15.33	15.33	23.55	11.6	20.12

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
		4	13.97	13.97	25.34	11.37	25.06
Type 6 (20, 0.9)	**30**	**1**	111.88	111.88	115.61	65.93	83.89
		2	24.61	24.61	24.95	3.01	9.78
		3	12.83	12.83	21.74	9.77	20.59
		4	10.84	10.84	24.54	9.13	23.89
	40	**1**	176.71	176.71	167.7	125.17	113.11
		2	26.11	26.11	31.31	3.24	7.07
		3	14.96	14.96	24.41	9.85	22.65
		4	10.48	10.48	22.92	8.28	21.84
	50	**1**	214.31	214.31	208.22	123.9	104.76
		2	24.12	24.12	27.8	-2.65	0.58
		3	14.28	14.28	24.96	8.86	19.24
		4	9.5	9.5	21.98	7.52	21.84
Type 7 (30, 0.7)	**30**	**1**	129.49	129.49	124.35	31.73	19.83
		2	22.95	22.95	24.13	-5.73	-2.1
		3	15.36	15.36	24.57	6.65	22.09
		4	14.59	14.59	27.55	11.52	27.33
	40	**1**	170.46	170.46	185.43	87.85	98.51
		2	36.53	36.53	40.31	9.82	12.46
		3	13.17	13.17	22.02	8.12	19.32
		4	13.81	13.81	25.76	10.11	25.76
	50	**1**	253.26	253.26	242.62	159.08	153.32
		2	34.29	34.29	39.22	4.68	10.92
		3	13.44	13.44	20.77	8.82	17.66
		4	10.78	10.78	20.72	6.82	20.22
Type 8 (30, 0.8)	**30**	**1**	84.41	84.41	80.77	24.23	10.01
		2	17.37	17.37	17.09	-19.92	-2.53
		3	12.69	12.69	23.25	4.61	18.61
		4	8.19	8.19	22.59	3.28	21.21
	40	**1**	185.49	185.49	176.9	73.98	74.42
		2	20.09	20.09	26.34	-0.02	1.91
		3	12.41	12.41	25.26	8.92	24.67
		4	8.85	8.85	23.17	5.5	22.73
	50	**1**	239.72	239.72	236.98	161.64	149.14
		2	19.87	19.87	16.65	-0.78	-4.46
		3	9.73	9.73	20.68	-1.16	12.54
		4	8.88	8.88	21.73	4.44	20.93
Type 9 (30, 0.9)	**30**	**1**	131.28	131.28	118.95	88.4	83.17

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
		2	17.16	17.16	25.09	-14.52	-1.84
		3	7.85	7.85	25.84	1.99	23.52
		4	6.63	6.63	25.42	4.7	25.21
	40	1	192.82	192.82	198.52	115.35	135.69
		2	33.82	33.82	39.83	-7.13	5.25
		3	6.35	6.35	20.38	-1.11	14.37
		4	4.54	4.54	23.8	1.48	23.41
	50	1	203.22	203.22	208.52	111.18	124.1
		2	32.8	32.8	36.65	-3.3	8.25
		3	7.18	7.18	21.83	3.43	20.9
		4	5.76	5.76	23.32	2.92	22.97

Table 12 Numerical results, comparison of proposed algorithms and FIFO model for in Average Wait Time (30, 40, 50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
Type 1 (10, 0.7)	30	1	94.95	94.95	78.93	37.61	34.6
		2	29.65	29.65	30.65	8.42	4.98
		3	17.47	17.47	18.33	16.3	17.44
		4	20.62	20.62	24.88	19.21	24.27
	40	1	177.57	177.57	192.15	96.75	109.8
		2	32.08	32.08	33.53	9.94	7.7
		3	20.41	20.41	23.17	16.69	19.48
		4	19.61	19.61	24.44	18.18	24.12
	50	1	213.37	213.37	205.31	102.1	87.7
		2	36	36	36.46	8	-0.2
		3	21.54	21.54	20.75	16.34	16.2
		4	20.24	20.24	25.12	19.79	24.64
Type 2 (10, 0.8)	30	1	120.09	120.09	114.26	70.07	77.66
		2	23.5	23.5	24.84	3.69	7.02
		3	23.35	23.35	26.83	20.35	23.47
		4	19.65	19.65	24.74	18.35	24
	40	1	149.97	149.97	136.22	54.2	54.18
		2	23.87	23.87	26.11	-4.35	-6.18
		3	15.11	15.11	19.18	13.57	16.36
		4	19.65	19.65	24.16	18.18	23.32

