POLITECNICO DI MILANO

Department of Management, Economics and Industrial Engineering
Master's Degree in Management Engineering

Routing Flexibility and Workers Reallocation in a shop floor with parallel machines and Workload Control

Supervisor:
Prof. Alberto Portioli Staudacher

Tutor:
Eng. Federica Costa

Author:
Giuseppe Schilleci - Matricola 894066
Davide Manzionna – Matricola 894494

Academic year 2018/2019

Abstract (English)

As a response to high-variety low-volume manufacturing environment of Make-to-Order firms, production managers may adopt a workload control system. The latter is a control method that manages the workload and creates a smooth production flow by filtering jobs released to the shop floor. Although workload control systems have been demonstrated by academics to consistently improve shop floor performances, it has been poorly applied by practitioners probably due to the oversimplified models studied by scholars. One of these key simplifications is the lack of consideration of parallel machines in a station of a job shop, that instead are frequently found in real shop configurations. The thesis aims to cover such gap by studying a shop floor with a station with parallel machines under a workload control system. More importantly, the paper analyzes two capacity adjusting methods applied in parallel machines that have not been tackled in the workload control literature, that are: the routing flexibility and the reallocation of workers only within the station with parallel machines.

The thesis first independently studies routing flexibility and workers reallocation on a flow shop (with parallel machines in a station) where it is implemented a workload control and then compares the two methods under different system parameters.

Both routing flexibility and workers reallocation have been found to lead to consistent improvements of the main system performances (gross throughput time and shop floor time) and surprisingly these benefits are independent by the unbalance level of the system.

In what concerns the specific parameters of the routing flexibility model, it has been discovered that such algorithm brought already most of the benefits (77 percent) with a low level of interchangeability (20 percent). While in the workers reallocation method, it is impacted more by the transfer time and level of workers' efficiency than the permanence time and the decentralized "when" rule outperformed the centralized one.

Finally, the routing flexibility and workers reallocation have been compared in both the balanced and unbalanced case. The two models led to similar results in the balanced scenario, whilst the routing flexibility slightly outperformed the workers reallocation in the unbalanced scenario.

Abstract (Italian)

Per affrontare un ambiente manufatturiero caratterizzato da alta varietà e bassi volumi, come è quello generalmente incontrato da aziende Make-to-Order, i responsabili di produzione possono adottare sistemi di controllo del carico di lavoro. Questo metodo di controllo filtra gli ordini che vengono rilasciati in produzione, perseguendo l'obiettivo di creare un flusso di produzione regolare. Nonostante diversi studi accademici abbiano dimostrato che può portare consistenti benefici nelle performance produttive, le applicazioni nel campo pratico sono state limitate. Questo probabilmente è dovuto al fatto che i modelli analizzati negli studi di simulazione vengono costruiti con assunzioni troppo distaccate da reali processi produttivi. Una di queste è che non si considerano generalmente macchine produttive operanti in parallelo, cosa che tuttavia avviene di frequente in applicazioni reali. La tesi si pone l'obiettivo di colmare questo divario studiando un layout con macchine in parallelo in cui viene applicato un sistema di controllo del carico di lavoro. In particolare, lo studio analizza due metodi di bilanciamento della capacità produttiva, che in letteratura non sono stati trattati in contesto di macchine in parallelo. Questi metodi sono la flessibilità del routing e la riallocazione dei lavoratori, con quest'ultima effettuata soltanto tra le stazioni con macchine in parallelo. La tesi studia indipendentemente l'applicazione dei due metodi in un flow shop (con macchine in parallelo in una stazione) nel quale è implementato un sistema di controllo dei carichi di lavoro, e infine paragona i due metodi secondo diversi parametri di sistema.

È emerso che sia la flessibilità del routing che la riallocazione portano a benefici consistenti riguardo le principali performance del sistema (tempo di produzione lordo e tempo di attraversamento) e che, sorprendentemente, tali benefici sono ottenuti indipendente dal livello di sbilanciamento del sistema. Invece per quanto riguarda gli specifici parametri, la flessibilità del routing ha dimostrato che porta la maggior parte dei benefici al sistema (77 per cento) con un basso livello di intercambiabilità degli ordini (20 per cento). Mentre è stato statisticamente provato che nella riallocazione dei lavoratori il tempo di trasferimento e livello di efficienza degli operatori hanno più impatto rispetto al tempo di permanenza; e la che regola di riallocazione di tipo decentralizzata porta risultati migliori rispetto a quella centralizzata. Infine, è stato fatto un confronto fra la flessibilità del routing e la riallocazione dei lavoratori sia un caso di sistema bilanciato che in uno sbilanciato. I due metodi portato al sistema quasi lo stesso livello di benefici nel suo caso bilanciato, mentre la flessibilità del routing ha dato risultati leggermente migliori nel caso di sistema sbilanciato.

Executive Summary

With the recent shift towards mass customization, several manufacturing companies have adapted to a make-to-order (MTO) production system. In this context, the manufacturing environments are characterized by a demand that generally has a high variety and low volume. Furthermore, customers have increased their requests regarding quality, delivery time, flexibility and variety of the offer. As competition has increased under many aspects, MTO firms had the necessity to find a way to adapt their production system to the changes of the market. The concept of workload control was born to give an answer to this.

The workload control is a method used by companies to regulate the production process, in order to create a smooth and balanced production flow. The goal of the workload control mechanism is to avoid shop congestion and to prevent the creation of bottlenecks. At the same time it aims to obtain high machine utilizations and to improve the delivery time of the system. In a workload control system the customers' orders, after being accepted, are first held in an initial buffer, called pre-shop pool (PSP).

In a Workload Control concept, customer orders' are considered as jobs and they are released from the PSP according to an algorithm, that regulates the workload to be sent into the system. This first leverage is called input control. Many different types of input control algorithms are implemented in manufacturing companies, and their main objective is to limit the total amount of workload in the system to avoid shop congestion and then to obtain a smooth and balanced production flow.

- Workload Limiting algorithm is one of the most used input control algorithms in literature. In this case, an upper bound is set, which is called workload norm, and jobs are released into the system until that predetermined bound is reached. This algorithm will be used to apply input control in the model developed in this thesis.

The other important leverage that the workload control can use regards capacity adjustments. This is called output control, and it refers mainly to dynamically adapting workers or machines to the current necessities of the system. In particular, the thesis will be focused on the study of two output control mechanisms: the workers reallocation and the routing flexibility.

- Workers reallocation is one method of capacity control widely studied in literature. It basically consists in transferring the workers among the different machines, according to an algorithm with the aim of solving the bottlenecks between the stations. Whenever the queue in front of one station gets too long, the system relocates one or more workers to that station in order to rebalance the workload. The relocated workers will spend a certain

amount of time in that station, helping other workers in performing their activities, thus reducing the amount of work accumulated in that station.

- Routing flexibility, instead, refers to the possibility of the system to change the routing of some jobs between similar machines. As it does the other model, also in this case the goal is to prevent the creation of bottlenecks. Jobs that are interchangeable may be moved to less saturated stations to balance the production flow. This is done according to an algorithm, which evaluates the current workload of each station and decides which job should be relocated.

The goal of this thesis is to study the application of these two output control models, in a flow-shop configuration containing parallel machines. Indeed in literature, studies seldom consider the existence of parallel machines when modelling job or flow shops. The simplification "one station = one machine" is often done, in order to focus the attention on the other characteristics of the shop. However, as claimed by Miragliotta and Perona (2010), some of the assumptions generally taken by researchers often lead to provide guidelines and conclusions that are far from the application to a real context. Another simplification that is common in literature is considering machines that are all equal, in order not to create intrinsic imbalances in the system.

Since the aim of the thesis is also to build a model as much close to reality as possible, it has been decided to consider a configuration which entails parallel machines, and to test the two different models (workers reallocation and routing flexibility) also in scenarios with machines having an intrinsic imbalance caused by their different speed. This last fact, to the best of the authors knowledge, has not been implemented in in workload control studies in literature so far.

The fact of having machines with different speed is common in real contexts. In particular, when machines are working in parallel, it is likely that they may have not been bought at the same time, or that they are of different models. This leads to the frequent situation in which MTO companies have parallel machines in line that are working at a different speed or with different batches.

For these reasons, it has been considered in the thesis the model of a flow shop, composed by a total of five stations, of which four contain a single machine and one contains two machines working in parallel. Only one study has been found in literature to have made a comparison between the two models of workers reallocation and routing flexibility. Moreover, no author has yet considered these two models in a parallel stations' configuration, containing machines that are not all equal.

In the thesis, indeed, the two output control models will also be tested in an unbalanced scenario, in which the two machines running in parallel have an imbalance of 10 percent regarding their speed. The goal is to assess the performance of the system and the ability of the two models in solving also the intrinsic imbalances that have been generated.

The first model studied has been the routing flexibility. The main important parameter for this case is the level of interchangeability. This is the percentage that indicates how many jobs that are flowing into the system could potentially be worked by both the parallel machines. This happens for example when the two parallel machines can work products of different measures. The interchangeable products will be the ones containing the measures that are overlapping as they are workable by both the two machines. The routing flexibility algorithm studied in the thesis changes the routing of a job from the most saturated to the least saturated parallel machines, whenever the imbalance of their queues has reached a certain threshold.

Results in this model show that, the routing flexibility model can significantly improve the time performances of the system. Moreover, it has been found that with a low level of interchangeability (20 percent) the model is already capable to obtain most of the benefits (77 percent) that would be achievable with a complete level of interchangeability (100 percent). This means that any further increase in the level interchangeability would only bring to limited marginal advantages.

Then, by testing the model in the unbalanced scenario, it has been demonstrated how the routing flexibility can successfully solve the intrinsic imbalances. Moreover, the higher is the imbalance of the system, the more efficient the routing flexibility algorithm has proven to be. As a matter of fact, in the unbalanced scenario it has been able to obtain around the 80 percent of benefits already with an interchangeability of 10 percent.

The second model to be independently studied was the workers reallocation. In this case, the reallocation of workers has been constrained only between the two parallel machines, also to ease the comparison with the routing flexibility. Results suggest that the transfer time (which is the time needed for a worker to move from his machine to the one he is reallocated to) and the efficiency (capability to effectively help the other worker, measured as the percentage of useful time obtained when relocated) are more impacting on the result than the permanence time (which is the minimum time that the worker needs to spend in a station when relocated).

Regarding the when rule, the decentralized and centralize rule have been tested. The first one allows the relocation of a worker only whenever the machine to which he is assigned is idle, so when there are no jobs in its queue; while the second rule releases this constraint, so that a worker can be

reallocated anytime. Results in this case show how the decentralized outperforms the centralized rule, and a curve displaying the tradeoff between number of relocations and effect on the gross throughput time has been built. The curve illustrates that, after a certain threshold that is reached with the decentralized rule, increasing the number of relocations by using the centralized rule only brings to worsen the results in terms of gross throughput time.

Finally, the model has been tested against the intrinsic imbalances of the unbalanced scenario. Similarly to the routing flexibility case, also the workers reallocation model has shown to be efficient in solving the new bottlenecks, being able to achieve significant reductions of gross throughput time and shop floor throughput time.

In the last part of the thesis, the two models have been compared, both in the balanced (all machine equal) and in the unbalanced scenario (parallel machines with an imbalance in speed of 10 percent). Comparing the two methods, it has been shown how both are able to improve the time performance of the system, obtaining similar results. In the unbalanced case, however, the routing flexibility model slightly outperformed the other model. Anyway, being the difference very small, it could be due to the choice of the levels of the parameters done for the simulation.

The important fact found in this case, however, is that when applied to the unbalanced case both the models have been successful in solving the bottlenecks. Surprisingly, they have achieved this result without increasing the total number of relocations done. This fact demonstrated that the two models are robust against intrinsic imbalances, and that they can successfully solve them obtaining results that are similar to the balanced case.

List of Contents

List of Tables

List of Figures

List of Acronyms

ANOVA	-	Analysis Of Variance
CNC	-	Computer Numerical Control
FCFS	-	First Come First Served
FMC	-	Flexible Manufacturing Workcells
FMS	-	Flexible Manufacturing System
ERP	-	Enterprise Resource Planning
MES	-	Manufacturing Execution System
MSPE	-	Mean Squared Pure Error
MTO	-	Make To Order
MTS	-	Make To Stock
ORR	-	Order Review and Release
PPC	-	Production Planning and Control
PSP	-	Pre Shop Pool
PT	-	Processing Time
RF	-	Routing Flexibility
WIP	-	Work In Progress
WL	-	Workload Limiting
WLC	-	Workload Control
WR	-	Workers Reallocation

PART 1: Literature Review

1. Introduction

One of the most recent trends impacting manufacturing companies is product customization. Nowadays consumers ask for more and more personalized products that are much more complex to manage for a producer than the mass products characterizing the previous decades. As a result of the shift from mass production to customization, many manufacturing firms have been required to change their production system from Make-to-Stock (MTS) to Make-to-Order (MTO).

MTS is a manufacturing system according to which products are produced based on forecast of the future demand. Finished products are stocked and send to customers when the order is received. This solution reduces the lead time to customer and suits for stable demands (easily predictable and not variable) and for high volume production with low variety, where there is a repetitive and standardized production. All these requirements of a MTS system are not in line with the product customization trend. In fact, the latter entails a high variety and low volume production, where demand is extremely variable. Rather, the management of customized products better fit with a MTO system.

As opposed to MTS, in a MTO system production starts only when an order has been received. As production is "pulled" by on time customer demand, it is possible to deal with the specific customized products because the system is much more reactive and flexible. In order to assure a proper flexibility level in the shop floor, the production layout most adopted by MTO companies is a job-shop.

Despite the adoption of a reactive production through MTO system and job-shop layout, MTO firms still face difficulties in addressing the increasing customized products requested by customer. This is due to the consequent high level of demand variability that causes a continuous and unpredictable movement of workload among stations. Thus, it is complex to evaluate the actual production capacity and to understand where the bottlenecks are. This prevents MTO companies from properly estimating production lead time, forecasting delivery date and being punctual on delivery. This a crucial issue for MTO firms because those metrics are extremely important for customers and hence are order winning performances.

Production managers tend to react to this problem by releasing the all the workload to the shop floor. However, this dramatically increases the WIP, which in turn worsens lead time and leads to delays in deliveries (Portioli & Tantardini, 2012). To address the high variability and unpredictability level present in MTO companies, the latter may adopt Production Planning and Control (PPC) techniques. One of the most discussed PPC tool is the Workload Control system.

A Workload Control system has shown to be effective in the high-variety low-volume manufacturing environment of MTO firms. It indeed controls the workload and creates a smooth production flow by only releasing jobs to the shop floor that do not generate shop congestion or excessive queues. More specifically, in a workload control system jobs are stored in a Pre Shop Pool (PSP) before being released to the shop floor. In the PSP there is an Order review and Release (ORR) algorithm that is responsible of the sequencing decision and release decision. The sequencing decision concerns the sequence for reviewing orders, while the releasing decision determines whether to release jobs.

The workload control has demonstrated to consistently improve shop floor performances, such as production lead time and WIP decrease and better due date compliance. Despite these benefits, such system has been poorly applied by practitioners. Bertolini, Romagnoli and Zamorri (2015), Yuan (2017), Thürer, et al. (2012) and Miragliotta and Perona (2000) argue that one of the main reasons of the poor applications of practitioners are the many simplifications made in the simulated models by academics that are far from reality and hence not replicable in real production systems.

These simplifications also regard the shop configurations, such as the general assumption made by most academics of "one station = one machine" (see for example Land, Stevenson, Thürer, & Gaalman, 2015; Thürer, Stevenson, Land, & Fredendall, 2018; Park & Bobrowski, 1989). Such simplification does not reflect real job shop configurations, as each station is often made up by similar or equal machines (Henrich, Land, & Gaalman 2006; Stevenson, 2006).

The lack of consideration of more machines per station prevented academics to study one of the key feature of a job shop station: the presence of parallel machines in a station. The thesis claims the importance of considering parallel machines when studying a workload control system, because – as affirmed by Miragliotta and Perona (2000) and Bokhorst & Gaalman (2009) – not taking them into account is one of the reasons of the poor practical implementation of workload systems. Moreover, it is interesting studying the impact of workload control on parallel machines, since the latter face unbalances issues and thus are likely to be the bottleneck that could constrain order release from the pre-shop pool. (Bokhorst & Gaalman, 2009; Sirikrai & Yenradee, 2006). Given the

frequent workload imbalance among parallel, when studying the latter it is needed to study its two main capacity balancing method: routing flexibility and workers reallocation on parallel,

Routing flexibility is an advantage typical of job shops and consists on the flexibility of jobs in changing their initial routing to a new routing leading to better shop floor performances (i.e. where the job may flow faster). The change of a routing of jobs is typically applied in real job shop among parallel machines, because they are often interchangeable. This means that a job can be worked indifferently by all machines that are in parallel in a station.

Even though Henrich, Land and Gaalman (2007), Henrich, Land and Gaalman (2006), Fernandes and Carmo-Silva (2011) and Zhao, Gao, Chen and Xu (2015) have shown the consistent benefits that routing flexibility can bring to a workload control system, this solution has been poorly discussed in the workload control literature. The thesis aims to cover such gap by performing a thorough analysis on routing flexibility in parallel machines of a shop floor under a workload control system.

Workers reallocation is an output control method of a workload control system that consists on moving workers from oversaturated to undersatured machines. Although the reallocation of workers in a shop floor under the implementation of workload control has proven to drive to consistent benefits in performances, it seems that the assumptions made by most researchers are not applicable in real industrial contexts (Yue, Slomp, Molleman & Van Der Zee, 2008).

Fruggiero, Fera, Iannone and Lambiase (2015) claim that most of the research on reallocation of workers made too simplistic assumption such as the equality of efficiency of workers in the transferred machines and the lack of consideration of transfer time. These two assumptions have driven researchers to consider that workers could be reallocated to all machines in a shop floor (Yue, Slomp, Molleman, & Van Der Zee, 2008). However, if it is properly considered the decrement in efficiency in external machines and mostly the impact of transfer time, reallocating workers in all machines may be far from real scenario. Apart from this fact, several studies showed that the majority of improvements were reached with a limited level of workers flexibility, i.e. letting transfer only few operators on limited machines (Felan & Fry, 2001; Park, 1991; Malhotra, Fry, Kher, & Donohue, 1993; Fry et al., 1995; Campbell, 1999).

Due to such results and the thesis broad objective to test a simulated model as much as close to reality, the paper studies workers reallocation only within parallel machines. In fact, in such a way, the reallocation is limited, and this is in line with the just mentioned studies. Moreover, it seems more realistic to reallocate workers only among parallel machines, because:

- the decrease in efficiency is less stringent, as parallel machines are similar or even equal.
- there is less impact of transfer time, since parallel machines are in the same station and then close to each other.
- it requires less expensive cross-training, which is employed only for workers operating in parallel machines.
- the reallocation is necessary due to the workload imbalance affecting parallel machines.

Finally, given routing flexibility and workers reallocation have the same objective of readjusting the capacity of parallel machines, the thesis compares them by varying their common system parameters. This would allow to assess which of the two outperforms the other and mostly under which conditions and levels of system parameters.

1.1 Objective of the thesis

The objective of the thesis is to deeply analyze and study a workload control system under a shop configuration as much as close to real industrial contexts. To this aim, it has been simulated a flow shop with 5 stations, one of which with two machines working in parallel. More importantly, it has been studied two pragmatic output control methods for the station with parallel machines that have been poorly discussed in the workload control literature. These methods are the routing flexibility and workers reallocation within parallel machines.

The paper first independently analyzes the routing flexibility and workers reallocation model with their respective parameters, then makes a comparison between the two methods by varying their common system parameters.

It is here presented the specific three research questions of the thesis that will be deeply discussed later:

> 1) What is the contribution of the routing flexibility to performances and how these are affected by the level of interchangeability? What is the minimum level of interchangeability that leads to most of the result?
>
> 2) What is the contribution of workers reallocation to performances and how these are affected by the efficiency of the workers, the permanence time and the transfer time

between machines? Which when rule (decentralized, centralized) leads to the most benefits in a parallel flow-shop configuration?

3) What is the respective contribution of the two methods (routing flexibility and workers reallocation) to performances, to the variation of system parameters such as stations' imbalance?

1.2 Research methodology

The thesis uses the simulation approach to address its research questions. It has been used simulation through the "SimPy" module of the programming language "Python". In order to test the robustness and sensibility of each model studied under different factors or parameters it has been performed both a "one at-a-time" and ANOVA (Fractional Factorial) experiment, which has been analysed with the use of Minitab. Moreover, parameters of the simulated model are mainly taken from literature. It has been tested a wide range of values of each parameter in order to have an as much as wide validation of the results.

The model with routing flexibility and the one with workers reallocation will be first independently studied by changing their respective model parameters. Since these two models have their own model parameters, when compared, it has been changed only their common system parameters. In this way, it can be tested the robustness of the system and properly understand which model performs better under which conditions.