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
	50	1	236.71	236.71	222.86	132.83	133.56
		2	64.99	64.99	56.96	12.9	7.85
		3	19.06	19.06	21.38	15.96	15.04
		4	22.6	22.6	28.07	21.84	27.84
Type 3 (10,0.9)	30	1	122.71	122.71	108.71	44.63	39.17
		2	34.45	34.45	32.33	12.34	13.43
		3	16.56	16.56	20.07	11.23	15.49
		4	19	19	26.21	18.46	26.21
	40	1	167.44	167.44	160.36	93.02	97.84
		2	23.91	23.91	24.03	8.09	13.31
		3	18.35	18.35	19.38	12.91	13.71
		4	19.07	19.07	26.3	18.62	26.3
	50	1	197.66	197.66	200.34	111.52	99.56
		2	35.66	35.66	36.32	2.91	4.83
		3	18.87	18.87	24.58	16.89	23.68
		4	19.7	19.7	26.04	18.97	25.92
Type 4 (20, 0.7)	30	1	114.37	114.37	110.6	38.35	39.03
		2	26.13	26.13	20.99	9.25	6.47
		3	14.96	14.96	21.02	11.91	20.15
		4	17.61	17.61	26.84	15.83	26.84
	40	1	152.46	152.46	147.99	71.02	77.23
		2	24.12	24.12	22.35	-4.97	-5.27
		3	17.37	17.37	24.11	10.39	21.57
		4	15.31	15.31	24.14	12.42	23.44
	50	1	223.39	223.39	224.96	165.56	157.37
		2	44.6	44.6	51.21	-5.84	4.78
		3	16.3	16.3	23.76	13.31	21.33
		4	15.19	15.19	23.67	12.6	22.84
Type 5 (20, 0.8)	30	1	142.37	142.37	151.87	71.61	74.19
		2	17.93	17.93	19.88	-7.47	-1.2
		3	14.57	14.57	23.99	9.72	22.12
		4	15.12	15.12	26.33	13.03	26.33
	40	1	191.11	191.11	188.08	114.47	118.54
		2	29.17	29.17	34.03	-4	-0.85
		3	17.28	17.28	24.79	13.56	21.01
		4	14.46	14.46	24.36	11.59	23.52
	50	1	210.82	210.82	212.69	131.28	125.83
		2	38	38	39.64	0.76	1.75

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
		3	15.33	15.33	23.55	11.6	20.12
		4	14.14	14.14	25.34	11.54	25.06
Type 6 (20, 0.9)	**30**	**1**	111.88	111.88	115.61	65.93	83.89
		2	24.61	24.61	24.95	3.01	9.78
		3	12.83	12.83	21.74	9.77	20.59
		4	10.84	10.84	24.54	9.13	23.89
	40	**1**	176.71	176.71	167.7	125.17	113.11
		2	26.11	26.11	31.31	3.24	7.07
		3	14.96	14.96	24.41	9.85	22.65
		4	10.48	10.48	22.92	8.28	21.84
	50	**1**	214.31	214.31	208.22	123.9	104.76
		2	24.12	24.12	27.8	-2.65	0.58
		3	14.28	14.28	24.96	8.86	19.24
		4	9.58	9.58	21.98	7.52	21.84
Type 7 (30, 0.7)	**30**	**1**	129.49	129.49	124.35	31.73	19.83
		2	22.95	22.95	24.13	-5.73	-2.1
		3	15.36	15.36	24.57	7.63	22.09
		4	14.59	14.59	27.55	11.52	27.33
	40	**1**	170.46	170.46	185.43	87.85	98.51
		2	36.53	36.53	40.31	9.82	12.46
		3	13.2	13.2	22.02	8.15	19.32
		4	13.81	13.81	25.76	10.11	25.76
	50	**1**	253.26	253.26	242.62	159.08	153.32
		2	34.29	34.29	39.22	4.68	10.92
		3	13.68	13.68	20.77	8.91	17.66
		4	10.87	10.87	20.75	6.91	20.28
Type 8 (30, 0.8)	**30**	**1**	84.41	84.41	80.77	24.23	10.01
		2	17.37	17.37	17.09	-12.31	-2.53
		3	12.69	12.69	23.25	4.61	18.61
		4	8.19	8.19	22.59	3.28	21.21
	40	**1**	185.49	185.49	176.9	73.98	74.42
		2	20.09	20.09	26.34	-0.02	1.91
		3	12.41	12.41	25.26	8.92	24.67
		4	8.85	8.85	23.17	5.58	22.73
	50	**1**	239.72	239.72	236.98	161.64	149.14
		2	19.87	19.87	16.65	-0.78	-4.46
		3	9.73	9.73	20.68	-1.16	12.54
		4	9.04	9.04	21.73	4.57	20.93