1.3 Thesis outline

The remaining part of the thesis is organized as following:

- Chapter from 2 to 6 report the literature review, discussing respectively of the workload control concept (2), the order review and release (3), the different shop configurations (4), the routing flexibility (5) and the workers reallocation (6)
- Chapter 7 reports the discussion of literature
- Chapter from 8 to 10 introduce the methodology followed in the thesis, in particular the description of the simulated model (8), the model proposed of routing flexibility (9), the model proposed of workers reallocation (10)
- Chapter 11 explain in detail the parameters used in the simulation
- Chapter 12 reports the design of experiment conducted
- Chapter 13 shows the discussion of results
- Chapter 14 contains the conclusions of the thesis

2. Workload control

Among the several production planning and control concepts developed and analyzed by academics, the workload control is one of the most significant. Workload control is a control technique which decouples job arrivals and planning phase from the production (Melnyk & Ragatz, 1989). It aims at guaranteeing a smooth flow in a shop floor by leveling demand and production over time (Melnyk & Ragatz, 1989). This method is particularly used in make-to-order companies (MTO) where it is not possible to synchronize the flows and imbalances between production and demand are frequent (Stevenson et al., 2005; Thürer et al., 2012; Thürer et al., 2013). MTO companies operate in a low-volume and high-variety environment. They suffer from highly variability in production lead time, which is caused by imbalances in stations loads with orders spending long time in queues (Zäpfel & Missbauer, 1993).

Workload control algorithms prevent job congestion by releasing the orders to the shop floor at the proper time, trying to obtain a balanced system with limited and smoothly proceeding queues. To this aim, the algorithm does not immediately release production orders to the shop floor, but first collects them in a back-log file called pre-shop pool (PSP). It is here decided the sequence and timing of the orders that will be released. In such a way, demand is decoupled from production and it is then reduced its uncertainty which negatively affects MTO companies. As Figure 1 shows, the pre-shop pool (PSP) splits the moments of order entry and order release before the shop floor.

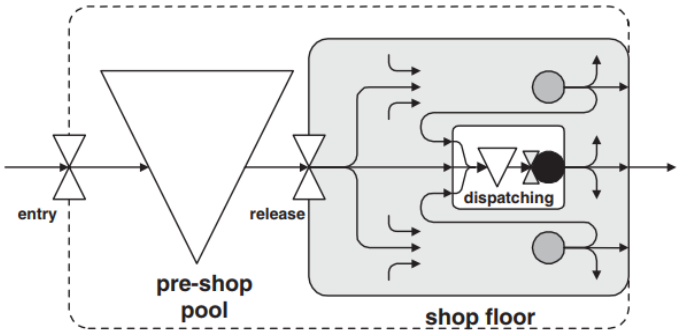


Figure 1 - Decision moments in the flow of a job (adapted from Land 2006)

The workload control system can lead to significant benefits. In particular, when the stations in the shop floor are under-loaded, it can increase the amount of orders released in order to avoid low saturation of both machines and workers. While, if stations are overloaded, it can reduce the number of released jobs in order not to create shop congestion and consequent bottlenecks.

This has an important effect in stabilizing the throughput times and improving the overall time performance of the system (Land, Stevenson, & Thürer, 2013). As a matter of fact, the lack of a workload control algorithm would result in orders being immediately released, that would transfer directly the variability of the demand into the system. This would lead to consequent variable and unpredictable throughput times, which in turn would prevent MTO companies from estimating an accurate delivery time.

The application of the workload control system thus leads to two main advantages: a *reduction of the WIP level* in the shop floor and a *better balanced workload* among different stations. These two benefits entail many consequent positive outcomes which are listed below: protection from demand variability (Breithaupt, Land, & Nyhuis, 2002; Land & Gaalman, 1996; Bertrand & Van Ooijen, 2002); improvement of shop utilization and decrease of shop congestion. (Bechte, 1988; Bergamaschi, et al., 1997; Breithaupt, Land, & Nyhuis, 2002; Stevenson, Hendry, & Kingsman, 2005; Baykasoğlu & Gçken, 2010); reduction and stabilization of shop floor throughput times (Hendry & Wong, 1994; Sabuncuoglu & Karapinar, 1999).

Moreover, other benefits arise both regarding cost and performance:

- *Lower inventory holding cost*: through the reduction of the WIP level, less resources are immobilized within the production system (productive resources and raw materials). This happens because orders spend more time within the pooling system and when released proceed smoothly across production. (Oosterman, Land, & Gaalman, 2000).

- *Increase flexibility*: retaining orders in the pre-shop pool and delaying their release in the shop floor reduces the probability to incur in changes of the order when the latter has been already released. This reduction of the impact of changes is a significant benefit for MTO companies, since producing highly customized products involves frequent order changes with short notice (Land & Gaalman, 1996; Stevenson & Hendry, 2006). The flexibility in facing change of order also leads to a lower cost of order cancellation, as it is postponed the moment an order will be processed (Breithaupt, Land, & Nyhuis, 2002).

- Less variability in production lead times leads also to a *better due date compliance*, with more reliable estimation of due dates. This has a positive impact on the trust that customers have on the company. If less uncertainty is given to customers, they can reduce their inventories, improving their relationship with the company (Kingsman & Hendry, 2002).

Although workload control brings all these benefits and fits in the general manufacturing environment of MTO companies, its implementation in real contexts is still quite limited. As a result, there are only few academic articles that address the application of workload control in a real company context. In this regard, Bertolini, Romagnoli and Zammori (2015), Yuan (2017) and Thürer, et al. (2012) claim that such lack of practical application is due to the over-simplified workload systems studied in literature, that are poorly applicable in real companies. For instance, the majority of researchers assume in their models stations and buffers with unlimited capacity. However, such conditions do not match real contexts, and their lack of consideration strongly biases the results (Sabuncuoglu & Karapinar, 1999).

Furthermore, a constraint to the implementation of workload control algorithms is given by the consistent information and data constantly needed from the shop floor. To decide the timing and sequencing of orders in the PSP to release, the system requires to know exactly the current load of each station and the relative positions of already released orders. Nowadays, not many MTO firms are able to track with accuracy this amount of updated and live information. Nevertheless, the fast-paced growing Industry 4.0 and Internet Of Things are likely to reshape current MTO environments, allowing to track all data and thus facilitating the application of workload control algorithms.

2.1 Input control

Workload control algorithms can control and improve the throughput time of jobs in the shop floor through two main leverages: the input and the output control. The input control manages the release of new orders in the shop floor. To filter the proper amount of orders to be released it needs to retrieve data about the jobs (Kingsman & Hendry, 2002).

More specifically, it uses two types of information:

- *Workload information*: this information includes both the workload of the orders in the PSP that are scheduled to be released, and the orders that are already present in the shop floor. The workload itself is the sum of the processing times of the orders to be and already released. It should also take into account setup times, which however, as claimed by Thürer, et al. (2012), is generally not taken into account by all researchers.

- *Progress control information*: it gives information about the positions of jobs in the shop floor. Knowing where the jobs are currently being processed is crucial for computing the workload, because it allows to calculate updated and more accurate values.

Figure 2 may help to understand when orders start impacting the workload, by identifying all stages the orders have to get through in the workload control algorithm.

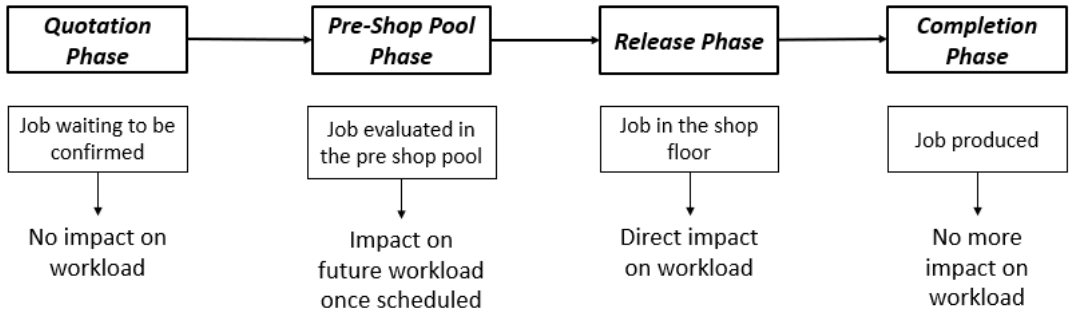


Figure 2 -The impact on workload for the different jobs' phases

The first stage is the Quotation Phase, where an order is sent by a customer who still has not agreed both price and due dates with the manufacturing company. Thus, the order is waiting for confirmation and does not count into the system workload. When the order has been confirmed by the two firms, it is moved to the *Pre Shop Pool Phase* (i.e. the PSP). Here, the workload control algorithm evaluates its release to the shop floor. After being scheduled, the order starts accounting to the system workload and to all the future station that it has to visit in the shop floor. Once definitely released (*Release Phase*), the order has a direct impact on each visited station, which accounts on the Shop Throughput Time. In the last *Completion Phase*, the order exits from the system and stops its workload contribution.

The input control can also manage further information such as the job priority (Kingsman & Hendry, 2002), job size (Kingsman, Tatsiopoulos, & Hendry, 1989), job complexity, product families (Bertolini, Romagnoli, & Zamorri, 2015), set-up times (Thürer, et al., 2012) and production losses (Kingsman & Hendry, 2002; Yuan, 2017). The decision about whether to include those other data strongly depends on the type of input control implemented.

2.2 Output control

While the input control decides which orders are to be released in order to create a smoother flow in production, the output control tries to improve the balance of the flow, by controlling and allocating production capacity according to the jobs that have already been released. This can be useful because the application of the only input control may still lead to unsaturation or imbalances among machines in the shop floor. To carry out production capacity adjustments, three possible levers are generally used: machines, manpower and subcontracting (Kingsman, Tatsiopoulos, & Hendry, 1989; Yuan, 2017).

Machines: The output control can enhance the production capacity of machines by acting on two levers: the *efficiency rate* or *utilization rate*. The efficiency rate indicates the maximum processing capacity of machines, and the output control can increase it by merely adding new machines on the shop floor. This intervention would allow a higher production capacity since more machines would work at the same time increasing the output rate. The utilization rate, instead, depends on the workload released on the shop floor. The output control can improve it by enhancing the flexibility of machines. This indeed entails that machines can process more products in terms of variety, which can increase their utilization rate. On the whole, the two alternatives improve performances by better adjusting the capacity of machines. However, their constraint is that they require high capital investments.

Manpower: controlling the output means also increasing and readjusting the production capacity of workers, who can be more or less proficient according to the task they are performing, and the training received on that task. The output control can improve workers production capacity through two main levers, allowing the reallocation of workers among stations or introducing overtime:

- *Workers reallocation*: it is a very simple method to adjust production capacity that does not require new introduction of resources. Reallocating workers means transferring current workers with the shop floor from the station they are idle to the one that is overloaded. Thus, in the overloaded station there could be more than one worker processing jobs. It is a very effective solution that prevents the creation of excessive bottlenecks that can slow down the output rate and cause imbalances of workload among stations. When adopting this production capacity solution, it should be strongly taken into account the differences in efficiency of operators when working in other stations, and consequently the training they need in this regard. Exploiting workers reallocation requires investment in cross-training the workforce that may represent an ongoing expense rather than a one time initial cost.

The cost of the training and the productivity losses due to the training period have also to be considered (Kher & Malhotra, 1994; Fry, Kher & Malhotra, 1995).

- *Overtime*: together with temporary employments could be a solution to adjust production capacity to deal with unforeseen rises in demand and consequent excessive workload to shop floor. However, this method has to comply with regulations, because firms cannot introduce overtime only for the time needed to absorb the extra workload, and at the same time they cannot exploit this lever over a certain threshold (usually 10% of total workers' time). Workers are contractually paid for a minimum amount of hours when overtime is applied. In this concern, Francas (2016) indeed suggests that reallocating workers or recurring to high use of overtime can often be not the best choice when small adjustments may be sufficient to balance the excess of workload.

Subcontracting: eventually the output control can rely on subcontracting part of their production/workload to external manufacturers. Although there is not direct control on the extra production delivered to the supplier, it is the most used short-term solution in case of emergency, since it allows to satisfy peaks in demand without the need to increase production capacity.

These output control leverages bear different costs that influence the decision on which leverage to apply. For instance, working on machines to adjust capacity requires an investment on new tools/equipment that may allow to carry out more tasks (so increasing their flexibility). Workers reallocation, instead, requires providing training to workers. Thus, both solutions entail fixed costs that have a financial long-term impact. Contrarily, overtime and subcontracting imply mainly variable costs. Therefore, the choice of the output control variable becomes a pure strategic decision that is affected also by the characteristics of the demand. In case the demand shows a periodic and steady instability, it seems more reasonable to use long-term solutions such as investing on machines equipment or workforce cross-training. While if there are infrequent demand peaks, overtime or subcontracting may be implemented.

2.3 Input/output control implementation

Input and output control can act on three levels of the workload algorithm: job entry, job release and priority dispatching (Breithaupt, Land, & Nyhuis, 2002; Kingsman & Hendry, 2002; Bergamaschi, Cigolini, Perona, & Portioli, 1997).

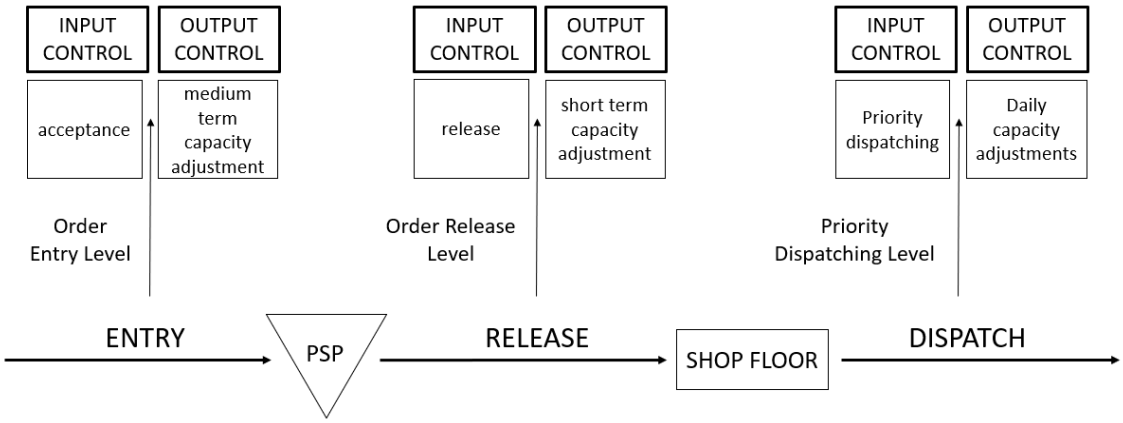


Figure 3 -Input and output control mechanisms (adapted from Breithaupt et al., 2002)

- *Job entry level*: in this first stage the workload control defines whether to accept or not a customer's order. This is a crucial stage because it connects the marketing department to the operations. It is in particular activated the control that analyzing possible prices or promotions for the order, together with the workload in the shop floor, decides whether it is convenient to accept the order and eventually under which circumstances. For example, if the shop floor is already overloaded, the input control may reject or delay orders that will bring an excessive workload to the shop floor or can accept it at a higher price. In case of underloaded stations instead, the input control will try to accept as much orders as possible by proposing discounts to clients (Bergamaschi, Cigolini, Perona, & Portioli, 1997; Kingsman, Tatsiopoulos, & Hendy, 1989).

Many authors (for example Bertolini, Romagnoli, & Zammori, 2015; Moreira, & Alves, 2012; Kingsman & Hendry, 2002) studied the implementation of input control in the very early stage of order entry. Moreira and Alves (2012) developed an algorithm that accepted orders whose impact to station loads did not lead to oversaturation and heavy jobs congestion. Bertolini, Romagnoli and Zammori (2015) instead accepted jobs that did not cause excessive queues in the bottleneck station. Kingsman and Hendry (2002) has been one of the few scholars to apply both input and output control at job entry level. He used the input control in order to confirm only orders that could be

practically delivered to customers within due date, while the output control was implemented through workers reallocation and overtime. This second lever was applied with the key aim to strongly decrease the throughput time of jobs that are not going to be delivered within the contractual due date. The main issue of combining input and output control at job entry level is that the second tends to limit the benefits of the first one (Moreira & Alves, 2012). The output control brings to an unlimited production capacity and then makes the work of the input control superfluous as it always considers constrained capacity. Literature has shown consistent benefits in applying the workload control at the job entry level (i.e. for example the possibility to reject orders), however several academics argued it not be applicable in real industry contexts. In fact, it is improbable that a company will reject orders even if accepting them would lead to a worsening of shop floor performances (Portioli & Tantardini, 2012). In the worst case scenario, firms can negotiate a more flexible or a longer due date in order to accept the order anyway.

- *Order release level*: after orders have been accepted in the first stage, they are recorded in the Pre-Shop Pool (PSP) in order to decide when they are going to be released. An order will be released if its workload does not lead to oversaturate stations loads. However, as opposed to the order entry level, if an order leads to oversaturated stations it will not be rejected, but it is kept in the PSP until the shop floor show an available capacity to maintain its load (Bertolini, Romagnoli, & Zammori, 2015; Bergamaschi, Cigolini, Perona, & Portioli, 1997). In general, the order release stage is the level where the workload control has been implemented and discussed most. Due to the relevance of this stage, it is thoroughly analyzed in the Chapter 3.
- *Priority dispatching*: once jobs have been released from the PSP, they can be subjected to a further and last level of control: the priority dispatching rule. The aim is defining the order and sequence of jobs that are in the same queue in order to fast reduce queue length and better balance stations load. Given that the objective of better balancing loads has already been covered by the input control at Order Release Stage, the benefits of the dispatching rule are quite limited (Land & Gaalman, 1996; Bertolini, Romagnoli, & Zammori, 2015; Bergamaschi, Cigolini, Perona, & Portioli, 1997; Fredendall, Ojha, & Patterson 2010). Despite its marginal advantages, its application has been thoroughly studied in literature, regarding for example which priority dispatching rule needs to be used. In fact, if a proper rule on how to give priority to jobs in queue is not defined, some jobs may be retained in the PSP for a consistent amount of time (Kingsman, & Hendry, 2002). To avoid this issue, it is only needed to apply one of the simplest dispatching rule (i.e. FIFO), without the need for any complex rules.

3. Order review and release

Order Review and Release (ORR) models address how to deal with orders in the workload control at the order release level. In literature many ORR algorithms have been discussed, many carrying their own assumptions. Regardless of the type of ORR algorithm implemented in the order release stage, Land and Gaalman (1995), Sabuncuoglu and Karapmnar (1999), Bertolini, Romagnoli and Zammori (2015) and Land, Stevenson and Thürer (2014) claimed that these models need to have three main functions: a timing, a limiting and a balancing function.

- *Timing function*: the algorithm when releasing order needs to take into account the defined due date. The objective is that every order has to be released so that its completion date meets the due date that has been agreed with the customer.

- *Limiting function*: orders should be released in order to comply with a workload level (called "workload norm") of the stations. There are three types of limiting algorithms: the first one sets an upper bound norm, meaning that an order is released only if its load contribution to the current workload of every station does not exceed that norm. The second algorithm defines a lower bound norm: jobs are released till the workload of stations gets above a defined minimum norm. Finally, the third algorithm sets both an upper and lower bound norm in order to ensure that stations have always jobs to process but not in such way that excessive workloads create overloading.

- *Balancing function*: such type of algorithms aim at releasing jobs from the pre-shop pool that balance stations loads, by avoiding bottlenecks but at the same time also guaranteeing a flow with low station idleness.

Each of these three aspects has an important function for a manufacturing environment. The main challenge for a firm applying workload control is to implement an algorithm that satisfies at best all three functions. There are generally conflicts to apply each function together, and the real objective turns into how to obtain the right balance. Indeed, if an algorithm applies only the timing function, all jobs are going to be released according to their planned release date. As a result of this narrow consideration, it is likely that release of orders will lead to both an overloading (limiting function not respected) or unbalanced stations (balancing function not respected). Limiting algorithms instead take as reference the overall shop floor workload to decide whether to release jobs,

neglecting in this way stations imbalances and pending contractual due dates. Balancing algorithms, finally, may retain jobs in the pre-shop pool even for a long time, especially in cases in which their eventual release would lead to unbalanced stations. For this kinds of jobs, it would probably be difficult to attain to the due date. Similarly, this algorithm could keep releasing jobs to the shop floor merely because stations are balanced, even if there is a current serious overloading. Therefore, these examples show how the single algorithm by itself present conflicts with the other two, which may lead to poor performance. To achieve the best performances, it would be ideally necessary an algorithm that takes into account all three functions at the same time. For practical reasons, the most refined algorithms usually can take into account up to two of these functions, and the trade-off between them is evaluated according to the situation.

3.1 ORR classification

The ORR algorithms discussed in the literature are many and of various types. To properly classify them, it has been adopted the categorization proposed by Bergamaschi, Cigolini, Perona and Portioli (1997). Algorithms are grouped according to two main attributes: the type of function of the algorithm and its time convention:

- *Function adopted*: algorithms can be classified according to their relative function. As discussed in Chapter 3, we can consequently have timing, limiting and balancing algorithms.

- *Time convention*: it is indicated the time in which a workload algorithm considers the release decision. It can be identified two main types of workload control algorithms, the periodic and continuous. Periodic algorithms evaluate whether to release jobs from the pre-shop pool at periodic instants of time, i.e. every day, week, etc. Contrarily, the continuous algorithms perform the release decisions not at fixed interval of time, but it checks constantly and in case they are triggered by specific events. For instance, such algorithm releases a job whenever the workload of a station reaches a lower bound norm or simply when it is idle as there are no jobs to process in its buffer. Moreover, it has also been studied in the literature hybrid algorithms that mix both the periodic and continuous release decision.

Considering the just mentioned attributes, it is presented some of the algorithms most studied and common in literature:

- *Immediate release*: it is not implemented any type of control both at job entry, order release and priority dispatching level. Thus, jobs are immediately released to the shop floor once they have been confirmed by customers.