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	Best Fit	First Fit	No Wait	Refined Harmonic	Refined Harmonic No Wait
Type 9 (30, 0.9)	**30**	**1**	131.28	131.28	118.95	88.4	83.17
		2	17.16	17.16	25.09	-14.53	-1.84
		3	7.85	7.85	25.84	1.99	23.52
		4	6.63	6.63	25.42	4.7	25.21
	40	**1**	192.82	192.82	198.52	115.35	135.69
		2	33.82	33.82	39.83	-7.13	5.25
		3	6.35	6.35	20.38	-0.66	14.37
		4	4.54	4.54	23.89	1.56	23.49
	50	**1**	203.22	203.22	208.52	111.18	124.1
		2	32.8	32.8	36.65	-3.3	8.25
		3	7.32	7.32	21.83	3.57	20.9
		4	5.76	5.76	23.32	2.92	22.97

Section 2

This section presents the numerical results for the whole sterilization cycle; washing step, conditioning step and, sterilization step. Two tables are provided for Cycle Cmax and Average Cycle Flow Time. Each table has 13 columns consisting of input parameters; Input Type (Wait Time Threshold, Capacity Threshold), Number of Jobs, Number of Machines, and the proposed algorithms. Each column for algorithms represents first the algorithm used in the washing step, then the algorithm used in the sterilization step ($Alg_{washing} - Alg_{sterilization}$). The average test results are represented only for 30, 40, and 50 jobs and 1 to 4 machines. The numerical results are calculated based on the difference between the objective value of the FIFO-FIFO model and the proposed models. For example, in Table 14, column BF-BF shows the decrease in Average Cycle Flow Time compared to the FIFO-FIFO model in minutes; $Avg\ Cycle\ Flow\ Time_{FIFO-FIFO} - Avg\ Cycle\ Flow\ Time_{BF-BF}$.

Table 13 Numerical results, comparison of proposed algorithms and FIFO-FIFO model for Cycle Cmax (30, 40, 50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
Type 1 (10, 0.7)	30	1	134.4	134.4	133.2	134.4	134.4	133.2	107.8	107.8	-25.2	-39.8
		2	9.2	27.8	9.4	9.2	27.8	7	20.6	5.2	-28	-35.6
		3	-22.4	-3.2	-18.8	-22.4	-3.2	-19	0.6	-14.8	-29.4	-22
		4	-18.6	2.4	-18.2	-18.6	2.4	-17.8	9.4	-9	-18.2	-8.6
	40	1	203.8	203.8	202.6	203.8	203.8	202.4	201.6	201.6	19.8	30.2
		2	32.4	42.8	32.4	32.4	42.8	32.8	43.8	28.4	-26.4	-25.8
		3	-27.4	-10	-25.6	-27.4	-10	-29	-4.8	-21.4	-25.4	-27.8
		4	-12.8	-3	-12.6	-12.8	-3	-18.4	0.6	-15.4	-21.4	-14.2
	50	1	250.6	250.6	250.2	250.6	250.6	250	234	234	30	1
		2	42.4	61.8	45.8	42.4	61.8	43.2	63.6	41.2	-35.4	-47.4
		3	-16.6	9.2	-7	-16.6	9.2	-16.4	4.2	-16.2	-21.6	-29.2
		4	4.8	7	4.8	4.8	7	5.4	19	9	5.6	9.6
Type 2 (10, 0.8)	30	1	155	155	153.8	155	155	154.2	114.2	114.2	20	27.8
		2	27.4	31.2	21.6	27.4	31.2	23.4	30.2	15.2	-41.2	-42.6
		3	-29	-9.4	-28.2	-29	-9.4	-27.8	-3.8	-14.8	-31.2	-22.6
		4	-3.8	-2	-3.4	-3.8	-2	-4	7.8	-4.8	-13.8	-4.6
	40	1	212.8	212.8	211.6	212.8	212.8	212.4	183.8	183.8	-20	-24.8
		2	23	38.4	24.4	23	38.4	24.2	41.6	32	-45.2	-52.6
		3	-9.4	0.8	-8.8	-9.4	0.8	-6.6	8	-8.2	-8.8	-12.6

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
		4	-20.2	-3.4	-18	-20.2	-3.4	-20	0	-6	-20.6	-6
	50	**1**	309	309	308.6	309	309	307.4	268.8	268.8	62	48
		2	82.4	101	88.4	82.4	101	88.4	82.8	65.6	-60.2	-40.8
		3	3.4	24	3.8	3.4	24	4.2	19.6	7.8	-4.8	-4.2
		4	-17	11	-11.8	-17	11	-16.4	6.6	-10.8	-16.4	-10.8
Type 3 (10,0.9)	**30**	**1**	201.6	201.6	192.4	201.6	201.6	200.4	162.8	162.8	-7.2	-34
		2	17.4	34.2	17.8	17.4	34.2	17.8	32.2	5.4	-37.8	-34.8
		3	-11.8	0.4	-16.6	-11.8	0.4	-11.4	5.8	-7.6	-14.6	-10.2
		4	-11.2	-0.4	-11	-11.2	-0.4	-17.4	5	-5	-12	-17.6
	40	**1**	213.8	213.8	212.8	213.8	213.8	213	175	175	4.8	15
		2	20.2	29.8	21.4	20.2	29.8	20.6	22	4.6	-49.8	-43
		3	-21.4	-3.6	-12	-21.4	-3.6	-20	2.2	-21.6	-32.2	-29.4
		4	3.2	12	3	3.2	12	3	19.4	-3.6	3.2	-0.6
	50	**1**	293.2	293.2	292.6	293.2	293.2	292.4	279.2	279.2	36.6	22.2
		2	19	28.8	18.2	19	28.8	6	25	10.2	-62.2	-60.2
		3	-37.6	-10.4	-36.4	-37.6	-10.4	-36.2	-11	-27.6	-44.8	-37.6
		4	-20.2	-3.2	-28.2	-20.2	-3.2	-28.8	5.2	-19	-28.6	-18.6
Type 4 (20, 0.7)	**30**	**1**	179.8	179.8	179.2	179.8	179.8	178.6	149.6	149.6	-17.6	-34.4
		2	11.2	24.2	12.6	11.2	24.2	12.4	21.6	14.2	-9.2	-21.2
		3	-15.2	-9.4	-10.4	-15.2	-9.4	-10.6	2.2	-6	-24	-18.2
		4	-20.2	6.6	-22.4	-20.2	6.6	-26.2	16.2	-5.2	-3	-4.6
	40	**1**	213.8	213.8	212.2	213.8	213.8	212.8	189.2	189.2	5.6	5
		2	27.8	41	26.2	27.8	41	27	25	10.8	-44.2	-56.2
		3	-0.2	3.8	-7.6	-0.2	3.8	-8.2	3	-9	-17.4	-16.2
		4	-15.2	-11.4	-26.8	-15.2	-11.4	-18.4	4.2	-9.6	-34.8	-8.8
	50	**1**	269.4	269.4	268.4	269.4	269.4	268.4	263.4	263.4	98	92
		2	40.6	56.6	39	40.6	56.6	39.8	66.2	51.8	-37.2	-41.6
		3	-16.8	-1	-16.2	-16.8	-1	-15.2	4.6	-14.6	-37.2	-18.6
		4	-16.8	-5	-11.2	-16.8	-5	-13.8	-1	-7.4	-15.6	-4.4
Type 5 (20, 0.8)	**30**	**1**	180.6	180.6	179.8	180.6	180.6	179.2	172.2	172.2	8.8	0.8
		2	-8.8	7.6	-9.2	-8.8	7.6	-2.2	4	-2.4	-62.4	-51.4
		3	-11	5	-18.6	-11	5	-12.2	13.6	-0.4	-23.2	-17.4
		4	-10.8	0.4	-11	-10.8	0.4	-10.8	10.2	-0.4	-16.8	0
	40	**1**	264	264	263.4	264	264	262.8	250.6	250.6	79.2	67.2
		2	34	48	28.8	34	48	31	49.4	34	-40.4	-53.8