- *Interval release*: such algorithm does not introduce any type of control, exactly like the immediate release, however, as opposed to the latter, postpones the release of orders in a set frequency (periodic algorithm). Bergamaschi, Cigolini, Perona and Portioli (1997) claim that the interval release is more used than the immediate release, because it better mirrors a real manufacturing context, where indeed orders are not immediately released but are first sent to a production manager that evaluates their release at fixed periods of time.

- *Workload limiting*: This algorithm takes the release decisions at fixed interval of time (periodic algorithm). Then jobs are released only if their workload does not exceed the norm of all the stations that are the routing of that job. Since this model is the one applied in the study of this thesis, it is detailed discussed and explained in Chapter 3.2.

- *Hybrid workload limiting*: It adds to the workload limiting model a further continuous release decision. A job is released not only periodically, but also when it is activated by the event trigger of starvation avoidance. In a nutshell, every time a station is idle, job that has as first machine in its routing that machine idle is released. Thus, in this circumstance, the job is released regardless of the traditional periodic evaluation. Such hybrid algorithm has been implemented by Land and Gaalman (1995), Thürer, Stevenson and Land (2016) and Fernandes, Thürer, Silva and Carmo-Silva (2016).

- *Continuous workload limiting*: this is a pure continuous algorithm that releases jobs when the stations workload are below a defined workload norm. The release continues until station workloads has increased to the set norm. Fernandes, Thürer, Silva and Carmo-Silva (2016) show the application of such model.

- *Balancing*: as opposed to the above algorithms that adopt the timing function, the balancing model follows the balancing function. This algorithm defines a target workload norm that,

on the contrary to the above models, can be exceeded if it alternatively leads to better balanced workloads.

Finally, algorithms based on due dates generally work in a similar way. They implement the timing function by calculating their least feasible release date, and are released accordingly. The difference between these models relies on how the least feasible release date is calculated (see for example Fredendall, Ojham, & Patterson, 2010; Moreira & Alves, 2012).

On the whole, periodic workload algorithms have been demonstrated to perform better in balancing the workload among stations compared to continuous algorithms (Fernandes, Thürer, Silva, & Carmo-Silva, 2016; Bergamaschi, Cigolini, Perona, & Portioli, 1997). The further crucial advantage of periodic over continuous algorithms is that they are much less expensive to implement. In fact, their release decision at fixed interval of time does not require any complex and expensive continuous monitoring system. However, periodic models have the drawback to be less reactive than the continuous ones. For instance, if it is adopted a periodic algorithm and a machine is idle, it can resolve such starvation only in the next periodic decision, which could be the next day or even the next week. Contrarily, a continuous model immediately resolves the machine idleness by releasing a job from the pre-shop pool. The tradeoff between poor reactiveness of periodic algorithms and complexity of continuous algorithms is successfully solved by hybrid models. Fernandes, Silva and Carmo-Silva (2016) showed the benefits of hybrid algorithms most of all for low values of workload norms. In particular, such models lead to lower shop congestion, so decreasing the shop floor throughput time, and a higher utilization of machines/workers due to the activation of the starvation avoidance system. Nevertheless, the literature has illustrated these interesting benefits in the simplest layout of a job shop; in more realistic circumstances of systematic imbalances and flow shops layout it is likely to bring excessive accumulation of jobs in the shop floor.

Different types of algorithms on more realistic shop floor configurations have instead been studied by Sabuncuoglu and Karapmnar (1999), who introduced limited handling system and finite buffers capacity. Their analysis illustrated that continuous models bring lower gross throughput time and tardiness, while the periodic ones perform better in terms of lateness by reducing its standard deviation.

The balancing and limiting algorithms have been compared under a pure flow shop configuration in the study carried out by Portioli and Tantardini (2012). He showed that the balancing algorithm outperforms the limiting algorithm in term of gross throughput time and shop floor throughput time,

two crucial parameters of the workload control concept. Furthermore, the advantage of the balancing model is more accentuated to an increase of the variability of the processing time.

The introduction of due date information on a hybrid limiting algorithm has been studied by Thürer, Land, Stevenson and Fredendall (2017). He found out that integrating due date information on his algorithm strongly reduced gross throughput times for low workload norms. It interestingly discovered that for high norms, instead, introducing due date information leads to an evident worsening of jobs tardiness.

It is necessary to highlight that each paper has tested its results only in one shop configuration and it cannot be assumed their scalability to other configurations, since the workload algorithms are strongly affected by the layout analysed. The conclusions of the academic studies discussed above are indeed dependent on the assumptions made in their respective models.

3.2 Workload limiting

Workload limiting algorithms were first proposed by Bechte (1988), Bobrowski and Park (1989), and the first implementations in literature appear with Land and Gaalman (1995). This method refers to control the released of orders by setting up an upper bound to the workload that can be assigned to each station, directly controlling the maximum level of WIP inventory carried by the system. These algorithms had a high success in workload literature, and have been implemented with different methods. For example different ways to calculate jobs workload have been tested by Oosterman, Land and Gaalman (2000), or the application of dynamical parameters to assess real workload with workload limiting algorithms have been investigated by Perona and Portioli (1996), and integrations of the timing function with due-date information have been tested by Thürer, Land, Stevenson and Fredendall (2017).

Generally, the algorithm follows these main steps:

1. First there is the *sequencing* part, in which all the jobs that are currently in the PSP are evaluated and prioritized according to a sequencing rule. In general it is used an Earliest Due Date rule or a First Come First Served.

2. After this, it starts the *selection* part, in which every job is evaluated for the release starting from the first one that was defined in the sequence. The evaluation of a job proceeds as follows: -

The algorithm considers the workloads of the operations contained in the job's routing, i.e. the different processing times that the job will require to the different workstations that it needs to visit for its production. These workloads will contribute to the loads of these workstations. To calculate the load contribution, it is generally used the method of corrected aggregate load, in which the contribution of the job on a station depends on his relative positioning to the station (in particular, considering the routing of the job, the farther is the station from the current job position, the lower will be the contribution of that job in that station). By summing this value to the current load of the station (left part of the formula below), the total expected load of that station is obtained. This value is then compared to the norm defined for that station (right part of the formula), which represents the maximum amount of workload that can be assigned to that station:

$$\frac{PT_{i,j}}{p_{i,j}} + W_j \leq N_j$$

where, for every station j;

- Pti,j is the processing time of job I at station j;
- pi,j is the position of the station j within the routing of job i (1,2, etc.)
- Wj is the current load of station j
- Nj is the norm of the station j

- If norms are not violated, i.e. the workload of all work centers in the routing of the job plus the contribution of that particular job is smaller or equal than the workload norm assigned to that particular station, the job is released into the shop floor, and its load assigned to the work centers in its routing. If the norm is instead violated for a particular station (see Figure 4 for example), the job is retained in the PSP until the next release period.

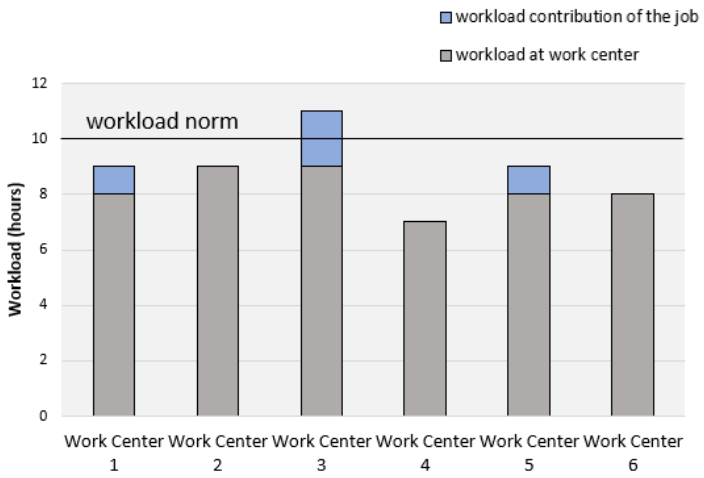


Figure 4 – The function of a workload norm

-Then the algorithm passes to the next job in the PSP with the highest priority and which has still not been evaluated. In case all jobs in the PSP have already been evaluated, the algorithm stops and waits to be reactivated.

Generally, workload limiting algorithms are activated *periodically*, i.e. at the beginning of every shift or once a week. Furthermore, there are also further implementations of workload limiting algorithm that comprehends *starvation avoidance* (Thürer, Stevenson, & Land, 2016; Fernandes, Thürer, Silva, & Carmo-Silva, 2016). In these cases, the algorithm proceeds normally at the indicated period. Anyway, whenever the load of a station drops to zero, the algorithm is activated and the first job in the PSP that has that station as the first position of its routing is released. This is done regardless the system situation, and leads to eliminate situations of temporary starvation.

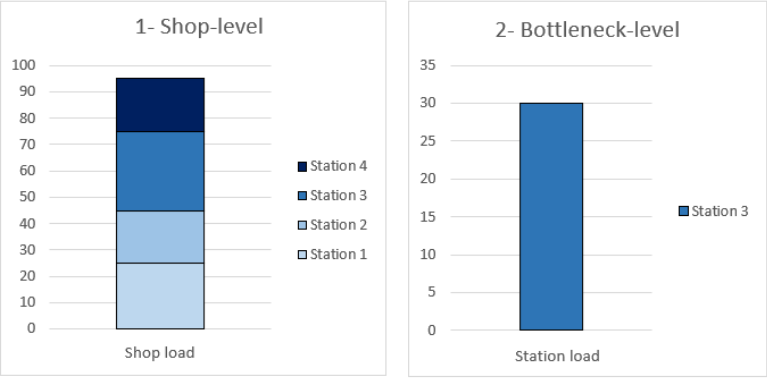
3.3 Methods to calculate the workload

As discussed in the previous section, workload limiting algorithms gradually release orders according to the current load of the system, in particular they check if the load of each station after the release of the order does not exceed a certain predetermined value (workload norm). Thus, in literature different methods to calculate the workload of stations have been widely discussed, and they refer to the following characteristics:

- **The measure adopted**: workload can be expressed as the *number of jobs* that have been assigned to a certain station, or as the *work content* measured as the processing times of the

jobs assigned to that station. The first method is easier to implement since it requires a minimum level of information. Anyway, especially in cases of high processing time variability, considering only the number of jobs can provide unreliable estimates. On the contrary, the second method requires a higher amount of data to be available within the system, and also the possibility to provide reliable estimates of the expected processing times of the jobs in the system. Anyway, thanks to the introduction of informative systems (such as ERP and especially MES) that track production records, it has increased the possibility to have this data available. This is the reason why most authors in workload control use work content as a measure for stations' workload in their models.

- **The aggregation metho**d: it refers to the level of detail for which the workload is considered. Bergamaschi, Cigolini, Perona and Portioli (1997) reviewed different aggregation methods that are used in workload literature. Workload can be aggregated at the *level of total shop* (1), at *the bottleneck level* (2) or at *station level* (3). Case 1 is the easiest to implement, where jobs contribute to the overall workload from the time they enter to the time they exit. With this method, however, the algorithm cannot take into account the current loads of different stations, making it more difficult to avoid station bottlenecks and idleness. Case 2, instead, is useful when bottlenecks have very different performances compared to the rest of the system, and hence they significantly affect the pace of the whole system. Case 3, finally, provides the most complete view but requires pervasive information flows along the system. Production data and real-time time information over jobs position are required in order to calculate the current workload of each station. Most of workload algorithms studied in literature are built at the level of detail of a station. This method is in fact the only one that allows order release algorithms to properly take into account stations load before the release of new jobs (Fredendall, Ojha, & Patterson, 2010).



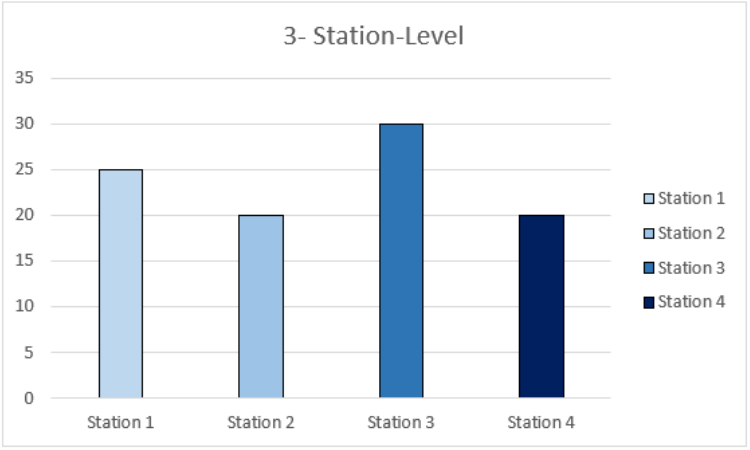


Figure 5 – Different levels of aggregation of the workload

- **Jobs contribution over time**: finally, the last element takes into account how to compute the contribution of the jobs to the workload of the system, considering that they will visit the stations according to their routing and over different periods of time. The main methods used in literature are summarized by Bergamaschi, Cigolini, Perona and Portioli (1997):

  -*Atemporal*: the contribution of a job is computed by summing all its processing times to the loads of the stations it has to visit from the moment it enters to the moment it exits. This is done regardless of the current position of the job within the system, so without considering for example that the job has already been processed by a station. For these reasons, this method gives a rough estimate of the current workload of the system.

  -*Time-bucketing*: tries to compute the expected impact of the job to stations' load over time, by scheduling production of jobs allocating them to a station in a proper time slot. It then estimates times of completion and hence calculates the current workloads of the stations. This method presents some difficulties for implementation since it requires an accurate modeling of the system and of scheduling constraints, in order to define a feasible and optimized sequence of production.

  -*Probabilistic*: this method estimates the contribution to stations' loads by considering all the processing times in the routing of the jobs (similarly to the atemporal approach), but updating the loadings over time and depreciating the future loadings according to the jobs position. This is done to take into account the probability that the job will visit the following station during the next shift. For example, one method commonly used in literature consists in considering 100% of the processing time of the job to the load of its current station, and then to decrease its contribution to 50% for the second station in the job's routing, 33% for

the third and so on. Probabilistic methods are vastly used in literature since they provide a proper trade-off between the consideration of future workloads but also without overcomplicating the model.

Stations in jobs routing	1	2	3	4
Current job position	visited	current	to visit	to visit
Job processing time (min)	20	30	40	30
Contribution to station load (min)	0	30	20	10
Allocated % of processing time	0%	100%	50%	33%

Table 1 – Allocation of workload depending of jobs position

Some critics to probabilistic methods were raised by Land and Gaalman (1995), and they refer to the following reasons: (1) the method assumes that a job can be split into smaller parts, since it takes into consideration a percentage of the processing time; (2) it doesn't take into consideration the effect of the status of the system on the processing times, as it is expected that a higher saturated shop would result in higher time in queues and hence a lower contribution the following stations. Despite this, the benefits of using this method have been extensively proven by Oosterman, Land and Gaalman (2000), and as they provide the best practical trade-off to calculate the impact of workload in a system they are widely used in literature. This method will also be the one used in our thesis.

3.4 Methods to calculate the workload norm

Calculating the workload of the stations means assessing the expected level of congestion that the system will have once the orders are released. In the previous section we have discussed the most important characteristics to consider in this process. When deciding whether an order has to be released in the system or not, the workload limiting algorithm first calculates the workload of that station considering also the contribution given by that job, then checks whether the value of workload obtained exceeds a certain threshold. This threshold is called *workload norm*.

Setting the proper norm is a very delicate job, since if the norm is too strict (i.e. the value set too low), few jobs will be released within the system, with an expected increase in the gross throughput time (jobs spending much time in queue in the PSP). There is also the risk that some jobs that require high working time may be indefinitely postponed to production due not to exceed the norm. On the other side, if norms are set too loose (i.e. the value is too high), the workload control algorithm will release high amounts of orders into the system, which would likely cause congestions and long queues. This would result in a sharp increase of the shop throughput time (time spent within the production system after the release), with a consequent increase in WIP and length of queues. It is worth noting that with very high norms, the functioning of the workload limiting algorithm is basically similar to an immediate release algorithm, where orders are released at the moment of their arrival and being the norm not a real limit to job release anymore. Anyway, since WIP reduction and system balancing are the two objectives of workload control for the reasons explained in Chapter 2, workload limiting algorithms must be set within the proper norms in order to obtain a smooth flow and preventing bottlenecks.

Despite the importance of setting the proper norms in the system, in literature it has not been defined a unique framework to calculate them. Thürer (2011) investigated a structured method to assess the optimal norms. The result was that norms are strongly affected by shop floor characteristics, such as the number of work centers and the presence of a dominant flow direction. Generally, researchers use a trial-and-error approach as explained in Land and Gaalman (1995), in which they set an initial low value of the norm, which must be close to a situation of no-control (i.e. close to a situation in which the heaviest jobs are never being released, spending infinite time in queues). So the setting of the norm starts from the first norm being under control. Then, more levels of the norms are considered by increasing it up to the point that brings to a worsening in performances. In this way it will be obtained a high and a low norm, that will be the boundaries of the investigation. Different norms with values between these two up and down limits will be considered, according to the level of detail seek. This is the method that has been followed in the study of this thesis.

4. Shop configuration

Another aspect which strongly influences the Order Review and Release system is the shop configuration. The layout of the production process, indeed, determines how a company carries out the operations in order to deliver products to customers. The benefits of the workload control are by consequence affected by the type of shop configuration where it is implemented (Sabuncuoglu, 1999). Workload control is typically studied in the layout characterizing high-variety low-volume MTO companies: the job shop.

Osterman (2000) identifies four types of shop configurations and classifies them according to two main parameters: the *routing length* and the *flow direction*. The routing length indicates the number of stations a job has to be processed in a specific shop model and it can either be constant or variable. If the routing length is constant, all jobs are going to visit the same number of stations; while under the variable case, each job has its own number of stations where it is going to be processed. The flow direction expresses the routing sequence of jobs in a particular job shop model that defines the direction of the flow, which can be directed or undirected. In the directed, jobs share the same routing sequence, whilst in the undirected each job has its specific sequence of routing.

		Routing length	
		Variable	Constant
Flow	Undirected	Pure job shop (JS)	Restricted job shop (RJS)
	Directed	General flow shop (GFS)	Pure flow shop (FS)

Figure 6 – Shop configurations (adapted from Osterman, 2000)

It is now presented all the four job shop models identified by Osterman (2000):

- *Pure job shop*: in this layout there is an undirected flow of jobs which are processed in a random number of stations. The lack of both a standard flow and routing makes the pure job shop the most difficult layout to manage, since it is complicated the identification of the bottleneck station (Fernandes, 2016; Sabuncuoglu, 1999; Bertolini, 2015; Thürer, 2017; Land, 2015; Land, 2014; Moreira, & Alves, 2012; Kingsman, 2002; Oosterman, 2000).

- *Pure flow shops*: among the four models, pure flow shops are the least complicated because there is both a directed flow and a fixed routing length. As a matter of fact, jobs cannot be distinguished in the shop floor as having the same routing, this creates a repetitive flow that helps managers in planning and controlling activities.

- *Restricted job shops* and *generalized flow shops*: these two shop configurations are in the middle of the two extremes (pure job shop and flow shops). In the restricted job shops, orders visit the same number of stations but in diverse order. While in the generalized flow shops, orders follow the same direction of flow in the shop floor, but have their own specific number of stations to visit.

The shop configuration most analyzed in workload control literature is the pure job shop (Kundu, 2016). Nevertheless, a new stream of research (e.g. Oosterman, Land, & Gaalma, 2000; Portioli & Tantardini, 2012; Miragliotta & Perona, 2010; McCreery & Krajewski, 1999) has started applying WLC methods to flow-shops configurations. They indeed claim that, generally, shops of MTO companies present a sort of dominant flow in production. For example, there are some activities that are always carried out at the beginning, such as initial quality controls, or first raw material processing. While other activities, such as high temperature treatments, are generally carried out at the end. The identification of a starting and ending point in a shop floor helps detecting a steady flow in production. Moreover, even low-volume high-variety MTO companies have more and more streamlined their production flow through the application of lean philosophy. Therefore, shop configurations can be modeled as a flow-shop (Portioli & Tantardini, 2012).

4.1 Shop configuration applications

After having explained the new research stream focused on flow-shop configurations, it is worth noting that there are other issues for which WLC has not been widely applied by practitioners. Bertolini, Romagnoli and Zamorri (2015), Yuan (2017), Thürer, et al. (2012) and Miragliotta and Perona (2000) argue that many simplifications made in the simulated models are far from reality. These simplifications also regard the shop configurations, such as the general assumption made by most academics of “one station = one machine” (see for example Land, Stevenson, Thürer, & Gaalman, 2015; Thürer, Stevenson, Land, & Fredendall, 2018; Park & Bobrowski, 1989). Such

simplification does not reflect real job shop configurations, as each station is often made up by similar or equal machines (Henrich, Land, & Gaalman 2006; Stevenson, 2006). It is crucial to consider that more machines present within a station not only because it mirrors practical shops, but also because it allows to properly take into account the frequent imbalances among those machines.

Some authors that have considered in their workload control models more than one machine per station are: Felan and Fry (2001), Yue, Slomp, Molleman and Van Der Zee (2008), Weng, Wu, Qi and Zheng (2008), Sagawa and Land (2018), Ruiz-Torres and Mahmoodi (2007), Bokhorst and Gaalman (2009) and Malhotra, Fry, Kher and Donohue (1993).

4.2 Parallel machines in a station

The consideration of more than one machine per station in turn entails a key layout characteristic which has not been regarded in the workload control literature: the presence of parallel machines in a shop station. In fact, a job shop is by definition made up by stations singularly performing the same or similar operation (one station for milling, one station for mixing, etc.), hence machines present in station of a job shop are perfectly in parallel, since they perform similar or equal tasks.

The lack of consideration of parallel machines in the workload control literature can be one of the reasons why such concept has been scarcely applied in real cases (Miragliotta & Perona, 2000). In order to increase the likelihood that workload control is going to be implemented by practitioners, it seems necessary to study a simulated model as much as close to the reality production contexts. To this aim, it should not be neglected by literature the presence of parallel machines in a station of a shop.

Moreover, it is interesting studying the impact of workload control on parallel machines, since the latter face unbalances issues and thus are likely to be the bottleneck that could constrain order release from the pre-shop pool. (Bokhorst & Gaalman, 2009; Sirikrai & Yenradee, 2006). The presence of unbalances within two parallel machines is due to usual different processing time of such machines, that is in turn caused by the different ages of two parallel machines. In fact, it is likely one parallel machine to be newer than the other, hence having a lower processing time. Most of studies considering parallel machines under the workload control implemented have however assumed for simplicity that their processing time was equal (Felan & Fry, 2001; Yue, Slomp,

Molleman, & Van Der Zee, 2008). Weng, Wu, Qi and Zheng (2008) have been one of the few academics respecting the differences in processing time of parallel machines.

On the whole, given the importance of studying the impact of workload control on layout respecting real shop configurations, the thesis has analysed the impact of workload control on a flow shop with parallel machines, considering both the case with similar and the case with different processing times.

5. Routing flexibility

The consideration of more machines per station and the consequent presence of parallel machines in the workload control study open a new chapter: the routing flexibility. Routing flexibility is an advantage typical of job shops and consists on the flexibility of jobs in changing their initial routing (Shewchuk, 1998). If the release of a job in the shop floor leads to machines unbalances or excessive queues, it can be changed its routing to an alternative one which instead brings better shop floor performances. Thus, under routing flexibility, a job does not have only its standard routing, but it can also exploit an alternative routing to be activated according to specific current shop floor performances.

The change of routing of a job is applied to equal or similar machines performing the same operation. The traditional implementation of routing flexibility is within parallel machines. In fact, if a job is supposed to be processed by a machine which is oversaturated, it can be used the routing flexibility of such job, in order to change its routing towards a machine parallel to the first one that instead is undersaturated.

The concept of routing flexibility is usually applied and perfectly fits in the job shop production layout for two main reasons. First, the requirement of the routing flexibility on similarity of machines is strongly respected by a job shop. The latter is indeed made up by "departments" or "stations" where are present similar machines performing the same function, this allows the exploitation of routing flexibility within each station of a job shop.

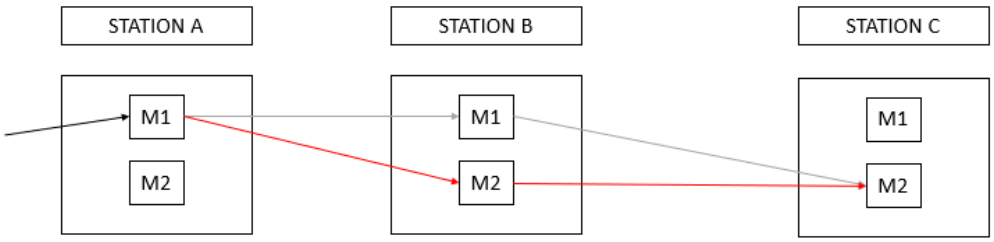


Figure 7 – An example of routing flexibility

The image above shows an example of the application of routing flexibility in one of the three stations (A, B, C) of a job shop. It is in particular applied the change of route in the station B, where there are two machines in parallel. After the job has been processed by the station A, it is going to be worked by the machine M1 of the station B. However, given such machine is oversaturated, the

job changes its route towards the machine M2 of the station B which does not have oversaturation issues.

The second reason that makes the job shop configuration constantly applicable by the routing flexibility is the presence of frequent imbalances within machines of a station in terms of machine saturation and jobs congestion (Kundu, 2016). The lack of a standard flow and the demand variations affecting job shop causes imbalances among machines of the same station and thus more opportunity to balance the capacity of those machines through routing flexibility.

On the whole, the routing flexibility has the advantage to smooth production flows by rebalancing the machine saturation and limiting the creation of excessive bottlenecks and to guarantee a continuous production by avoiding machine breakdowns. About the last advantage, in case a shop floor has a high throughput rate, it is not easy to change the flow of pieces towards the non broken machines if it is not implemented an automatic routing flexibility system.

The routing flexibility has then proven to be a valid capacity adjustment method, but from a broad view it is also seen as a system's flexibility to adapt changes in volume or capability (Chan, 2001). The relevance of the routing flexibility can be better understood if it is beard in mind that it is considered the second category of flexibility on the more and more discussed Flexible Manufacturing System (FMS). Changing the route of a job can express the ability of a FMS to handle even large-scale changes like consistent increase in production volume.

5.1 Routing flexibility in the workload control

Despite the importance of the routing flexibility and its application in the layout (job-shop) where it also applied the workload control, it has been poorly discussed in the workload control literature. Nevertheless, it appears interesting analyzing how routing flexibility impacts the performances usually studied in the workload control literature. In fact, the implementation of routing flexibility may positively affect the two main workload control KPIs: the Shop Floor Time (SFT, i.e. the throughput time of jobs in the shop floor) and the Gross Throughput Time (GTT, i.e. the SFT plus the time jobs wait in the pre-shop pool before being released).

More specifically, the routing flexibility impacts the SFT, since it reduces buffer queues and the consequent average time jobs spend queuing. For instance, thanks to routing flexibility, a job that is waiting in a long queue to be worked by its default machine can change the routing to a similar

machine performing the same process, that instead does not have any jobs queuing. As a result, the time such job has spent in queue can strongly decrease. In general, the routing flexibility leads to a smoother and better balanced flow of jobs in the shop floor that can decrease the SFT.

In what concerns the impacts of routing flexibility on the GTT, the former has an indirect impact on the latter. As a matter of fact, the better balance of jobs among machines decreases the probability to have oversaturated machines that hinder the release of jobs from the pre-shop pool. This is especially true for the workload limiting algorithm, since the latter control the workload of each machine and, if there is at least one machine that exceeds the workload norm, the job is withheld in the pre-shop pool. Therefore, the presence of only one machine oversaturated causes the non release of jobs to the shop floor that will be evaluated to be released in the next cycle of the limiting algorithm. If the evaluation period is not frequent (such as the weekly one), the presence of an excessive bottleneck machine causes further imbalances among stations because jobs cannot be released until the next cycle. This drawback of the limiting algorithm can be solved through the application of routing flexibility. As a matter of fact, if there is a bottleneck machine, the reallocation of its jobs in other machines – that can be triggered thanks to routing flexibility – is going to decrease the load of that machine. It is then avoided that jobs are not released due to a single oversaturated machine. As a consequence, jobs decrease their average time waiting in the pre-shop pool (i.e. GTT reduction) since more jobs are released to the shop floor thanks to the implementation of routing flexibility.

Nevertheless, the workload control literature has poorly considered the routing flexibility (Thürer, Stevenson, & Silva, 2011; Henrich, Land, & Gaalman, 2007; Zhao, Gao, Chen & Xu, 2015, Stevenson, 2006, Fernandes & Carmo-Silva, 2011). This is likely to derive from the simplistic assumption made by researchers on considering only one machine per station. Under this assumption, is neglected the opportunity for changing the routing of jobs in the shop floor.

Among the papers studying routing flexibility it is worth mentioning the analysis carried out by Henrich, Land and Gaalman (2007), Henrich, Land and Gaalman (2006), Fernandes and Carmo-Silva (2011) and Zhao, Gao, Chen and Xu (2015). Henrich, Land and Gaalman (2007) studied how workload control benefited from the implementation of routing flexibility. The paper studied a job shop with 7 machines, 2 of which were parallel machines where it was exploited the routing flexibility. Even if the routing flexibility was implemented for just one station, the benefits were consistent in terms both of SFT and GTT. Furthermore, the routing flexibility showed relevant reductions of those KPIs only with a limited power – called “interchangeability” as it will be later explained – of the algorithm.

5.2 Routing flexibility parameters

The implementation of the routing flexibility concept on a workload control system requires to necessarily address three parameters: the level of interchangeability, the routing decision and the grouping machines decision.

-**Level of machines interchangeability**: interchangeability of machines indicates the ability of machines to perform similar operations (Henrich, Land, & Gaalman, 2007). If two machines are perfectly equal and perform the same task, their respective level of interchangeability is 100%, because they can process the same product types. In such a case, jobs can be indistinctly worked either in the first or in the second machine. Thus, all jobs can change their routing within the two 100% interchangeable machines. Contrarily, the semi-interchangeability indicates that two machines are similar but not fully interchangeable. To better explain the semi-interchangeability, it is taken the example proposed by Henrich, Land and Gaalman (2007). The latter considers two machines, machine A and machine B, that operate the task on the whole. However, machine A can work products within the range of 20 and 70 mm, while machine B the ones within 5 and 55 mm as shown in Figure 8.

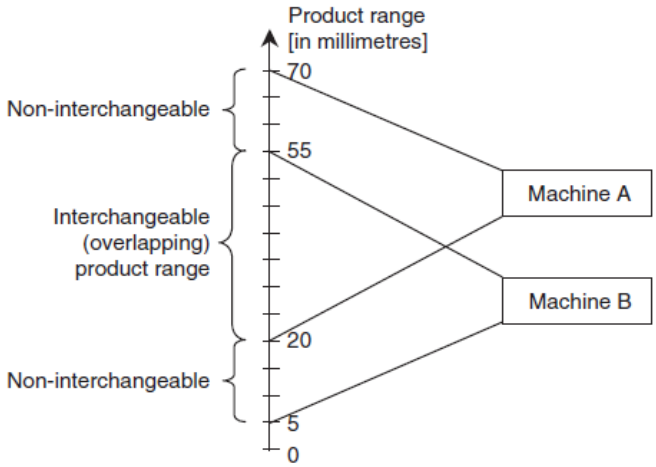


Figure 8 – Jobs' interchangeability (adapted from Henrich, Land and Gaalman, 2007)

As it can be noticed from the illustration above, machine A and B can both work products that are in the common range of 20 – 55 mm. Therefore, the two machines are not fully interchangeable, but they are semi-interchangeable for products falling in the common range. From a routing flexibility perspective, it entails that not all products can exploit their change of routing within machine A and

B, but only the products that have a range within 20 and 55 mm. In other words, the power of routing flexibility is limited with not fully interchangeable machines, because only a percentage of products can take advantages from routing flexibility.

The presence of semi-interchangeable machines is the most frequent among similar machines (Zhao, Gao, Chen & Xu, 2015). In fact, it is unlikely that even two apparent identical machines can indistinctly process the same product types, hence the it is difficult to have a condition of full interchangeability (Henrich, Land, & Gaalman, 2007). Machines processing the same operation are much more likely to be semi-interchangeable for several reasons, such as the differences in: power of the machines, number of axes, tolerances adopted, etc. If it is for instance considered two parallel machines in a station, it is improbable that they will be exactly the same (full interchangeability), since usually one machine has been acquired later or is more technologically advanced than the other. Semi-interchangeable machines can also differ in what concerns operational characteristics such as difference in processing times, set-ups and operator's skills (Henrich, Land, & Gaalman, 2007). Finally, the last level of interchangeability is the no-interchangeability, which means that it cannot be exploited the routing flexibility, since each job can be worked only in its default machine.

Researchers have demonstrated in many studies (see for example Henrich, Land and Gaalman (2007), Zhao, Gao, Chen and Xu (2015) and Fernandes and Carmo-Silva (2011)) increasing the level of interchangeability more than 50% leads to a poor marginal impact in terms of reduction of GTT and SFT. The most benefits of routing flexibility are already gained within a moderate level of interchangeability, around 20-30%.

-**Routing decision**: this leverage concerns at which stage of workload control to change the routing of jobs (i.e. apply the routing flexibility). There are two options to consider about when to activate the change of routing: at order release or at time of dispatching.

- *Order release*: taking the routing decision at order release entails that it is evaluated in the pre-shop pool stage in which interchangeable machines a job has to be processed. Thus, before being released, the workload control algorithms decides not only whether to release the job, but also the routing in the interchangeable machines. It needs to be highlighted that the routing for the other non-interchangeable machines is decided before the pre-shop pool level.

- *Dispatching*: taking the decision of the routing within interchangeable machines at the dispatching level means to postpone the routing decision at the shop floor. After jobs have been released, they do not know in which interchangeable machine to be processed. Such decision is taken in the buffer immediately before the interchangeable machines. Postponing the routing

decision at dispatching has a higher potential of balancing the workload among interchangeable machines in respect to taking the decision at order release. As a matter of fact, at order release level there is a less accurate view on the workload within interchangeable machines. While at the dispatching level it can be estimated the most updated values of workload or congestion among those machines. It may happen that the level of balance of interchangeable machines estimated at order release is completely overturned when the job is at the shop floor shop. Such temporary imbalances are detected only by taking the routing decision at the shop floor. Furthermore, the routing decision at dispatching can benefit from the pooling synergy effect (Henrich, Land, & Gaalman, 2007). The latter relates to the opportunity derived from routing flexibility to allow jobs in the queue before interchangeable machines to be allocated to first available machine. In a nutshell, it is avoided that a job firstly allocated to an oversaturated machine waits in queue while another interchangeable machine is immediately available. Thus, the pooling synergy reduces the time jobs wait in the queues preceding interchangeable machines.

-Grouping decision: this leverage indicates the decision if the interchangeable machines have to be grouped and considered as a unique machine or instead, they can be regarded as different machines. In the first case, the interchangeable machines have a unique workload norm, while in the second case each of them has its own workload norm. Such decision has an important effect on the workload control algorithm. In fact, if the interchangeable machines are grouped into a common and unique workload norm, when evaluating the release of jobs – under the limiting algorithm – it needs to be checked only one equation for such machines. Whilst if they are non grouped, the algorithm checks the equation for each interchangeable machine.

Grouping machines into a single workload norm does not require detailed load information on each interchangeable machine, but it has the drawback to lead to temporary overload or underload for those machine types (Henrich, Land, & Gaalman, 2007). It may happen that consecutive jobs are released just because they do not exceed the common workload norm of the interchangeable machines, but they are going to be worked only by one of them which is constantly oversaturated. Instead, considering the single workload norm of each interchangeable machine avoids this inconvenient occurrence.

Given the grouping decision is directly impacted by the routing decision, it is analyzed the combinations of those decisions.

Grouping decision	Routeing decision	
	One capacity group (one norm)	Two capacity groups (two norms)
At order release	(I)	(III)
At dispatching	(II)	(IV)

Figure 9 – Configurations of routing and grouping decisions (adapted from Henrich et al., 2004)

- *Routing decision at order release + grouping of interchangeable machines into one capacity group*: This combination is quite controversial, because, in order to decide the routing at order release level, it is needed workload information for each interchangeable machine. However, this information is disregarded when it is assumed the grouping into one common workload norm. As a result, this combination has shown weak performances in the shop floor (Henrich et al., 2004).

- *Routing decision at dispatching + grouping of interchangeable machines into one capacity group:* in this scenario, after jobs have been released according to the grouping condition and are to be processed by interchangeable machines, they are placed in a common queue where the routing decision is taken.

- *Routing decision at order release + non grouping of interchangeable machines:* the consideration of single norms for interchangeable machines allows a better balanced workload among them.

- *Routing decision at dispatching + non grouping of interchangeable machines:* on one hand, it is the scenario with most balancing opportunities, on the other hand it requires most detailed information of workload in the shop floor.

Henrich, Land and Gaalman (2007) showed that choosing the routing at dispatching and non grouping interchangeable machines outperforms the other scenarios, regardless of the level of interchangeability. Its benefits in terms of GTT and SFT and the balancing of workload are even more highlighted for high levels of interchangeability. While the pooling synergy effect is limited for an interchangeability lower than 15%, because there are less jobs to be potentially reallocated within interchangeable machines. The effect of pooling synergy effect are much more pronounced for high level of workload norms, because more jobs are released to the shop floor and the

consequent increase of jobs in queues enable the more frequent activation of pooling synergy. Furthermore, Henrich, Land and Gaalman (2007) and Fernandes and Carmo-Silva (2011) show how the non-grouping decision becomes more effective with low level of workload norms.

On the whole, the studies of routing flexibility on workload control algorithm has focused mainly on understanding the decision levers to apply, such as the level of interchangeability, routing and grouping decision. Instead, it has been poorly analyzed how the possibility to change routing of jobs affects the performances of a job shop where it is implemented the workload control. It lacks not only the impact of routing flexibility on key parameters such as GTT and SFT, but also on other important KPIs such as machine or worker saturation and the amount of jobs waiting in queues. All these KPIs should theoretically be affected by the routing flexibility and it seems worthwhile thoroughly studying them. Therefore, besides minimizing the routing flexibility itself, the thesis aims at conducting an accurate study on the above mentioned parameters in order to understand their proper correlation with the change of jobs' routing.

5.3 Routing flexibility application

The application of routing flexibility is meant to improve system performance in case of irregular, independent arrivals of jobs, which are often characteristics of the demand faced by MTOs companies. The goal of this application is indeed to solve the imbalances in the system in order to obtain a smooth and leaner production flow. The context in which routing flexibility can be applied must include parallel machines or manufacturing cells in which the same activities can be performed by different machines. Then, the level of interchangeability must be sufficient to cover the different tasks required by the job.

Regarding the characteristics of the production system, routing flexibility can be applied both in a *machine-intensive* and in a *worker-intensive* environment (Bokhorst, Slomp, & Gaalman, 2006). The first is the case for example of flexible manufacturing work cells (FMCs), in which the order of operations and the routing of the parts can be adjusted for optimization purposes (Ramirez, Zhu, & Benhabib, 1999). Routing flexibility in this case refers to changing the production flow that directly feeds one machine with another. This system is generally also capital intensive, as it requires not only CNC machines but also an advanced transport and control systems (Košťál, Peter, & Velisek, 2011). Anyway, with the spread of flexible manufacturing systems configurations, the diffusion of these kinds of systems has become more pervasive in the recent years.

The second case instead refers to production systems that strongly rely on manual labor, such as cellular manufacturing or manual assembly cells for low-volume products and in post-automated assembly for odd-form components (Mitzner, 2009). Routing flexibility is done in this case by assigning more jobs to those cells that have available capacity at the moment (Bokhorst, Slomp, & Gaalman, 2006).

While for machine-intensive environments routing flexibility is one of the methods commonly used to obtain a balanced production flow, instead for worker-intensive environments there is another possible solution that is commonly adopted to balance production capacity: the reallocation of workers between machines and cells, which will be explained in the next chapter.

6. Workers reallocation

Most of the workload control literature has assumed stations of a job shop to have fixed capacity that do not have limits in performing the operations needed to produce a job. Nevertheless, in a real production system – apart from fully automated machines – machines cannot work without workers. Workers play a crucial role most of all in job shops, where machines perform specific tasks in a station, and it is difficult to track a repetitive and standard flow within the shop floor. As a consequence, in a job shop, machines are frequently not fully utilized and then it is not cost-effective to keep them active for the full shift of a typical production day. In this scenario of frequent machines underutilization, the number of workers in a job shop is likely to be lower than the number of machines. In other words, workers need to work on more than one machine. Therefore, as opposed to most of the literature, machines do not have a fixed production capacity, but their performance and the one of the overall system strongly depends on workers and on their interaction with machines. The impact of workers on workload control systems has been poorly analyzed, but it seems more realistic to take them into account when approaching such studies.

Considering workers on the workload control literature means to analyze the reallocation of workers as an output capacity control tool (Chapter 1). Bartłomiej and Przemysław (2016) stress the importance of studying workers reallocation, pointing out that it is the only output capacity control tool which can potentially applied by every manufacturing company, regardless of their size or financial resources, with a limited investment on operators' training. Instead, the other output capacity control tools – purchasing of new machines, subcontracting and overtime – require more consistent financial resources.

The reallocation of workers can be exploited in a job shop, because the latter experiences frequent machines undersaturation or oversaturation that can be limited or solved through the transfer of workers from underloaded stations to overloaded ones. This enables to fully utilize the capacity of workers and consequently to create a more balanced flow in the shop floor.

6.1 Leverages of workers reallocation

The reallocation of workers under workload control system entails to consider three different rules: the "where", "when", "who" rule.

Where rule

The where rule literally decides where the workers have to be transferred during the reallocation time. Each worker is assigned to a standard machine where he spends most of the time, however, if it is triggered the event that activates the workers reallocation (e.g. when rule), he is going to be transferred to the station defined by the where rule. Hottenstein and Bowman (1998) and Sammarco, Neumann and Lambiase (2014) defined different where rules here below listed:

1. *Random* (RND) – The machine the worker has to transferred to during its reallocation is randomly chosen.
2. *First Come, First Served* (FCFS) - A worker is transferred to the machine with first come job in queue.
3. *First in System, First Served* (FISFS) – A worker is transferred to the machine with first in system, first served job in queue.
4. *Shortest Processing Time* (SOT) – A worker is transferred to the machine whose first job has the shortest operation time.
5. *Longest Number of jobs Queuing* (LNQ) – A worker is transferred to the machine whose buffer has the highest number of jobs.
6. *Largest Total Processing Time* (LTPT) – A worker is transferred to the machine whose buffer has the largest total processing time of jobs queuing.
7. *Worker efficiency* (WE) – A worker is transferred to the machine where he is most efficient. To every worker is assigned a level of efficiency for each machine.