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
		3	-14.2	1.2	-15.8	-14.2	1.2	-15.6	7	-9.8	-12.2	-15.2
		4	-9	-3.2	-10.8	-9	-3.2	-15.4	9.6	-8.8	-17	-11.8
	50	**1**	307.6	307.6	306.8	307.6	307.6	307.2	300.6	300.6	87.4	80.2
		2	55	73	51	55	73	52	76.4	53.2	-13	-31.8
		3	-5.4	6.6	-1	-5.4	6.6	-2	14	0.2	-16.8	-4.6
		4	-19.4	-9.6	-23.8	-19.4	-9.6	-20.6	-1	-16.8	-23.2	-16.8
Type 6 (20, 0.9)	**30**	**1**	104.2	104.2	103	104.2	104.2	103.4	99.8	99.8	-6.6	13.6
		2	14.4	29	12.4	14.4	29	14	25	8.2	-35.8	-21.8
		3	-2	-1.6	-4.6	-2	-1.6	-10.8	-3.8	-4	-2.4	-4.4
		4	-38	-12.6	-37.8	-38	-12.6	-26.6	0.6	-19.6	-39.6	-14.4
	40	**1**	234.6	234.6	234	234.6	234.6	233.2	177.4	177.4	87	55
		2	30	40.2	29.4	30	40.2	25.2	38.2	12	-31.6	-31.4
		3	-16.2	-6.8	-15.8	-16.2	-6.8	-15.4	6.8	-9.8	-24.2	-9
		4	-8.2	3.4	-8	-8.2	3.4	-7.6	13	5.2	-9.4	5.4
	50	**1**	302.4	302.4	301.8	302.4	302.4	302.4	280.2	280.2	80.4	47.8
		2	22.6	38.2	21.8	22.6	38.2	22.6	49.6	34.6	-36	-43.4
		3	-14.4	-7.4	-15.2	-14.4	-7.4	-11	4	2	-18.2	-14.8
		4	-5.6	7.6	-6.2	-5.6	7.6	-5.6	8.2	-10.4	-2.8	-10.2
Type 7 (30, 0.7)	**30**	**1**	160.4	160.4	159.4	160.4	160.4	159.8	139.8	139.8	-37.8	-56
		2	13	38.4	15.8	13	38.4	13	27.6	10	-54.4	-44.8
		3	-24.4	-15.8	-14	-24.4	-15.8	-21.6	-3.6	-29.2	-42.2	-11.8
		4	-23.8	-14	-27.2	-23.8	-14	-27.4	2.2	-28	-26.4	-26.4
	40	**1**	215.6	215.6	215	215.6	215.6	214.2	208	208	5.2	12
		2	36.2	48.4	35.2	36.2	48.4	35.6	49.8	21	-19	-27.8
		3	-20.6	-10.6	-33.8	-20.6	-10.6	-33.8	-1.6	-22.8	-33.6	-11.8
		4	-21.6	-11	-21.8	-21.6	-11	-21.8	7.2	-10.8	-22	-10.2
	50	**1**	345.4	345.4	344.4	345.4	345.4	345	305	305	125.2	109.4
		2	32	47.4	32.4	32	47.4	32.2	42	23.8	-17.4	-41.4
		3	-10	2.8	-0.8	-10	2.8	0.8	15.4	-4.6	-8.6	-10.4
		4	-2.4	0.8	-3.6	-2.4	0.8	2.6	4.2	-7.6	-17.2	-6
Type 8 (30, 0.8)	**30**	**1**	118.8	118.8	117	118.8	118.8	118	117.6	117.6	-53	-53.8
		2	-22	-5.6	-20.6	-22	-5.6	-24.8	-10	-30.6	-46.4	-64
		3	-14.8	-2.2	-7.4	-14.8	-2.2	0.2	1.6	6.6	-21.4	-2
		4	-21.4	-19.8	-22.8	-21.4	-19.8	-18.6	1	-5.2	-21.6	-5.8
	40	**1**	260	260	250	260	260	259	225.8	225.8	-11.2	-18.8

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
		2	11.6	27.6	9.8	11.6	27.6	11.6	36.2	22.2	-49.2	-43.8
		3	-11	3	-1.2	-11	3	-11.4	8.8	8.4	-6.6	-7
		4	-7.4	-10.4	-8.8	-7.4	-10.4	-7	0.6	-9	-22.8	-14
	50	**1**	325	325	324	325	325	325	294.6	294.6	117.8	74
		2	20.2	37	18	20.2	37	17.2	24.2	12.8	-45.6	-39.6
		3	6	9	3.8	6	9	8.2	16	11.8	-21.8	-20.6
		4	13.2	7.6	7.8	13.2	7.6	2.2	17.6	5.6	-6	-1
Type 9 (30, 0.9)	**30**	**1**	188.2	188.2	187	188.2	188.2	186.8	158.4	158.4	88.8	83.8
		2	20.6	25.2	11	20.6	25.2	20	25.6	7.6	-54.4	-53.4
		3	-26.8	-25.4	-24.2	-26.8	-25.4	-27.6	3.8	-12.8	-29.6	-14.6
		4	-35.2	-28.4	-36.6	-35.2	-28.4	-36	0.6	-19.2	-35.8	-18.8
	40	**1**	283.4	283.4	282.6	283.4	283.4	282	258.6	258.6	75	87
		2	35.8	43.6	35.4	35.8	43.6	39	51	40.2	-50.8	-52.6
		3	-24.4	-11.6	-26.6	-24.4	-11.6	-33.6	7.6	-7.6	-25	-17.6
		4	-28.8	-19.2	-29.6	-28.8	-19.2	-29.4	4.2	-12.8	-30.6	-18.8
	50	**1**	252.2	252.2	251.4	252.2	252.2	252.4	244	244	54.6	71.8
		2	47.8	60.4	36.4	47.8	60.4	47.4	47	26.6	-32.2	-20.2
		3	-20.4	-4.2	-18.2	-20.4	-4.2	-17.8	8.4	-16.4	-20.4	-16
		4	-37.2	-22.8	-29.2	-37.2	-22.8	-35	4	-11.2	-38.2	-10.8