Hottenstein and Bowman (1998) analyzed these rules and showed that the WE rule outperformed all the other rules. In general, each rule acts on different performance of a shop floor. The FISFS has been demonstrated to be the best in reducing the variance of the throughput time. The LNQ instead stood out from the others since it was the one reducing most the average throughput time of jobs and the idleness of workers. While the LTPT also strongly reduces such idleness but not as the LNQ. However, the LTPT was the rule most improving the delivery performance.

Nevertheless, the studies made by workload literature on the different when rule has been conducted on a pure job shop layout. Thus, the results just discussed above are not replicable for more realistic layout such as flow shop. Sammarco, Neumann and Lambiase (2014) and Bokhorst, Slomp and Gaalman (2004) indeed claim that the analysis on job shop may have the limit to be misguided, because such type of production layout does not strictly reflect real industrial shop configurations. Thus, if it is needed to analyze more realistic layout such as flow shop, Sammarco (2014) analyzed

much simpler and more pragmatic when rules. These rules addressed for a flow shop configuration are:

1. *Upstream* (UST): a worker is transferred to the most upstream machine.
2. *Downstream* (DNS): a worker is transferred to the most downstream machine.
3. *Closest* (CLS): a worker is transferred to the closest machine.
4. *Max-time* (MXT): a worker is transferred to the machine. Which has the longest operation time.
5. *Min-time* (MNT): a worker is transferred to the machine. Which has the slowest operation time.

Sammarco, Neumann and Lambiase (2014) realized that the DNS rule was the best among the others in terms of throughput time and WIP reduction. In general, it can be easily noticed how such rules applicable for flow shop can be implemented and fully adopted even by shop floor without system that calculated on time performance of machines. The UST, DNS and MCT rule are straightforward and immediately applicable, while the MXT and MNT just require the knowledge of the machine that has the longest/slowest operation time, this information is typically known even from experience in a real shop configuration.

When rule

The when rule defines when a worker has to be reallocated to a machine. This rule elaborates a triggering event that when occurred activates the transfer of workers. The literature has extensively analyzed such a rule, identifying mainly two types of when rules: the centralized and decentralized rule.

- *Centralized rule***:** according to the centralized rule, a worker is transferred to a machine only after he has completed the job on its default machine (Darwin, Hemant, & Wagner, 2009; Sammarco, Neumann, & Lambiase, 2014). Even if there is a strong imbalance of saturation of machines and hence the transfer of that worker is urgent, he has to first complete the current job on its standard machine before being reallocated.

- *Decentralized rule*: the decentralized rule allows a worker to be transferred only if there are no jobs to be processed in its default machine. Thus, if a worker's reallocation is needed and there are still jobs to be processed in his default machine, he cannot temporarily be moved to another machine. This worker has to process all the jobs queueing in his default machine before being reallocated. Given the time spent in processing all the jobs before moving to the machine oversaturated, it could even happen that when the worker is finally

ready to be moved it is no more needed its help. While finishing processing the jobs on its default machine, the other machine could have solved the saturation issue.

The main difference between the centralized and decentralized rule is that the former allows a more frequent transfer of jobs due to a less stringent movement condition, while the latter ensures that workers are transferred only if they are idle, so limiting their overall movements. In this concern, the more transfers allowed by the centralized rule may be counterproductive if the machines where workers move the most are far away from each other. This is due to the impact of transfer time that may even fully cover that advantages of reallocating workers (Uzun & Latif, 2010). In other words, the reduction of throughput time, deriving from a better workload balance gained thanks to reallocation of workers, could be lower than to the time workers wasted in moving during their reallocation.

Although the decentralized rule constraints the number of workers reallocation, limiting also the transfer time impact, it may excessively retain workers on their default machine for an excessive amount time to the disadvantage of a better balance workload driven by workers reallocation. In fact, it could be more efficient to transfer a worker to an oversaturated machine, even if he is not 100% idle in his default machine. The decentralized rule does not consider this tradeoff between "level" of idleness and the level of oversaturation of a machine requiring urgent help to process all its jobs queuing.

In the workload control simulation model of Sammarco Neumann and Lambiase (2014), the decentralized rule reduced the shop flow time 15% more than the centralized one. Similarly, Hottenstein and Bowman (1998) showed that the centralized rule did not lead to strong performance improvement in its simulation study. However, such results are highly affected by the parameters considered during the simulation and the shop floor characteristics, so they are not really scalable to other production layouts. More specifically, the performance of the centralized and decentralized rule depends whether the transfer time during reallocation is considered, and eventually how much, and the level of imbalances among machines.

Who rule

The "who" rule defines the worker who is going to be transferred to machines different from his default one. Such rule is the least considered in the workload control literature in respect to the where and when rule (Felan & Fry, 2001). Most researchers only focus on the when and where,

since their decision often indirectly imply who is going to be reallocated. This holds true if workers have all the same level of skills on operating on all the machines. However, such circumstance is likely to rarely happen in a real shop floor contexts, where there are some workers who are more experienced than others and then are more efficient on some machines. If for instance there is an oversaturated machine and two idle workers available to be transferred to that machine, it is reasonable to send the worker who has best skill at such external machine, instead of randomly choosing one of the – as it happens when the who rule is not considered.

Taking into account the who rule entails to consider the skills of a worker that can be described by two parameters: worker flexibility and worker efficiency.

- *Worker flexibility*: this attribute expresses the capacity of a worker to work on different machines. It can be identified two type of workers: the single-skilled and multi-skilled. The former is able to work only on his default machine and cannot be transferred to other machines. While the latter, besides his standard machine, has the ability to work on diverse machines and can be moved during the reallocation process of the workload control algorithm. The reallocation of the multi-skilled worker enables to reduce his idleness and enhance his productivity (Bartłomiej & Przemysław, 2015; Felan & Fry, 2001).

- *Worker efficiency*: it literally indicates the efficiency of a worker to process a job on a specific machine. Each worker has a value of efficiency on all the machines he is capable to operate. The processing time of a job does not merely depend on the pure characteristics of the job or of the machine where it is processed, but also on the efficiency of an operator on the machine. To better understand it, it is given the following example: a job that is going to be processed on a particular machine has a processing time of 1 minute. If it is supposed that the machine does not breakdown, the processing time of such job depends also on the efficiency of the worker on that machine. If he has an efficiency of 100%, the processing time of the job will be 1 minute. While if his efficiency is just 50%, the processing time of the job will be 2 minutes (1/0.5). Therefore, to find the real processing time of a job in a machine, it has to be divided by the efficiency of the worker operating on that machine (Valeva, Hewitt, Thomas, & Brown, 2017).

Both the flexibility and efficiency parameters of a worker are not fixed values, but they can be increased or decreased. The efficiency of a worker can be improved with the experience and it is particularly for machines requiring repetitive tasks. The flexibility of workers can be primarily enhanced through cross-training activities. Training workers to operate in different machines is a

crucial choice for a firm because it requires to understand who to train and most of all in how many machines. Higher are the machines a worker is trained to work to, higher is the investment in terms of time and cost pursued by the enterprise. It is likely that a worker takes more training sessions before being fully operational on all the machines he has been trained to. Furthermore, excessively increasing the level of flexibility may lead to a low level of efficiency on the trained machines that does not justify the cross-training investments.

Felan and Fry (2001) claim that the advantage of reallocating workers are maximized with a cross-training on at least two machines. Felan and Fry (2001) also suggest that it is more cost efficient to invest on cross training activities not all workers, but few of them. It is indeed important to keep the right mix specialized workers operating only on their default machine and flexible workers that can solve machine oversaturation issues. In such a way, it is preserved and guaranteed both a high level of efficiency through specialized workers and a flexible shop floor thanks to cross-trained workers.

6.2 Limits in the workers reallocation

Although the reallocation of workers in a shop floor under the implementation of workload control has proven to drive to consistent benefits in performances, it seems that the assumptions made by most researchers are not applicable in real industrial contexts (Yue, Slomp, Molleman & Van Der Zee, 2008).

Fruggiero, Fera, Iannone and Lambiase (2015) claim that most of the research on reallocation of workers made too simplistic assumption such as the equality of efficiency of workers in the transferred machines and the lack of consideration of transfer time. Considering workers having the same efficiency to external machines may lead to misleading results that are not scalable in real shop floor. As a matter of fact, a worker is specialized in one machine and even if has knowledge on further machines, it is unlikely that he operates exactly with the same speed in those machines. The same holds true for the strong assumption that there is any transfer time when workers move from machines. If it is made such assumption, the algorithm pushes towards an excessive number of reallocation of workers that may be not realistic.

These two assumptions have driven researchers to consider that workers could be reallocated to all machines in a shop floor (Yue, Slomp, Molleman, & Van Der Zee, 2008). However, if it properly considered the decrement in efficiency in external machines and mostly the impact of transfer time,

reallocating workers in all machines may be far from real scenario. Apart from this fact, several academics showed that the majority of improvements were reached with a limited level of workers' flexibility, i.e. letting transfer only few operators on limited machines. Felan and Fry (2001) studied the performances of a production system with workload control under different levels of workers' flexibility and concluded that there was not a strong difference between incremental and full flexibility. In a nutshell, the results with a flexibility of 2 (e.g. an operator can be transferred only in two machines) were almost similar with a full flexibility where workers could be reallocated among all machines.

Such conclusions have been reached also by Park (1991) and Malhotra, Fry, Kher and Donohue (1993), Fry et al. (1995) and Campbell (1999), where a minimum introduction of workers' flexibility was enough to gain the maximum benefits in terms of shop floor performance. Further increase in flexibility just led to marginal performance improvements that did not justify the higher cost needed for extensive cross-training.

6.3 Reallocation of workers within parallel machines

Given the application of workload control in real production system struggles, due to the simplistic and sometimes far from reality simulation model developed by most academics (Bertolini, Romagnoli, & Zamorri, 2015; Yuan, 2017; Thürer, Stevenson, Silva, Land, & Fredendall, 2012), one of the objective of this thesis is to assess a simulation model as much as possible close to real shop floor configurations. To this aim, it is analyzed the reallocation of only few workers to only limited machines. As aforementioned, allowing every worker to be transferred to every machine not only seems quite unrealistic, but also leads to poor marginal results over a limited flexibility and it is likely not to be cost-effective.

In order to make more realistic the reallocation of workers, it has been simulated a shop configuration where it is likely that workers are going to be transferred in real circumstances. It concerns the reallocation of only workers operating in parallel machines and within the latter. The reallocation of workers among parallel machines has been poorly analyzed by the workload and DRC literature. Bokhorst and Gaalman (2009) have studied a similar scenario and have stressed that cross-training should be pursued within homogeneous subgroups of machines such as parallel machines.

Rationales making interesting the study of workers reallocation within parallel machines are:

- *Limitations of reallocation***:** transferring workers only within parallel machines of a station supports the new literature studies, highlighting that most of benefits from workers flexibility can simply be gained through a limited reallocation of workers.

- *Cost effectiveness*: allowing the transfer only for workers operating in parallel machines requires less investments, in terms of both cost and time, in cross-training activities. The training on workers is indeed going to be performed only for ones working with the parallel machines.

- *Low impact of transfer time:* reallocating workers in a shop floor may be negatively impacted by the time lost for moving from a machine to another. However, this issue is minimized if it allowed the transfer of workers only among parallel machines, as the latter are usually located in the same department or in general close to each other.

- *Faster and cheaper cross training:* machines that are in parallel have to simultaneously perform the same task. Thus, they are identical or very similar. The similarity of parallel machines helps operators in learning how to work in both of them in a shorter time than other machines. This advantage turns into less training sessions required per operator and hence less costs.

- *Lower efficiency loss:* the just mentioned similarity of machines also entails that a worker may operate with a similar efficiency to the parallel machine of its default machine. As a matter of fact, machines in parallel are pretty similar and the experience and knowledge an operator nurtured on his standard machine can be applied even on its parallel machine.

- *Necessary reallocation:* parallel machines perform the same task, but they probably are not perfectly equal. More specifically. Such machines usually have different “ages”, i.e. one of the two is much older/newer. Having one new and one old machine that are in parallel is quite frequent in job shops; their different “ages”, however, means that they have also different processing times – the newer machine has a lower processing time than the older one. Such difference in turn leads to imbalances, as the older machine turns into the bottleneck and becomes oversaturated, while the newer one is typically undersaturated. This imbalance negatively impacts the shop floor time. The issue can be solved by reallocating

workers among parallel machines. In fact, the worker on the newer machine, frequently undersaturated, can be transferred to the older machine when it is oversaturated, so rebalancing the system and avoiding the creation of bottlenecks.

6.4 Workers reallocation vs routing flexibility

Workers reallocation and routing flexibility have the same objective of balancing workload to the shop floor. However, the DRC literature mainly focused on studying the impact of only the workers reallocation. It lacks a direct comparison between the workers reallocation and routing flexibility as capacity balancing tools. In this concern, Park and Bobrosky (1989) studied workers' flexibility on a workload control system and suggest that future research should address a comparison between workers reallocation and routing flexibility.

In the current literature, there are few studies that directly address the comparison between reallocating workers and changing the routing of jobs, one of those that worth mentioning is the study carried out by Bokhorst, Slomp and Gaalman (2006). The aim of the study of Bokhorst, Slomp and Gaalman (2006) was to analyze whether it was more efficient to adopt a workers reallocation or routing flexibility as capacity balancing methods in a simulated "cell" layout. The shop floor was made up by two independent cells, and each cell had 5 different machines and 3 workers operating on it. Under static condition, the three workers (A, B, C) on the first cell can work in all the machines in that cell, the same holds true for the D, E, F workers on the second cell as shown in the image below.

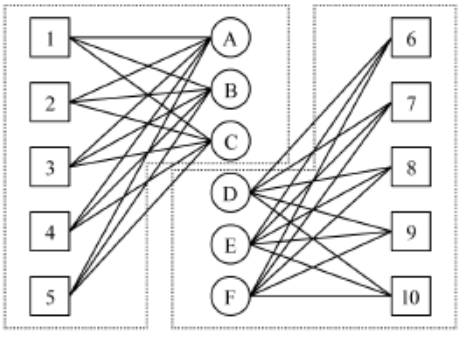


Figure 10 – Two cells configuration (adapted from Bokhorst et al., 2006)

Workers operating in one cell can be reallocated in the other cell when imbalances occur. The number of workers that can be temporarily transferred depends on a degree of reallocation that

Bokhorst, Slomp and Gaalman (2006) defines in 4 levels: 0%, 33%, 67% and 100%. At a 0% level, no workers are transferred to the other cell, while at the 33% only one (b), at 67% two workers (c) and at 100% level all workers (d) as illustrated in the image below.

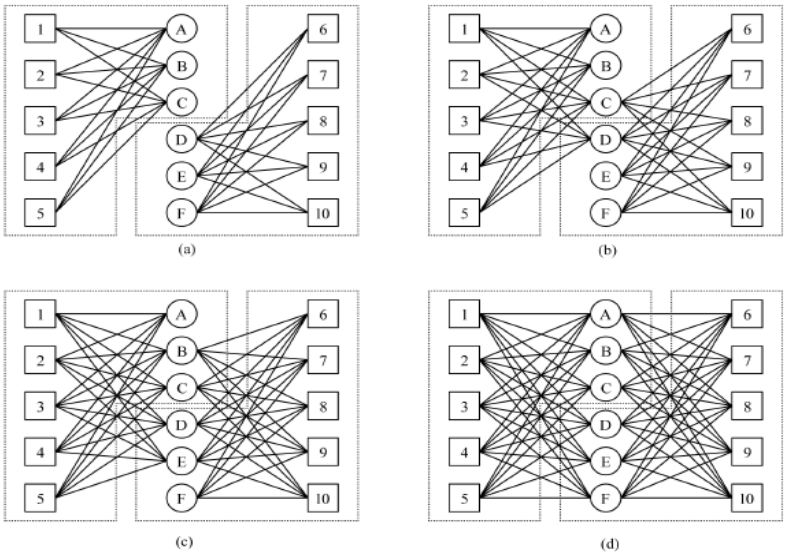


Figure 11 – Different levels of workers reallocation (adapted from Bokhorst et al., 2006)

In what concerns the routing flexibility, each machine on a cell has a similar machine on the other cell that has overlapping capabilities and then it could be exploited the change of routing of jobs. The similar machines are 1-6, 2-7, 3-8, 4-9, 5-10. For instance, if a job has to be processed on the machine 1 (first cell), it can be reallocated to the machine 6 (second cell) through the routing flexibility.

Bokhorst, Slomp and Gaalman (2006), as applied in the workers reallocation, define four levels of interchangeability (i.e. the percentage of jobs whose routing can be reconsidered): 0%, 33%, 67% and 100%. At 0% any jobs can be reallocated, at the 33% only 33% of jobs can be changed their routing and so on.

Bokhorst, Slomp and Gaalman (2006) found that both the two capacity balancing methods strongly reduce the mean flow time, but the maximal routing flexibility (at the 100% level) outperforms the maximal workers reallocation (at the 100% level). Bokhorst, Slomp and Gaalman (2006) give the possible explanation that the former method is able to adjust both overloaded workers and machines, while the latter can only lighten the workload of workers. Furthermore, it has been observed that most of the benefits with the two capacity balancing methods was gained with a medium-low level (33%), this is even more relevant for the routing flexibility. Increasing from 33% to the 100% level

leads to only about 20% of the benefits, as around 80% of the latter have been achieved with the 33% level.

These interesting results, however, are only valuable for the specific layout simulated in the study. As suggested by Bokhorst, Slomp and Gaalman (2006), it is needed a further research on the comparison between workers reallocation and routing flexibility in a more realistic flow shop layout. The aim of this thesis is to cover such a gap in the literature. Furthermore, their analysis did not count the implementation of a workload control system. The objective of this thesis is to actually study both singularly and in comparison, workers reallocation and routing flexibility under a workload control algorithm.

7. Discussion of literature

Here it is presented a summary of the main points highlighted in the literature review. Some considerations will be drafted regarding what is still missing in literature and what possible links between the topics presented could be better investigated. These considerations will be the ground on which the three research questions of this thesis have been built, which will be exposed in the final part of this section.

7.1 Main points in literature

- Workload control aims at obtaining a smooth flow in the shop floor by levelling demand and production over time. To achieve this, two different types of control can be used: *input control* (by managing the time of release of new orders) and *output control* (by adjusting production capacity thanks to machines, manpower and subcontracting)

- Input control mechanisms have been widely studied in the literature, setting up a very high and various number of configurations of Order Review and Release systems. *Workload limiting* (WL) algorithms, which are a method of control at release level, are among the most studied. The release of an order is allowed only if its contribution to the workload of the system does not exceed some pre-set norms. This method, due to its relevance in the workload control literature and its versatility, will be also the one used in the thesis.

- Regarding the *shop configuration*, authors in literature have traditionally focused on job shops. A new recent stream of research addressed this cause, stating how MTO and just in time companies, thanks also to streamlining processes performed according to the Lean philosophy, generally have a production configuration more similar to a flow shop. Hence, studies in the workload control literature should concentrate on this kind of configuration.

- Generally, over the years workload control literature has made significant progress in framing and defining workload control mechanisms. However, the *application by practitioners in real contexts* has been so far limited. The reasons for this have been referred to the simplifications that are generally made in the simulated models, which often place

them far from the real context. Unlimited capacity of queues, the lack of consideration of sequence-dependent setup times, and finally the rule "one station = one machine" are among the most used simplifications adopted by researchers. However, few studies in literature have tried to address the practical applicability of workload control mechanisms.

- Indeed, the study of systems that contemplates also *parallel machines* has often been neglected. This topic bears some potentialities mainly for two reasons: first it narrows the gap of the models from the real contexts, and second it opens up to new possibilities of output control such as routing flexibility and workers reallocations. Moreover, only few studies in literature have considered shops with parallel machines, and the ones that have done it have generally used the assumption of machines with the same processing times. Instead, different authors have claimed that considering different processing times and parallel machines are both characteristics that would make the models closer to reality (Henrich, Land, & Gaalman, 2007; Miragliotta & Perona, 2014). Indeed, the consideration of different processing times in a parallel machines configuration, to the best of our knowledge, has not been investigated in literature so far.

- *Routing flexibility* is a characteristic of a shop in which the production flow can be moved between similar machines. The type of routing flexibility investigated in this thesis is the one regarding the production flow feeding two different parallel machines. This is a method of output control which can alleviate the imbalances between stations that are generated inside a shop floor. The most important parameter for the routing flexibility is the level of machines interchangeability, which considers the percentage of jobs that can potentially be moved between one machine and another. The method can be applied in machine-intensive workplace, in which it has been implemented a control and transport system to move the production flow.

- *Workers reallocation* is a method of output control that consists in moving workers between one machine and another. The goal is the same of the routing flexibility, which is solving the system imbalances and improving the performance of the system. This method has been widely studied in the literature, and it is applicable mainly in context human-intensive, in which the transfer of a worker in another workstation can effectively decrease the total amount of time needed to perform the activities.

7.2 Why studying a system with parallel machines

As mentioned in the previous section, one of the reasons for the poor application of workload control systems by practitioners are the oversimplified simulation models carried out by academics that were far from real industrial implementations (Bertolini, Romagnoli, & Zammori 2015; Yue, 2017; Thürer, Silva, Stevenson, & Land, 2012; Miragliotta & Perona, 2000). Some assumptions made by researchers make the results of a study of a workload control algorithm not repeatable in a real production system.

One of those assumptions is interchanging a station with a machine, that is regarding stations in a job shop made up by only a single machine (see for example Land, Stevenson, Thürer, &, Gaalman, 2015; Thürer, Stevenson, Land, & Fredendall, 2018; Park & Bobrowski, 1989). This simplification does not mirror real shop configurations, since a station in a job shop is composed by more similar machines performing the same task and hence typically working in parallel. Considering the presence of a single machine per station prevented academics from studying the impact of workload control on parallel machines of a single station (Miragliotta & Perona, 2000). Therefore, in order to simulate a production layout as close as possible to a real one, the thesis proposes to analyze the more pragmatic shop floor configurations with stations made up by parallel machines. Moreover, it is interesting studying the impact of workload control on parallel machines, since the latter face unbalances issues and thus are likely to be the bottleneck that could constrain order release from the pre-shop pool (Bokhorst & Gaalman, 2009; Sirikrai & Yenradee, 2006).

Given the frequent workload imbalances present in parallel machines, when studying the impact of workload control on those machines is necessary driving the attention to output control/capacity balancing methods, which indeed may solve such unbalance issue. The two capacity balancing methods analyzed in the thesis are: the routing flexibility and the workers reallocation.

7.3 Why studying routing flexibility

Routing flexibility is an advantage of a job shop layout that consists on changing the initial routing of a job to a new one leading to better shop floor performances. This technique has been poorly discussed in the workload control literature, but should be properly analysed since it is exploited in real job shops (Miragliotta & Perona, 2000).

The traditional implementation of routing flexibility is within parallel machines. In fact, if a job is supposed to be processed by a machine which is oversaturated, it can be used the routing flexibility of such job, in order to change its routing towards a machine parallel to the first one that instead is undersaturated. Thus, it is interesting studying the routing flexibility, because, by rebalancing the machine saturation and limiting the creation of excessive bottlenecks, it may be a valid capacity balancing method for the frequent imbalances that suffer parallel machines.

Furthermore, the workload control studies directly addressing the routing flexibility technique mainly focused on understanding which parameter (level of interchangeability, grouping and routing decision) was the better under which conditions (Thürer, Stevenson, & Silva, 2011; Henrich, Land, & Gaalman, 2007; Zhao, Gao, Chen & Xu, 2015; Stevenson, 2006, Fernandes & Carmo-Silva, 2011). It lacks a study that thoroughly addresses how routing flexibility impacts various performances of a production system under workload control. The thesis aims to cover such gap.

7.4 Why studying workers reallocation

As opposed to the routing flexibility, the workers reallocation has been already analyzed in the workload control literature as output control technique. Nevertheless, most of researchers in their simulation study assumed that all workers could be transferred in all machines while being reallocated (Yue, Slomp, Molleman, & Van Der Zee, 2008). If it properly considered the decrement in efficiency in external machines and mostly the impact of transfer time, reallocating workers in all machines may be far from real scenario.

Besides this fact, several studies showed that the majority of improvements were reached with a limited level of workers' flexibility, i.e. letting transfer only few operators on limited machines (Felan and Fry, 2001; Park & Malhotra, 1991; Fry, Kher, & Donohue, 1993; Fry et al., 1995; Campbell, 1999). In the light of these results and in order to simulate a case as much pragmatic as possible, the thesis limits the reallocation of workers to the ones working on parallel machines that in turn can be moved within such machines.

7.5 Comparing the two methods

Limited academics directly compared routing flexibility and workers reallocation as capacity balancing technique. One of the few studies has been performed by Bokhorst, Slomp and Gaalman (2006), who however did not consider the implementation of any workload control literature, but just analyzed the two methods in a cell layout without any production system control. The objective of the thesis is also to cover this gap in the literature, by directly comparing routing flexibility and workers reallocation in a shop floor under a workload control system.

7.6 Research questions

The main purpose of this thesis is to study the effect that routing flexibility and workers reallocation have on a flow shop system characterized by parallel machines. The research questions answered in this thesis are the following:

- What is the contribution of the routing flexibility to performances and how these are affected by the level of interchangeability? What is the minimum level of interchangeability that leads to most of the result?

- What is the contribution of workers reallocation to performances and how these are affected by the efficiency of the workers, the permanence time and the transfer time between machines? Which when rule (decentralized, centralized) leads to the most benefits in a parallel flow-shop configuration?

- What is the respective contribution of the two methods (routing flexibility and workers reallocation) to performances, to the variation of system parameters such as stations' imbalance?

The first two research questions aim at investigating the benefits obtained on the system with the use of the two different output control methods. The effect on performances of the different parameters will be tested and assessed with an ANOVA analysis. The goal is to study the effect of the parameters and to assess the best combination that optimizes performances in the two cases.

The last research question instead makes a comparison of the two methods, assessing how they respectively respond to different variations in the system. The final goal is to obtain guidelines that indicate how the two output control systems are able to respond to changes in the system and to what extent the improvement in performances is achieved.

PART 2: Methodology

For the study of this thesis and to answer the research questions presented in chapter 8.5 a simulator has been used. It is implemented in Python (version 3.7) with the access to external libraries SimPy for the definition of the simulation environment and NumPy for the statistics functions.

The reference model for this thesis is Portioli and Tantardini (2012), as most of the parameters adopted for the simulation are equal to the ones presented in that research. This has been done being this thesis is part of the same stream of research and to allow the comparison of results.

This chapter entails a description of the shop configuration, the system analyzed, the parameters adopted, and the algorithms implemented. Then the design of the experiment and its implementation are presented, to give the possibility to readers to replicate and test the results.

8. Description of the model

8.1 Shop configuration

The shop configuration tested in the simulated model is a flow shop. It has been chosen such job configuration because it is more likely to represent a real production layout, where it is frequent to find a predominant flow direction (Oosterman, Land, & Gaalman, 2000; Portioli & Tantardini, 2012).

The flow shop is represented in Figure 12 and is made up of 5 aligned stations marked with their respective sequential number. Station 1, 2, 4 and 5 have only a single machine. While station 3 is the only station that has 2 machines located in parallel. Six pools, one before each machine, are used to store work-in-progress during the production process. Two additional pools are placed: at the beginning of the process, in order to collect customers' orders (i.e. a pre-shop-pool, not shown in Figure 12), and one at the end of the production process to store the finished goods.

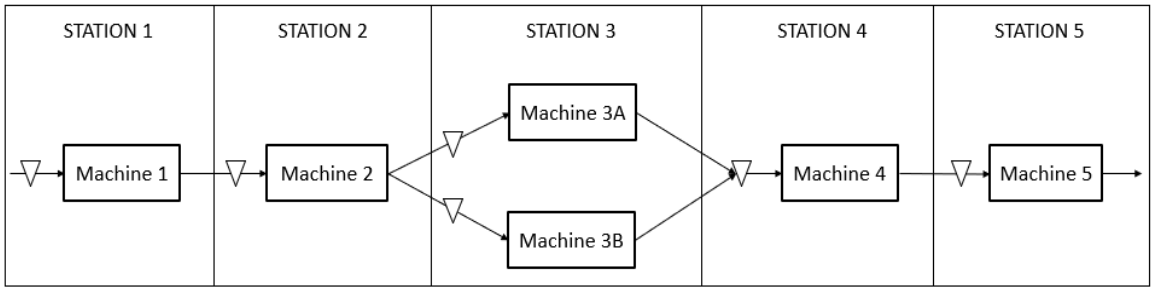


Figure 12 – Flow shop configuration with two parallel machines

Being the system a flow-shop, there are in this case only two possible routes that the products need to follow.

	Station 1	Station 2	Station 3	Station 4	Station 5
Route 1	*Machine 1*	*Machine 2*	*Machine 3A*	*Machine 4*	*Machine 5*
Route 2	*Machine 1*	*Machine 2*	*Machine 3B*	*Machine 4*	*Machine 5*

Table 2 – Two possible routings in the model

The routing flexibility is going to be applied in the third station because it is the only one having parallel machines. So, to be able to make a comparison between the two methods, we have considered the workers reallocation only between the two machines in parallel.

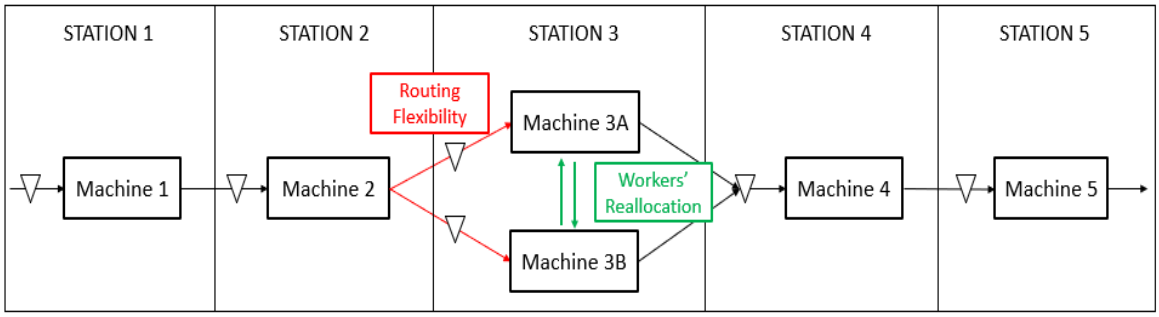


Figure 13 – The two models' application on the shop configuration

Although most of literature on workers reallocation, shown in the table below, considers more stations having parallel machines, the literature of routing flexibility on parallel machines examined only one station with parallel machines (see for example Bokhorst & Gaalman, 2009; Fernandes & Carmo-Silva, 2011; Zhao, Gao, Chen & Xu, 2015).

	Number of stations	Number of stations with parallel machines
Bobrowski and Park (1993)	9	9
Yue , Slomp , Molleman and Van Der Zee (2008)	4	4
Felan and Timothy (2001)	4	4
Malhotra, Fry, Kher and Donohue (1993)	6	6
Bokhorst, Slomp and Gaalman (2004)	3	3
Ma, Chu and Zuo (2010)	5	2
Sirikrai and Yenradee (2006)	6	2
Wang, Wang, Liu and Xu (2013)	5	4

Table 3 – Examples of configurations in literature

It has been decided to apply routing flexibility and workers reallocation in the only station with parallel machines since this allows to perform a more accurate analysis on that station. If instead all stations had parallel machines and consequently routing flexibility and workers reallocation could have been applied on each station, it would have been more difficult to assess their impact on the station. In fact, the imbalances brought by parallel machines of a station are transmitted to the next stations, hence causing a proliferation of effects that make the assessment on performances of shop floor much complex.

In order to deeply focus on routing flexibility and workers reallocation, it was necessary not to include external sources of variations that could make the whole analysis poorly robust. More specifically, the following assumptions have been made:

- All pools in the flow shop have endless capacity, thus they can theoretically store infinite jobs in the shop floor
- Handling systems can carry an unlimited number of jobs.
- All jobs have the same priority, i.e. any job must be produced before or after other jobs.
- Raw materials are always available in the shop floor: the production of a job cannot be delayed due to stock-out of raw materials.
- Machines breakages are negligible.
- Setup times are assumed as non sequence-dependent and they are included in the processing times of the jobs. This is in accordance with most of the studies in the workload control literature (Thürer, Stevenson, Silva, Land, & Fredendall, 2012).

8.2 Types of workforce

To each machine of the flow shop is assigned one worker, which leads to an overall of 6 workers on the shop floor. The application of workers reallocation allows workers not to remain in their default machine, rather they can be temporarily transferred to other machines. In this regard, it has been considered two workforce scenarios:

- **Static**: this scenario does not allow to exploit the transfer of workers, since it forces them to work only on their default machine.

- **Reactive**: workers can be transferred from their default machine to other machines when workers idleness or/and machine oversaturation occur. As better explained in Chapter 6.2, the reactive algorithm takes into account two main parameters to decide whether workers should be transferred and where: the machine load and the efficiency of workers on external machines. Whenever a worker is idle or there is a workload imbalance between machines (e.g. decentralized and centralized rule), it is triggered the reallocation of an idle worker to an external oversaturated machine.

As opposed to most of DRC literature where reactive workers are allowed to be transferred to any machine in the shop floor, *the thesis constrains the reallocation of reactive workers to only the two parallel machines of the station 3*. The two workers on the station 3 are the only reactive workers, while the workers on the stations 1, 2, 4 and 5 are constrained to their station and hence static.

	Station 1	Station 2	Station 3		Station 4	Station 5
Worker name	*Worker 1*	*Worker 2*	*Worker 3A*	*Worker 3B*	*Worker 4*	*Worker 5*
Type of worker	*Static*	*Static*	*Reactive*	*Reactive*	*Static*	*Static*

Table 4 – Types of workforce considered in the model

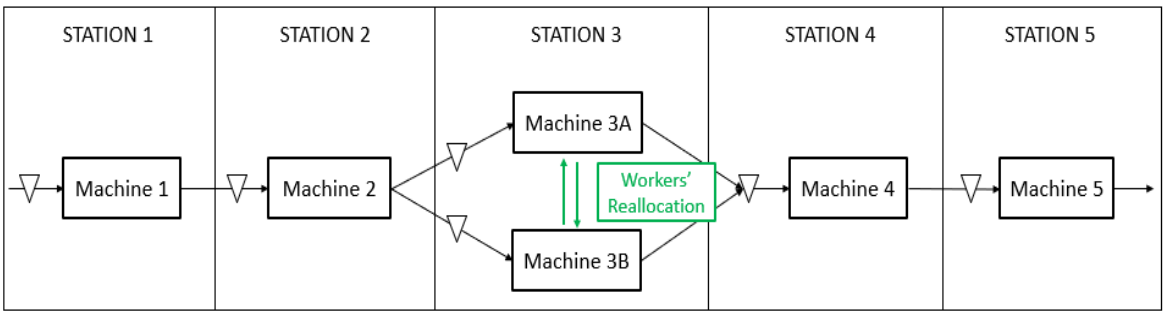


Figure 14 – Application of workers reallocation

Therefore, if the machine 3A of the station 3 is oversaturated and requires the extra capacity from an external worker, only the worker of the machine 3B can be transferred there. As discussed in Chapter 7.2, the frequent imbalances occurring in parallel machines imply that whenever 3A is oversaturated, 3B may be undersaturated and hence likely to provide extra capacity to 3A.

Concerning their level of efficiency, three values have been tested: 100%, 75%, 50%. The assumption is that every worker is 100% efficient in his default station, while the two reactive workers, when reallocated, can contribute to the station with the same or a decreased value of efficiency. The following tables show the distribution of efficiency among the workers for the three scenarios.

	Machine 1	Machine 2	Machine 3A	Machine 3B	Machine 4	Machine 5
Worker 1	100%	0	0	0	0	0
Worker 2	0	100%	0	0	0	0
Worker 3A	0	0	100%	100%	0	0
Worker 3B	0	0	100%	100%	0	0
Worker 4	0	0	0	0	100%	0
Worker 5	0	0	0	0	0	100%

Table 5 – Workers flexibility with full efficiency

	Machine 1	Machine 2	Machine 3A	Machine 3B	Machine 4	Machine 5
Worker 1	100%	0	0	0	0	0
Worker 2	0	100%	0	0	0	0
Worker 3A	0	0	100%	75%	0	0
Worker 3B	0	0	75%	100%	0	0
Worker 4	0	0	0	0	100%	0
Worker 5	0	0	0	0	0	100%

Table 6 – Workers flexibility with moderate efficiency

	Machine 1	Machine 2	Machine 3A	Machine 3B	Machine 4	Machine 5
Worker 1	100%	0	0	0	0	0
Worker 2	0	100%	0	0	0	0
Worker 3A	0	0	100%	50%	0	0
Worker 3B	0	0	50%	100%	0	0
Worker 4	0	0	0	0	100%	0
Worker 5	0	0	0	0	0	100%

Table 7 – Workers flexibility with medium efficiency

8.3 Order release algorithm

To widen the implications and results of the thesis, it has been applied an order release algorithm that is most common on the workload control literature: the *workload limiting algorithm*.

As explained in Chapter 3.2, the workload limiting algorithm is a periodic algorithm that evaluates a fixed interval of time the release of jobs from the pre-shop pool. The limiting algorithm implemented in the thesis, in particular, has a periodic interval of time of 480 minutes. Moreover, out of the three types of limiting algorithm respectively setting upper bound norm, lower bound norm and both upper and lower bound norm, the thesis refers only to the one setting an upper bound norm. Thus, a job is released on the shop floor only if its load contribution to the current workload of every station does not exceed each of their norms.

The implementation of the routing flexibility and workers reallocation algorithms allows to create a sub-classification of the broad limiting algorithm in its three more specific algorithms, which are shown below:

- Workload limiting: it is the basic limiting algorithm without both routing flexibility and workers reallocation algorithms.

- Workload limiting with routing flexibility: the routing flexibility is applied under the limiting algorithm. Routing flexibility is activated through a sub-algorithm of the main limiting algorithm. All the workers in this configuration are static and cannot be reallocated.

- Workload limiting with workers reallocation: the workers reallocation is implemented in the limiting algorithm. The reactive type of workforce is a sub-algorithm of the main limiting algorithm. No routing flexibility is allowed.

As mentioned in the previous chapter, the purpose of the thesis is to study the implementation of two different kinds of output control systems, the routing flexibility and the workers reallocation, in a flow shop characterized with parallel machines.

After having explained the model under investigation, which is a flow shop composed by five stations with two machines in parallel, it is now possible to better explain the two proposed models. The first one refers to the application of *routing flexibility* algorithm to balance the flow between the two parallel machines, while the second one is the application of different *workers reallocation* rules in order to solve the imbalances in the same station under investigation. In the remaining part of this chapter the two models will be presented.

9. First proposed model: Routing Flexibility

Routing flexibility consists on the flexibility of jobs in changing their initial routing (Shewchuk, 1998). If the release of a job in the shop floor leads to machines unbalances or excessive queues, it can be changed its routing to an alternative one, which instead brings better shop floor performances. In the model studied there are two possible alternative routings, one that passes from Machine 3A, and one from Machine 3B.

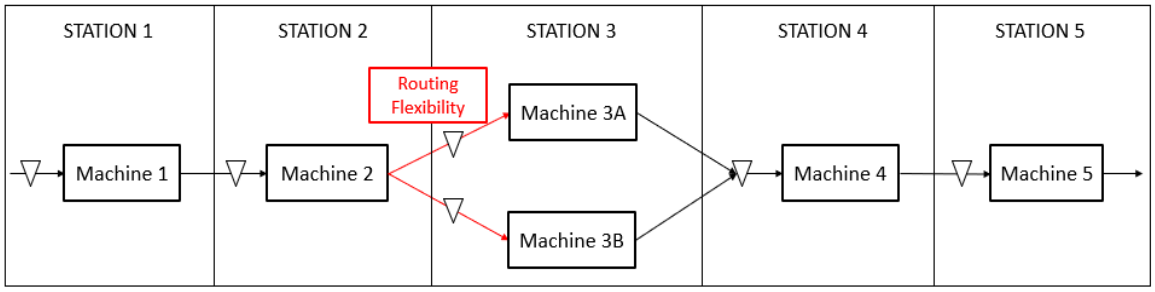


Figure 15 – Application of routing flexibility

The implementation of the routing flexibility concept on a workload control system requires to choose parameters of three decisional areas: the *level of interchangeability*, the *routing decision* and the *grouping machines decision*.

9.1 Level of interchangeability

Interchangeability of machines indicates the ability of machines to perform similar operations (Henrich, Land, & Gaalman, 2007). As it has been explained in Chapter 5.2, a level of interchangeability of parallel machines reflects how many jobs can be indistinctly worked in either in the first or in the second parallel machine. Therefore, a percentage of interchangeability can be seen as the percentage of jobs that can exploit the routing flexibility.

As a result, the interchangeability in the simulated model has been designed as an attribute of jobs generated in the system. This attribute has been assigned not to all jobs, but only for a percentage of jobs that represents the level of interchangeability. For instance, if it has been chosen a level of interchangeability of 30%, the 30% of jobs are going to be assigned the attribute “interchangeable”. The routing flexibility may be applied to only such 30% of jobs, which means that only those jobs

with this attribute can be changed from one machine with another, while the others will need to follow their initial routing.

The results of routing flexibility can vary according to the level of interchangeability analyzed. As a result, most studies such as the ones of Henrich, Land and Gaalman (2006), Henrich, Land and Gaalman (2007) and Fernandes and Carmo-Silva (2011) have considered a wide range of levels of interchangeability.

The thesis has analyzed the levels of interchangeability presented by Henrich, Land and Gaalman (2007):

	Interchangeability levels considered (between Machine 3A and 3B)					
Station 3	*0%*	*5%*	*10%*	*20%*	*50%*	*100%*

Table 8 – Levels of interchangeability studied

Table 9 explains how the interchangeability for the two different flows works. The table is made considering a total of 100 jobs that need to pass through the system. The value of 100 is used for the sake of simplicity.

Intechangeability (%)	Number of jobs that need to pass from Machine 3A	Number of jobs that need to pass from Machine 3B	Number of jobs that can either pass from 3A or 3B (interchangeable)
0	50	50	0
5	47.5	47.5	5
10	45	45	10
20	40	40	20
50	25	25	50
100	0	0	100

Table 9 – An example of jobs' interchangeability

In the thesis these levels of interchangeability will be tested, in order to see the marginal benefit obtained from the increase of this parameter.

9.2 Routing and grouping decision

As discussed in Chapter 5.1, the routing decision concerns whether to change the route at the pre-shop pool level or at dispatching (i.e. directly on the shop floor), while the grouping decision regards if the parallel machines are going to have each their workload norm or if they share only one norm.