Table 14 Numerical results, comparison of proposed algorithms and FIFO-FIFO model for Average Cycle Flow Time (30, 40, 50 jobs)

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
Type 1 (10, 0.7)	**30**	**1**	96.71	93.17	97.85	96.71	93.17	97	79.76	82.19	41.17	37.17
		2	28.58	26.59	27.26	28.58	26.59	27.42	27.53	27.57	8.84	2.89
		3	17.03	13.71	16.49	17.03	13.71	16.81	11.06	15.51	14.33	13.68
		4	14.65	12.55	16.63	14.65	12.55	16.84	15.97	20.53	15.06	19.26
	40	**1**	179.91	177.19	181.24	179.91	177.19	180.59	192.27	194.47	100.58	113.21
		2	31.2	29.09	31.03	31.2	29.09	32	31.45	33.22	7.51	6.39

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
		3	17.01	9.96	15.43	17.01	9.96	16.04	12	20.16	11.17	14.22
		4	15.64	13.01	13.8	15.64	13.01	15.47	16.6	17.1	12.69	16.75
	50	**1**	215.76	213.34	214.2	215.76	213.34	215.64	204.83	207.51	104.45	90.42
		2	36.05	34.38	36.52	36.05	34.38	35.81	36.36	36.37	8.26	-2.37
		3	19.73	16.46	19.42	19.73	16.46	19.24	14.69	18.85	14.12	12.87
		4	15.96	12.68	16.56	15.96	12.68	16.11	18.69	21.11	16.71	20.51
Type 2 (10, 0.8)	**30**	**1**	126.25	121.42	125.4	126.25	121.42	125.27	114.49	120.39	76.27	84.01
		2	25.39	21.74	24.18	25.39	21.74	22.47	20.15	23.61	3.23	6.48
		3	16.19	12.7	14.33	16.19	12.7	14.31	15.41	18.78	13.29	16.27
		4	21.19	14.66	20.67	21.19	14.66	21.36	20.41	24.36	18.29	23.9
	40	**1**	153.08	149.75	153.54	153.08	149.75	153.34	137.08	139.26	58.28	57.34
		2	22.26	21.87	22.85	22.26	21.87	21.38	24.04	24.21	-7.59	-8.35
		3	11.42	11.59	11.77	11.42	11.59	10.49	14.34	16.82	10.18	13.82
		4	18.95	13.71	17.8	18.95	13.71	17.74	16.95	22.43	16.58	21.71
	50	**1**	237.51	235.31	238.76	237.51	235.31	237.39	221.47	223.93	133.66	134.92
		2	64.36	64.77	64.65	64.36	64.77	64.36	57.33	56.51	-7.56	6.75
		3	15.93	10.21	15.9	15.93	10.21	14.41	12.63	16.97	13.05	10.36
		4	16.95	15.73	16.74	16.95	15.73	15.98	20.5	21.72	15.27	21.6
Type 3 (10,0.9)	**30**	**1**	128.81	125.41	128.97	128.81	125.41	129.83	111.31	115.97	51.2	46.18
		2	32.58	32.49	32.59	32.58	32.49	33.41	31.33	31.17	10.17	14.1
		3	11.11	7.71	10.99	11.11	7.71	12.29	9.69	13.02	5.87	9.76
		4	14.96	10.53	15.73	14.96	10.53	15.67	17.07	21.65	14.32	22.51
	40	**1**	172.78	167.65	171.41	172.78	167.65	172.69	160.61	164.78	98.5	102.23
		2	21.3	20.26	20.38	21.3	20.26	20.15	19.06	19.52	2.75	10.14
		3	14.96	11.8	16.01	14.96	11.8	12.25	12.23	13.29	9.26	7.68
		4	18.74	12.94	19.42	18.74	12.94	18.64	20.84	25.54	17.99	25.52
	50	**1**	199.2	196	199.92	199.2	196	199.82	199.72	201.92	113.45	101.37
		2	33.81	33.06	35.02	33.81	33.06	33.32	32.71	34	0.73	2.63
		3	17.96	12.8	17.54	17.96	12.8	17.91	17.28	22.74	15.64	21.1
		4	15.52	12.38	15.19	15.52	12.38	15.84	17.37	21.9	14.12	21.67
Type 4 (20, 0.7)	**30**	**1**	121.96	118.07	121.63	121.96	118.07	121.97	112.89	116.37	45.15	46.59
		2	26.81	23.79	27.98	26.81	23.79	27.33	16.41	19.04	8.57	4.91
		3	15.45	12.68	16.01	15.45	12.68	17.39	13.34	19.01	10.15	18.28
		4	12.97	10.45	12.25	12.97	10.45	12.49	17.39	18.76	9.56	19.16
	40	**1**	156.54	151.48	155.98	156.54	151.48	156.87	148.84	152.21	75.09	81.34