The thesis applies the routing decision at dispatching and the non-grouping decision (e.g. parallels machines with their own workload norms) consequently to the results found by Henrich, Land and Gaalman (2007). The latter analysis indeed compared all possible scenarios of routing and grouping decision and concluded that the routing decision at dispatching and non-grouping decision of interchangeable machines outperform all other scenarios, regardless of the level of interchangeability of the system.

Therefore, in the thesis the change of routing of jobs is going to be performed directly at the shop floor (routing decision at dispatching) and not when jobs are still in the pre-shop pool (routing decision at order release). While regarding the grouping decision, the parallel machines have been considered as they were standard single machines, each with their own workload norms (non-grouping decision).

9.3 Routing flexibility algorithm

Academics addressing routing flexibility do not specifically explain and neither show how the change of the routing of jobs is performed in their simulated model and which are the conditions that drive the change of routing. This section instead thoroughly explains how the routing flexibility works and when it is activated.

Through the application of routing decision at dispatching level, the routing flexibility is activated when jobs are on the shop floor. As illustrated in Figure 16, in the model studied the routing flexibility consists in changing the path of a job that needs to pass from machine 3A to machine 3B or vice versa.

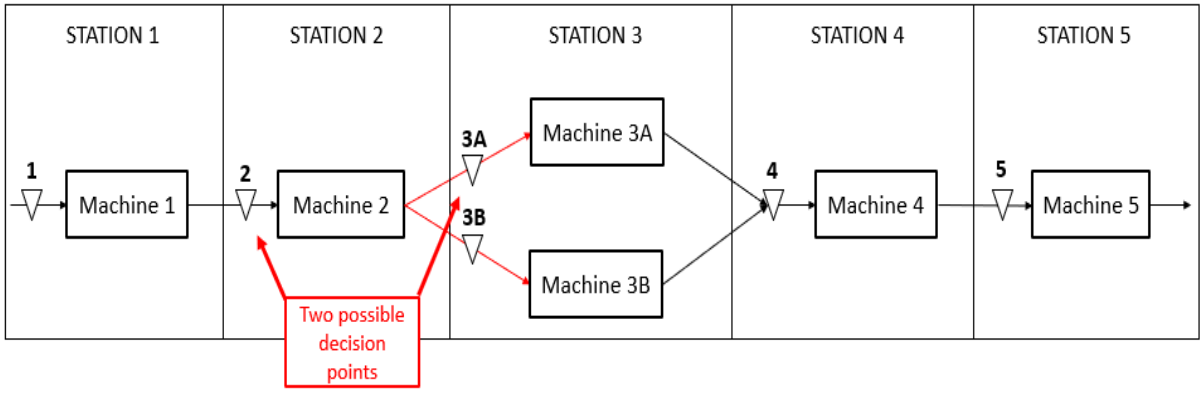


Figure 16 – Possible decision points for the application of routing flexibility

Anyway, the decision of changing the routing can be taken mainly in two different points of the shop floor (see figure above):

1. While the jobs are still queuing in pool 2, so before entering the station with the parallel machines (station 3).
2. While the jobs have already entered the queue in pool 3A or 3B, thus changing their queue.

The second decision point entails that when jobs are in the pool 3A, which is the pool for the jobs that are supposed to be worked only by the machine 3A, their routing can be changed to machine 3B and they have to be moved to the pool 3B, or vice versa. However, such an alternative of changing the routing at the closest point (pool 3A and 3B) of parallel machines has been set aside and not definitely applied for two main reasons.

First, it has been empirically compared to the alternative of changing the routing when jobs are in the pool 2 and the latter outperforms the former in terms of performance. Second, it is unlikely that jobs in a shop floor can be easily moved to pool 3B once they have already entered the queue at the pool 3A (and vice versa). This is because in a real context the two machines (3A and 3B) may be placed far from each other, making not practical to move the jobs from one queue to another, which will likely imply a waste of time and resource.

While, if the routing decision is taken at the station 2, which means that the routing of jobs is changed before they have entered one of the two queues, they will be able to follow directly the new routing, allowing an easier redirection of the production flow. Thus, for these reasons it seemed more applicable to decide whether jobs should be in pool 3A or pool 3B, to be worked respectively by machine 3A or 3B, in the pool preceding the station with parallel machines, i.e. the pool number 2.

Another possible decision point could have been in queue 1, so right after the release from the PSP. Anyway, the results found in Henrich, Land and Gaalman (2007) presented that taking routing decision at dispatching brought the best results (rather than at release). For this reason, taking the instead routing decision at pool number 2 looks to be the best trade-off.

Once explained where it is activated the routing flexibility, it will be now shown how the algorithm developed for this thesis works. The first part refers to the evaluation of the workload in the queues of pools 3A and 3B. The second part instead regards how the algorithm decides whether the routing of the interchangeable jobs should be changed between the two machines.

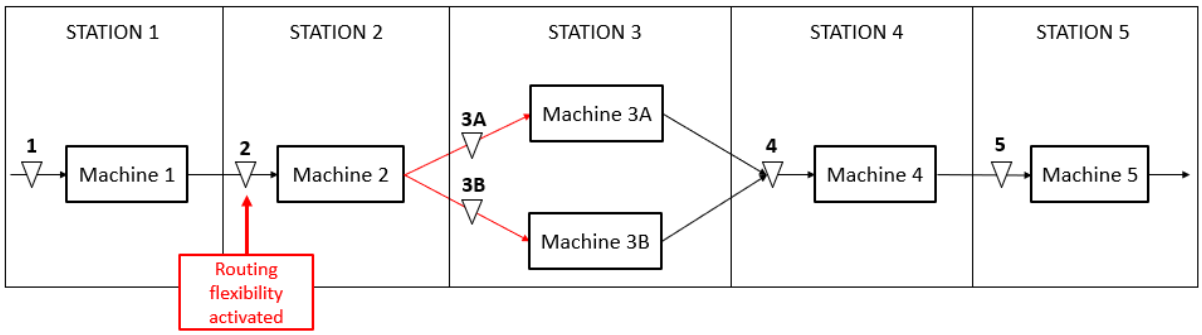


Figure 17 – Chosen decision point where routing flexibility is activated

The algorithm can be summarised in the following steps:

1. The queues in pools 3A and 3B are evaluated by summing the direct workload of the jobs currently present in those two queues. Two values are obtained, which are the direct workload on machine 3A and the direct workload on machine 3B.

$$W(j) = \sum i \; Job\,(i,j) \qquad \text{for every j and i}$$

where

- W(j) is the current workload on parallel machine j.
- Job (i, j) is the processing time of the job i at the parallel machine j.

2. Now the algorithm evaluates the jobs that are in pool 2 (the one before the two parallel machines). The goal is to calculate the forecasted workload of the parallel machines queues (pool 3A and 3B). To do that, for every job present in queue 2, the algorithm considers its standard routing and calculates the forecasted workload of the parallel machine that will be visited by that job, through the following formula:

$FW(j) = W(j) + PT\ (i,j)$ for every j and i

where

- FW(j) is the future workload on parallel machine j.
- W(j) is the current workload on parallel machine j calculated before.
- PT(i,j) is the processing time of the job i, currently in queue in pool 2, that according to its standard routing will visit the parallel machine j.

3. The algorithm checks out if the job under investigation (which is currently in pool 2) is interchangeable (this means that it could either go to one or the other parallel station by exploiting the routing flexibility).

4. If it is **interchangeable**, the algorithm compares the two forecasted workloads (FW3A that refers to pool 3A and FW3B that refers to pool 3B) to check if the job's routing should be changed in order to balance the two parallel machines.

   It is given a practical example that explains when the job changes its routing from its standard machine (i.e. the one that it should visit according to its routing) to the less saturated parallel machine. If a job has as standard machine the machine number 3, two different scenarios may happen:

Case	Meaning	What happens
1) FW3A > FW3B	The forecasted workload of pool 3A is greater than forecasted workload of pool 3B	The interchangeable job that should be worked in the machine 3A changes its routing towards the machine 3B.. This is done because the latter is less saturated, in order to better balance the workload of the two stations
2) FW3A < FW3B	The forecasted workload of pool 3A is greater than forecasted workload of pool 3B	In this case instead, the job does not change its routing towards machine 3B and keeps its standard machine 3A, because the latter is less saturated.

Table 10 – Two scenarios of forecasted workload

Opposite reasoning holds true for a job that has the machine 3B as standard machine. In fact, in case 1 such job keeps its standard routing towards machine 3B, while in case 2 it changes its routing towards machine 3A. While, in case the job under investigation is **not interchangeable**, the algorithm passes to the next job in queue in pool 2.

5. The algorithm checks all the jobs present in pool 2, repeating for each the steps 2-3-4. Every time a job's routing is changed from one machine to another, the values of the forecasted workloads are adjourned, by subtracting the processing time of the job to the previous workload of the first machine and adding it to the one where it has moved.

6. When every job in the pool 2 has been checked, the algorithm rests until another new job will come to pool 2.

Another important aspect that was considered in developing this algorithm was that one job's routing can be changed only one time. Indeed, the algorithm does not only check if the job is interchangeable (step 3), but also checks if the job has already been moved before. To keep track of this, the flag "Moved" was inserted. This flag value is set as False by default, but becomes True whenever the job is moved. This control has been implemented in order not to make the system too nervous avoiding many repeated changes of the same jobs. Instead, in this way it aims at reaching a better stability and balancing of the production flow.

10. Second proposed model: workers reallocation

The second model refers to the output control method of reallocating workers between machines. Given the rationales discussed in Chapter 6.3, the thesis studies the impact of limiting workers reallocation only within parallel machines. This has been done for different reasons: from the practical point of view, since parallel machines are similar, it is more likely that one worker, when transferred to the other machine, will maintain a high level of efficiency; another reason is that parallel machines are often placed close to each other, or at least in the same department, which justifies the consideration of low transfer times between the machines.

It is also analyzed how the results coming from the transfers of workers within parallel machines is affected by three main parameters of the model. The parameters studied are the **efficiency** of workers on the machine where they are being temporarily reallocated (as explained in Chapter 6.1), the impact of **transfer time** which is the time needed to move from one machine to another and finally the **permanence time** which is the minimum time that a transferred worker needs to spend in a machine before being allowed to come back. At the end of this chapter, the different values considered for these three parameters are reported.

The when rule adopted in this model is the **decentralized**. This means that one worker can be transferred to the other machine only when he is idle, and the queue before his machine is empty. The **centralized** rule, instead, allows the transfer of a worker also when his queue is not empty. The centralized will be tested only on few experiments for the comparison with the other rule. While, for most of the experiments of this thesis, the decentralized rule will be used.

10.1 Worker's reallocation algorithm

The algorithm can be summarized as following (*decentralized rule*):

1. Whenever a worker of one of the two parallel machines (3A and 3B) is idle, the algorithm is activated.

2. Being the worker idle, it means that the queue of his machine is empty. Thus the algorithm checks if the queue of the other parallel machine is greater than a pre-set threshold (for example 100 minutes of workload).

3. In case the queue of the other parallel machine is greater than the threshold, the worker is transferred to the other machine to employ his time in helping the other worker. As the transferred worker needs to move to the other machine, a *transfer time* has to pass before he can actually work on that machine.

4. The worker will remain on the other machine for a time equal to at least the *permanence time*, which is the minimum time that he needs to spend in the other machine. After the permanence time has passed, the worker will come back to its station if any of the following events have occurred:

- A new job has arrived to the default machine of the transferred worker;
- The current job in the machine where he was transferred has been finished.

If neither of these two events have occurred yet, the worker will remain in the machine until at least one of these two events will have happened.

5. Finally, the worker will come back to his default machine and the algorithm will cycle again, repeating the process from step 1.

This method, similarly to the routing flexibility, has the goal to solve the imbalances in the parallel station in order to solve bottlenecks between the two parallel stations and to obtain a smooth production flow.

The other goal of this thesis is to study the effects of the different parameters on the performance of the two models. Moreover, the two models will be compared considering how well do they respond to the change of some external parameters, in order to assess their robustness to different changing scenarios.

A summary of the main simulation parameters is presented in the following table:

Shop configuration studied	Flow shop with parallel machines
Number of stations	5
Number of machines	6
Number of workers	6
Dispatching rule	FCFS
Capacity of each stage	480 minutes / day
Working days length	480 days
Arrival rate distribution	Exponential
Processing time distribution	Truncated log-normal Mean 30, variance 900 Minimum 0 , maximum 360
Job contractual due date	Uniform [a,b] with a,b depending on the shop configuration
Levels of interchangeability	0% 5% 10% 20% 50% 100%
Workforce flexibility	1 station (for workers 1,2,4,5) 2 stations (for workers 3A,3B)
Workforce efficiency	0% 25% 50% 75% 100%
Transfer time	0 min 10 min 15 min 30 min
Permanence time	0 min 30 min

Table 11 – System parameters and theirs levels

11. System parameters

11.1 Jobs' arrival rate

The *arrival rate* of jobs follows an exponential distribution, as it has been shown to fairly approximate the arrival process of jobs in real production systems (Moreira & Alves, 2012). Adopting an exponential distribution entails that jobs are not generated at fixed intervals of time, but they are created continuously.

To define the *interarrival time*, which is the time between two jobs arrive to the system, the following formula has been adopted:

$$Interarrival\ time = \frac{Working\ hours\ * Target\ utilization}{Average\ processing\ time * Number\ of\ stations}$$

where

- Working hours: 480 minutes, hours in which the shop floor works
- Target utilization: 93.75%, it has been set this value taking as reference the study of Portioli and Tantardini (2012)
- Average processing time: 30 minutes
- Number of stations: 5

11.2 Jobs' due dates

In literature there is not a common shared methodology to define due dates. One of the most complete is the one suggested by Bertolini, Romagnoli and Zammori (2015) and Thürer, Stevenson and Land (2016). This methodology takes into account: a constant allowance, a factor generating variability and the number of stations in the system. This approach consists in obtaining the value of the job due date from a uniform distribution between α and β; where α is the due date value for which the percentage of tardy jobs is 20%, using an algorithm of immediate release of orders. Then,

β is obtained by multiplying the number of stations in the shop configuration for the ninety-fifth percentile of the processing time (83.6 minutes in the model), plus a constant allowance (2400 minutes).

$$Due\ date = Uniform\ [\alpha, \beta]$$

with:

$$\alpha = Duration\ of\ due\ date\ for\ which\ the\ percentage\ of\ tardy\ jobs\ is\ 20\%$$

$$\beta = Number\ of\ stations * 95th\ percentile\ of\ processing\ time + 2400$$

To determine α, different simulations have been performed letting the duration of due date vary in order to find out which value led to a percentage of tardy jobs of 20%. The algorithm used was the immediate release. In Figure 18 the results are shown:

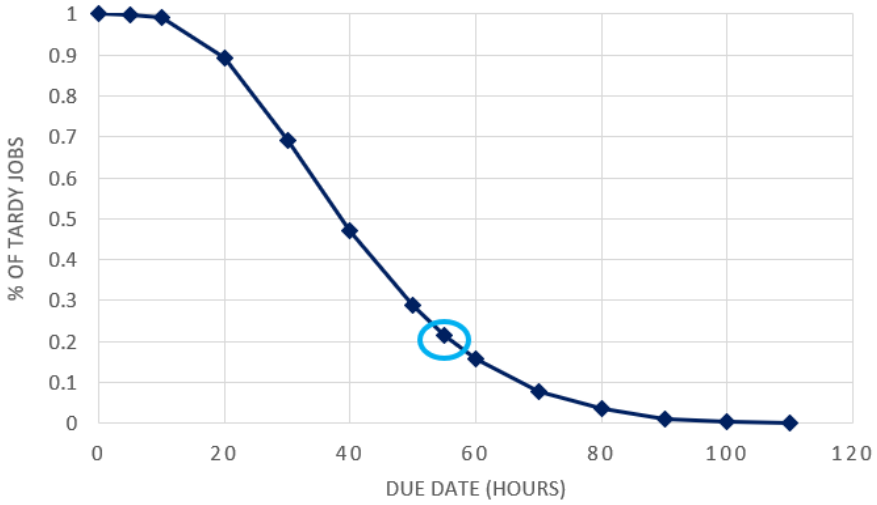


Figure 18 – Percentage of tardy jobs vs Level of due date

According to the graph, the value of 58 hours (3480 minutes) led to approximately 20% of tardy jobs. The values of α and β are presented in the next table:

Due date	MIN (β)	MAX (α)
Uniform [α, β]	2818	3480

Table 12 – Minimum and maximum due date values chosen

11.3 Jobs' processing time

The analysis performed by Portioli and Tantardini (2012) has been taken as a reference to define the distribution and values of processing time of jobs in the system.

The processing time of jobs follows a truncated log-normal distribution, with mean equal to 30 minutes, variance of 900 minutes, with observations ranging from 0 to 360 minutes. It has been crucial to consider the processing time as a truncated distribution, as otherwise it may happen that jobs may not be released but retained in the pre-shop pool throughout the full length of the simulation, as their too long processing time would immediately lead to exceed the workload norms. In such a way, it is prevented that these heavy jobs strongly affect the results of the limiting algorithm.

Furthermore, despite this reduction of system variance, the simulated model assures a high variability of processing time thanks to a wide range of values. In fact, letting observations range from 0 to 360 minutes with a mean processing time of 30 minutes, the maximum possible value of processing time will be 360 minutes, which is 12 times bigger of its mean value of 30 minutes. This high variability of processing time causes a high overall lead time variability in the system. This has been created on purpose, as the system is aimed at reflecting the variability faced by real job shop layouts (including flow shops) typical of MTOs companies (Zäpfel & Missbauer, 1993). This variability indeed causes imbalances in stations loads with orders spending long time in queues. This is the situation that the model implemented wants to solve.

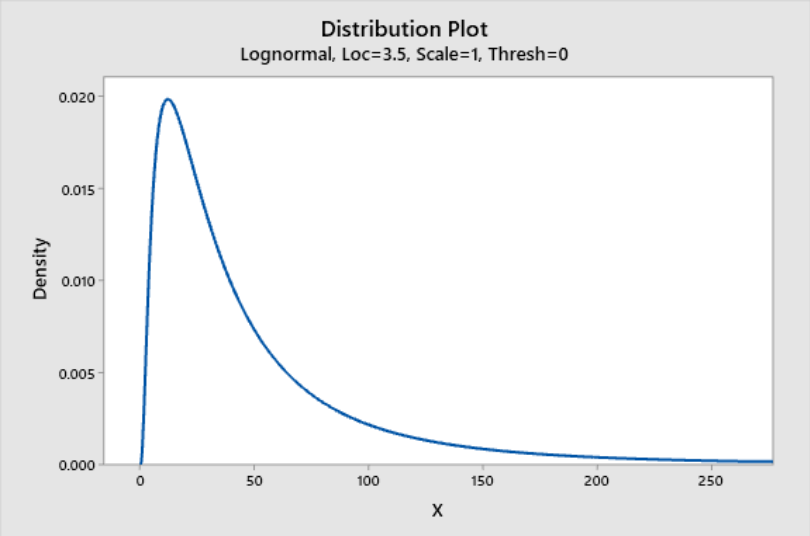


Figure 19 – Distribution of jobs' processing time

The distribution, mean and variance of processing time of jobs generated in the system are the same for all machines. However, the final value of processing time needed to be adjusted in parallel machines, as evidenced by Yue, Slomp, Molleman and Van Der Zee (2008), Bokhorst and Gaalman (2009) and Bokhorst, Slomp and Gaalman (2006).

As previously explained, there are two possible routings in the model of this thesis:

	Station 1	Station 2	Station 3	Station 4	Station 5
Route 1	*Machine 1*	*Machine 2*	*Machine 3A*	*Machine 4*	*Machine 5*
Route 2	*Machine 1*	*Machine 2*	*Machine 3B*	*Machine 4*	*Machine 5*

Table 13 – Two possible routings in the model

As it can be seen from the table above, necessarily the 100% of jobs needs to pass from machines 1, 2, 4, 5, while the flow splits between the two machines in parallel (3A and 3B). It has been decided, in order to avoid the creation of explicit bottlenecks in the system, to split the flow equally between the two parallel machines, as each will receive the 50% of the jobs.

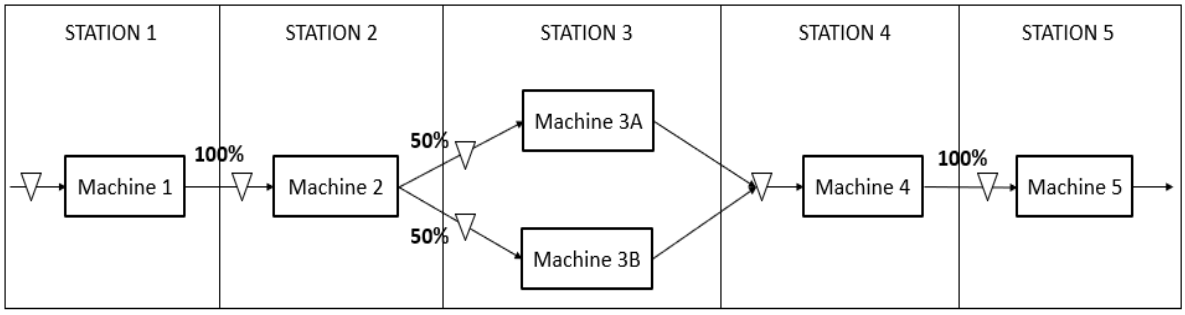


Figure 20 – Distribution of production flow

In order not to create an intrinsic instability in the system, the processing time of jobs in parallel machines (machines 3A and 3B) has been doubled with respect to the ones generated in the other machines in line (machines 1, 2, 4, 5), as suggested by Henrich, Land and Gaalman (2007). This has been necessary as the two parallel machines will have half of the jobs to process. In fact, if they had the same processing time of the other machines, having half of the jobs they would be merely utilized for around the 50% of their capacity, which is an excessive level of idleness. While, doubling their processing time of jobs in parallel machines, they are going to be utilized as much as

the machines in line. This circumstance where all machines in the flow shop have the same utilization is called *balanced scenario*.