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
		2	25.66	24.14	26.58	25.66	24.14	25.71	21.37	22.19	-3.55	-5.59
		3	11.04	8.6	11.29	11.04	8.6	10.34	12.42	18.06	3.84	16.29
		4	13.89	7.08	12.98	13.89	7.08	14.05	15.44	20.74	8.38	18.61
	50	**1**	224.3	221.95	222.6	224.3	221.95	223.13	224.05	226.04	166.65	158.96
		2	41.8	41.77	41.86	41.8	41.77	42.21	49.12	49.01	-7.82	1.78
		3	15.22	8.09	14.19	15.22	8.09	14.81	13.18	19.44	10	15.71
		4	14.05	12.37	14.72	14.05	12.37	14.36	16.73	19.14	10.4	19.1
Type 5 (20, 0.8)	**30**	**1**	146.51	141.99	147.11	146.51	141.99	146.95	150.75	155.96	76.21	78.68
		2	14.49	14.38	15.89	14.49	14.38	17.12	15.76	19.06	-7.97	-1.93
		3	15.49	10.73	14.91	15.49	10.73	16.52	16.21	24.6	10.39	21.65
		4	13.33	11.69	13.97	13.33	11.69	15.23	18.88	23.33	9.48	24.21
	40	**1**	193.82	190.63	193.7	193.82	190.63	192.34	189.37	190.99	116.69	121.27
		2	29.96	29.15	28.75	29.96	29.15	30.48	33.98	33.79	-4.41	-1.24
		3	16.93	10.74	18.16	16.93	10.74	17.99	15.86	21.88	11.37	16.52
		4	10.85	11.06	11.75	10.85	11.06	10.22	17.45	20.36	8.36	17.38
	50	**1**	213.3	210.76	211.47	213.3	210.76	211.87	212.32	214.98	133.4	127.32
		2	35.98	34.64	36.36	35.98	34.64	36.34	35.38	38.41	-0.09	0.89
		3	15.87	11.33	15.63	15.87	11.33	15.48	18.62	20.75	8.8	17.46
		4	10.33	7.01	11.05	10.33	7.01	10.96	16.69	19.85	7.38	20.77
Type 6 (20, 0.9)	**30**	**1**	119.64	113.7	120.63	119.64	113.7	121.45	117.71	124.1	74.64	93
		2	20.21	16.97	22.47	20.21	16.97	22.09	20.53	21.39	0.79	5.65
		3	8.43	7.73	7.34	8.43	7.73	9.79	13.73	17.1	6.89	16.27
		4	7.29	4.41	7	7.29	4.41	8.75	15.09	18.68	4.05	16.75
	40	**1**	179.78	177.24	179.01	179.78	177.24	179.79	166.15	169.48	128.05	115.67
		2	26.9	24.1	27.5	26.9	24.1	26.18	29.5	30.85	3.11	6.81
		3	15.11	9.39	14.96	15.11	9.39	14.36	15.7	21.84	8.9	17.89
		4	11.09	7.28	9.87	11.09	7.28	11.37	16.39	21.45	8.15	20.32
	50	**1**	217.35	215.1	217.18	217.35	215.1	216.93	208.91	211.4	126.5	108.19
		2	24.25	22.4	23.9	24.25	22.4	24.13	25.27	26.71	-2.69	-0.93
		3	11.57	10.89	12.86	11.57	10.89	11.58	18.44	21.34	5.72	15.78
		4	8.55	7.59	8.85	8.55	7.59	8.78	16.49	19.62	4.43	19.63
Type 7 (30, 0.7)	**30**	**1**	137.25	133.32	138.18	137.25	133.32	138.11	127.16	131.38	40.34	28.27
		2	19.15	18.99	19.24	19.15	18.99	20.69	22.21	21.99	-8.29	-6.07
		3	16.41	12.71	16.3	16.41	12.71	14.87	17.29	21.75	6.61	20.67
		4	12.98	7.31	12.59	12.98	7.31	14.14	17.91	24.01	11.63	24.29

Input Type (Wait Time Threshold, Capacity Threshold)	Number of Jobs	Number of Machines	BF - BF	BF - FIFO	BF - NoWait	FF - FF	FF - FIFO	FF - NoWait	NoWait - FIFO	NoWait - NoWait	RH - NoWait	RHNoWait - NoWait
	40	1	170.76	167.19	172.94	170.76	167.19	172.63	184.36	187.25	89.66	99.96
		2	32.29	29.21	32.54	32.29	29.21	33.39	36.26	37.6	8.21	9.81
		3	12.05	10.07	12.14	12.05	10.07	12.65	12.32	17.13	5.03	14.62
		4	11.95	10.15	11.38	11.95	10.15	11.08	18.04	21.09	3.71	20.24
	50	1	255.64	252.98	254.47	255.64	252.98	254.84	241.85	244.15	160.6	155.16
		2	33.88	32.27	33.62	33.88	32.27	34.07	36.12	37.57	4.25	9.93
		3	13.36	12.07	13.06	13.36	12.07	14.38	13.62	18.14	6.75	13.99
		4	11.52	8.66	10.73	11.52	8.66	13.14	14.39	18.64	7.25	18.18
Type 8 (30, 0.8)	30	1	89.1	86.03	90.03	89.1	86.03	90.66	83.04	86.11	30.53	16.24
		2	16	13.93	16.57	16	13.93	16.83	14.39	15.24	-22.27	-4.28
		3	13.69	10.11	16.45	13.69	10.11	14.05	19.41	21.58	4.02	16.75
		4	8.05	4.81	11.48	8.05	4.81	8.44	15.49	20.28	1.74	18.1
	40	1	189.69	187.3	189.27	189.69	187.3	189.49	178.44	181.31	78.65	78.58
		2	19.35	18.95	20.42	19.35	18.95	21	26.25	25.79	-1.96	1.73
		3	13.67	9.66	15.86	13.67	9.66	14.83	15.59	23.45	11.37	21.67
		4	6.7	4.32	9.07	6.7	4.32	8.92	18.89	20.75	4.08	20.17
	50	1	242.08	239.86	242.98	242.08	239.86	242.95	237.96	240.78	164.98	151.68
		2	21.13	17.29	18.94	21.13	17.29	19.13	11.79	14.06	-1.34	-6.14
		3	9.23	8.56	7.6	9.23	8.56	6.46	14.2	15.6	-6.11	7.45
		4	10.6	9.17	10.31	10.6	9.17	9.27	17.7	19.18	5.5	19.76
Type 9 (30, 0.9)	30	1	136.73	131.79	137.31	136.73	131.79	137	117.99	123.36	94.32	88.89
		2	17.44	15.89	18.54	17.44	15.89	17.47	23.35	25.81	-15.27	-0.66
		3	8.21	1.2	5.85	8.21	1.2	7	13.25	20.31	-0.42	20.04
		4	7.7	1.27	8.73	7.7	1.27	8.6	16.24	24.41	6.39	24.08
	40	1	197.93	195.21	197.69	197.93	195.21	196.52	199.26	201.83	120.33	140.19
		2	31.43	31.74	31.93	31.43	31.74	32.55	35.65	37.22	-9.28	2.44
		3	7.04	4.01	6.92	7.04	4.01	8.17	12.33	18.06	-2.62	10.48
		4	4.28	0.81	5.46	4.28	0.81	4.17	16.4	20	1.35	19.43
	50	1	204.4	201.9	206.27	204.4	201.9	206	207.79	210.8	113.27	125.94
		2	33.22	32.11	32.74	33.22	32.11	32.94	33.76	35.29	-3.91	7.64
		3	7.8	6.44	8.64	7.8	6.44	6.45	13.52	16.26	-0.65	16.72
		4	6.08	4.28	7.3	6.08	4.28	5.49	17.31	20.81	1.5	20.75