The table below summarizes the difference of processing time for parallel and line machines in the simulated model.

	Processing time					
Machine	Machine 1	Machine 2	Machine 3A	Machine 3B	Machine 5	Machine 6
Final value	(value generated)	(value generated)	(value generated) x 2	(value generated) x 2	(value generated)	(value generated)

Table 14 – Computation of the processing time for the machines in the model (balanced scenario)

The one presented was the balanced scenario, in which all the machines have the same rate processing time/demand.

However, one of the main characteristics of parallel machines is that they may have different performances due to their different "ages", which will cause for example the newer machine between the two to go faster than the old one (see Chapter 5.2). This difference in processing times may lead to a system suffering from workload imbalances, having two machines working at different speed. Although most authors have claimed that in real contexts parallel machines generally have different processing times, which are the cause of frequent intrinsic imbalances in the system, in literature the machines studied in the simulated models have instead been generally considered with the same processing time (see for example Fernandes & Carmo-Silva, 2011; Zhao, Gao, Chen & Xu, 2015; Felan & Fry, 2010).

Since the thesis has the goal to simulate a shop configurations as much as close to real shop floors, it will be examined also the case when parallel machines have different processing times, and it will be compared to the standard case when the processing time is the same for every machine. As a consequence, it will be assessed whether and how performances of routing flexibility and workers reallocation depend on the *imbalance level* of parallel machines.

To recreate the difference in processing time of parallel machines it has been multiplied the value generated in machines 3A and 3B not both by 2 as in balanced scenario discussed above, but by 2,1 and 1.9. In this case the average is still the same, but it has been created a difference in speed

between the two machines by nearly the 10% (difference of 0.2 over the average of 2). This scenario is called *unbalanced scenario* and it is summarized in the table below.

	Processing time					
Machine	Machine 1	Machine 2	Machine 3A	Machine 3B	Machine 5	Machine 6
Final value	(value generated)	(value generated)	(value generated) x 2.1	(value generated) x 1.9	(value generated)	(value generated)

Table 15 – Computation of the processing time for the machines (unbalanced scenario)

The multipliers of the processing time of 2.1 makes Machine 3A become the slower machine in the parallel system. Machine 3A is thus representing the "older" machine in the station. Whilst the 1.9 multiplier to machine 3B makes it become the faster and hence the "newer" machine in the parallel.

The values of 2.1 and 1.9 as multipliers have been empirically chosen as they lead to a proper level of imbalances among parallel machines. Indeed the machine 3A shows an average utilization of 97.6%, which is hence oversaturated with respect to the 93.2% utilization of the whole system, while the machine 3B is poorly saturated than other machines, as it has an average utilization of only 88.6%. This values have been calculated simulating the model with immediate release algorithms.

This scenario appears to be at the threshold, after which the next scenarios are too unbalanced to be considered, as they create strong intrinsic bottlenecks. Indeed, it has been checked that wider multipliers (such as 2.2 and 1.8, 2.3 and 1.7, etc.) cause an excessive and probably unrealistic imbalances of machines, with differences of saturation up to more than 20 percentage points. In the following table the main results of the empirical trials are reported.

Scenario	Multiplier	Imbalance (%)	Machine 3A Utilization (%)	Machine 3B Utilization (%)	Exit rate	Δ with standard exit rate (%)
Balanced	2-2	0%	93.2%	93.2%	0.03124	0%
Low imbalance	2.1-1.9	10%	97.6%	88.6%	0.03123	-0.05%
Moderate imbalance	2.15-1.85	15%	99.4%	86.3%	0.03114	-0.31%
High imbalance	2.2-1.8	20%	99.9%	83.9%	0.03089	-1.13%
Very high imbalance	2.3-1.7	30%	100.0%	79.3%	0.03022	-3.25%

Table 16 – Different unbalanced scenarios and their statistics

As it can be seen from the table above, for the scenarios of *high imbalance* (imbalance of 20%) and *very high imbalance* (imbalance of 30%) the difference in machine utilization becomes very important, and the exit rate decreases by more of the 1%. The *moderate imbalance* scenario, also, did not allow to reach the target obtaining an exit rate lower than the targeted one. Instead, the chosen multipliers of 2.1-1.9 (*low imbalance*) create imbalances in the system but do not lead to endless jobs waiting in queues that could obstruct the final production of jobs in the shop floor. The average ratio of jobs produced in the system and the jobs realized in the shop floor (output/input) is still around 97%, which can be regarded as an acceptable value. For the following reasons, the low imbalance scenario will be the one studied in this thesis. Hereafter, for a matter of simplicity it has been called the "*unbalanced scenario*". The results obtained in this scenario will be compared to the ones in the *balanced* one, in order to draw considerations on the performance of the algorithms and to test the robustness of the models to intrinsic imbalances in the system.

11.4 The effect of efficiency

As it has been said, with the workers reallocation the capacity of the system can be changed. In particular in the model, the two workers in the parallel station can be moved from one machine (machine 3A) to the other (machine 3B). Whenever a worker is moved to another machine, he will

use his time to help the other worker in performing the activities needed to complete the jobs. The capability of one worker to effectively carry out the activities in the other station is governed by the parameter of *efficiency*.

Indeed, a worker that possesses a value of efficiency of 100% in another machine, after staying 30 minutes in that station he will have carried out 30 minutes of workload to help the other. If the value of efficiency is instead 50%, if he stays 30 minutes only 15 minutes of effective workload will be delivered. The parameter of efficiency is considered as it is more realistic to assume that a worker would not have the same productivity in the other machines that he has in his default machine. Moreover, also not all the activities can be easily split in two, so when two workers will be on the same machine a percentage of time will inevitably be wasted. As written in Chapter 5.2, workers reallocation is most used in human-intensive environments, in which an additional worker can help the first worker for example in setup activities (placing instruments, pre-loading machines etc.) and in ending activities (placing finished products, setting up the machine for next jobs etc.). To study this effect, different levels of efficiency will be tested in the thesis.

To calculate the actual processing time of a job in a station where another worker has been reallocated, it will be used the following formula:

$$Actual\ processing\ time\ (i,j) = \frac{Processing\ time\ (i,j)}{\sum q\ Worker\ position\ (q,j) * \ Worker\ efficiency\ (q,j)}$$

where

- *Actual processing time (i, j)* is the actual time needed to perform a job i in a station j
- *Processing time (i, j)* is the time needed to perform a job i in a station j according to the processing time distribution adopted (truncated log-normal)
- *Worker position (q, j)* is a binary value that is 1 if the worker q is present at the station j and 0 if not
- *Worker efficiency (q, j)* is the value of efficiency, set as a percentage, that the worker q can deliver in station j

Indeed, if the processing time of a job is 30 minutes, and in that station there are two workers (one by default, with 100% efficiency, and one reallocated, with 50% efficiency) the actual processing time will be = 30 (minutes) / (100% + 50%) = 20 minutes.

11.5 Jobs' statistics considered

In the model, a job is an object that represents an order of a customer. As jobs may have very different number of lines, the processing time should reflect a significant variability (see Chapter 2.1). Then, to every job it will be associated the information on the *arrival date*, the *release date* and the *completion date*. Moreover, every job has its own *routing* (in the model there are two possible routings considered, see Chapter 8.1) and the *processing time* specific for every machine it needs to visit.

While a job follows its routing passing from machine to machine, its statistics are adjourned according to the different completions times. This is done in order to obtain statistics on the performance of the system in producing the requested jobs. The following jobs' statistics are calculated:

$$Gross\ Throughput\ Time\ (GTT) = Completion\ date\ (i) - Arrival\ date\ (i)$$

$$Shop\ Floor\ Throughput\ Time\ (SFT) = Completion\ date\ (i) - Release\ date\ (i)$$

GTT and SFT are the two most important performances evaluated. The SFT actually measures how long did it take for a job to pass the different production steps, so it represents the sum of the processing times in the machines and the time spent in the queues. Instead, the GTT, adds up to this measurement also the time that a job has spent in the PSP before being released. It indeed measures the total time that has passed from the acceptance of an order to its completion.

Then the statistics consider also the punctuality in completing jobs according to their due dates:

$$Lateness\ (i) = Completion\ date\ (i) - Due\ date\ (i)$$

$$Tardiness\ (i) = \max\ [\,0, Completion\ date\ (i) - Due\ date\ (i)]$$

$$Tardy\ (i) = \begin{cases} 1 & when\ \ Due\ date\ (i) < Completion\ date\ (i) \\ 0 & when\ \ Due\ date\ (i) \geq\ Completion\ date\ (i) \end{cases}$$

11.6 Workload norms

The workload literature does not refer to a scientific methodology to define and choose the workload norms to adopt in simulated studies. Rather, the norms are empirically defined as they strongly depend on the specific layout adopted.

The general rule is to consider a range of workload norms that range from a maximum and a minimum level. The norms are defined through a trial and error approach, where after simulating different values of them in the same system it is chosen a range of norms. In the thesis, the simulations performed to choose the values of norms have been done under the static type of workforce, in the "balanced scenario" discussed above.

To define the proper level of workload norms for the specific simulated model of the thesis the following steps have been followed:

1. Find the maximum value of workload norms after which the performances of the system in terms of GTT or SFT do not consistently improve. In other words, once found the maximum level of workload norms, further increasing its value will not bring any further benefit.

2. Find a minimum value of workload norms that drives queue lengths to converge to finite values. It is the opposite of the maximum value of workload norm found in the first point, as it is a stringent norm that strongly retains jobs in the pre-shop pool.

3. Choose values of workload norms in between the maximum and minimum norm respectively found in the first and second point. It is important to consider most those mid norms, since the maximum and minimum norms represent mainly the extreme points.

The values of workload norms that have been tried are 1900, 2000, 2200, 2400, 2800, 3600, 4800, 6600. The result is shown in next graph.

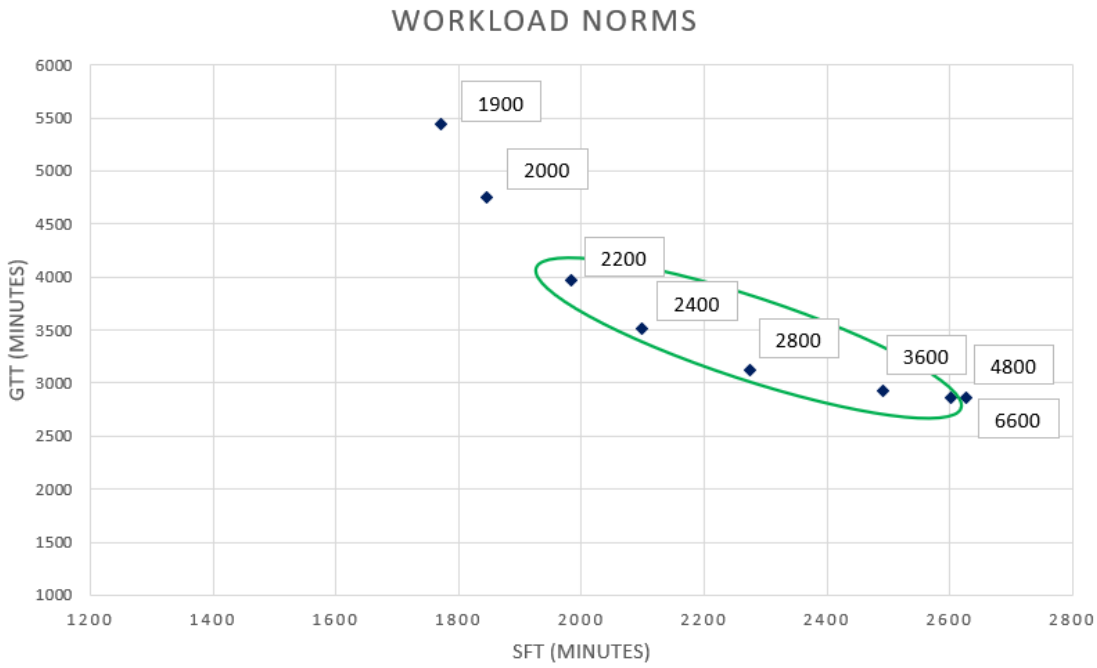


Figure 21 – Performance obtained with different norms (Workload limiting)

The first two norms (1900, 2000), which are the ones on the top left in the graph, have queue lengths not converging to finite values (it can be seen they follow an exponential trajectory), hence they have been excluded. Then, the highest norm (6600) shows the result which is in the far right of the graph. As it can be seen, this norm is not bringing any advantage to the performance if compared to norm 4800, so it must be excluded. The values of workload norms that have been chosen (the ones highlighted in green in the graph) are thus the following:

Shop configuration	Workload norms
Pure flow shop	2200, 2400, 2800, 3600, 4800

Table 17 – Workload norms chosen

11.7 Warm-up period, number of runs, length of the simulation

The goal of a simulation is to model a production environment that is as close as possible to a real context, and to obtain a result that is the least affected by randomness. This is done in order to be able to draft solid conclusions on the models built, which are not affected (or affected to the least

possible) by natural variations given by the probabilistic distributions used. So to reduce the variance between the experiments and to focus on the performance obtained with the parameters chosen in the simulation, a random technique is used for every replication. Therefore, each replication of the experiment will work with a different sample of jobs, while the same run of different experiments will face the same sample of customers orders. This is done to compare how the different models would work in solving the same situations.

The warm-up is the time needed by the system to be fully initialized, in which the queues start to fill and machines to process their first jobs. This period lasts until a steady state is reached by the system. For this reason, the experiments should not take into account the performance observed during this period. To calculate the length of the warm-up period, an empirical approach has been used. It has been taken the most important performance, i.e. the Gross Throughput Time, it has been calculated its average value between the different replications, and finally a moving average of 50 working days was built to reduce nervousness. The result is the following:

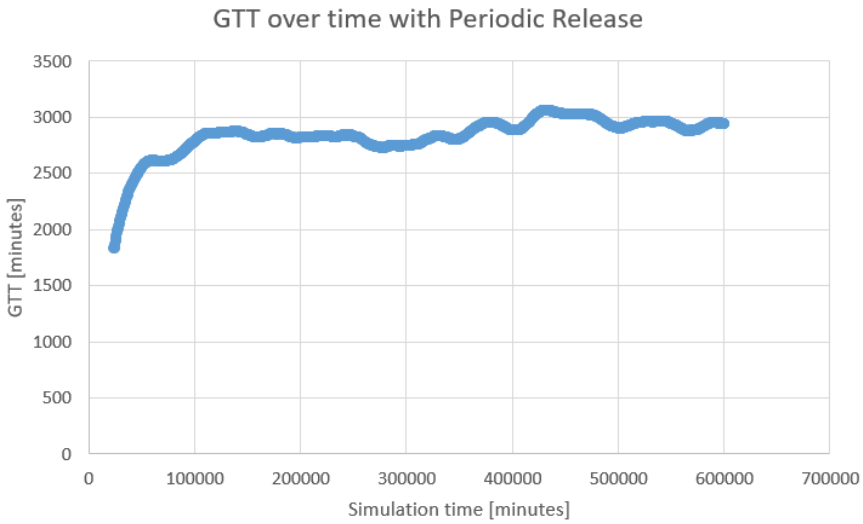


Figure 22 – Gross throughput time vs Simulation time

To avoid the impact of the model variability on the result, it has been chosen a warm-up period of 200,000 minutes (about 417 days), while the length of the simulation has been set to 500,000 minutes (about 1042 days).

Then, to define the number of runs to be performed for each experiment, it has been followed the method of the Mean Squared Pure Error (MSPE) for the most important performance, the Gross Throughput Time. The method consists in calculating the MSPE (formula below) for that performance, and to define the proper number of runs that stabilizes its value.

$$MSPE = \frac{\sum_{i=0}^{n}(x_i - \bar{x})^2}{n-1}$$

Where:

- x_i is the average value obtained in each run
- x is the average computed considering all the n runs performed

Considering Figure 23, the value of MSPE shows a convergence that starts around number 50. For this reason, the number of runs that will be used in this thesis for each experiment will be set to 50.

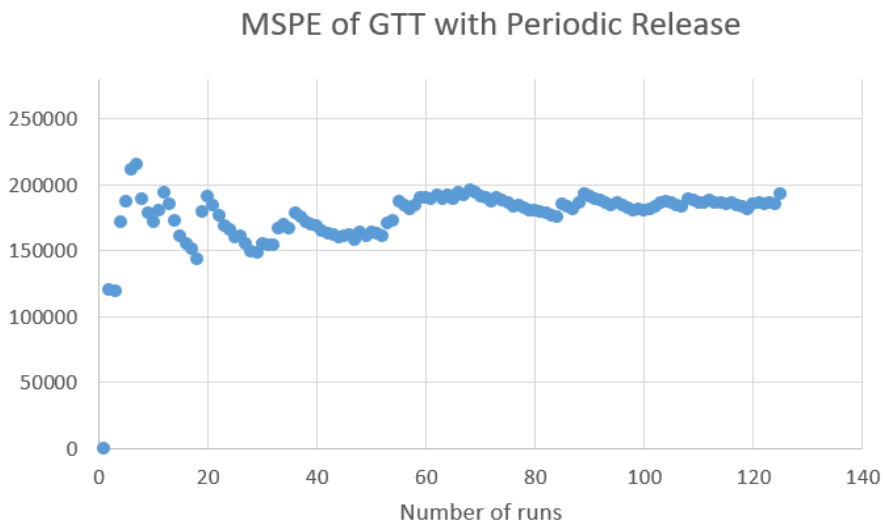


Figure 23 – MSPE vs Number of runs

12. Design of experiment

12.1 Parameters to study

In order to address the research questions, the following chapter makes a distinction of two types of parameters: the *system parameters* and *experimental parameters*.

The system parameters are the elements that are distinctive of the system and then they assume the same value during a simulation study. While the experimental parameters are variable of the specific model studied during the simulation. The values of experimental parameters is changed during experiments in order to analyze the response of the model to those changes.

The response is the performance under investigation, of which the most important are the Gross Throughput Time (time spent from a job's arrival to its completion) and the Shop floor Throughput Time (time spent from a job's release into the floor to its completion). A list with the explanation of also the other performances of the system that will be analysed is present at Chapter 12.2.

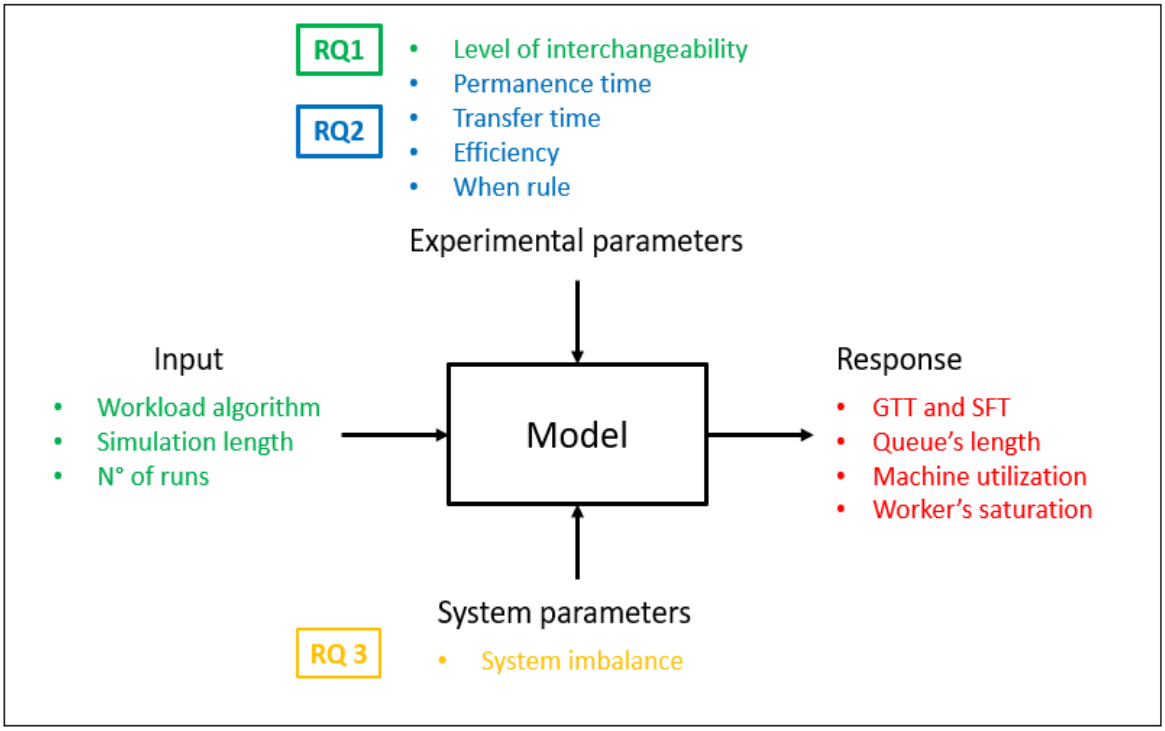


Figure 24 – Model's parameters

The first and second research question (respectively routing flexibility and workers reallocation) will be focused especially in measuring their performance at the variation of their experimental parameters. In the third research question, a confrontation of the different models is carried out. To this aim, the discussion will be focused on comparing the performance of the models and to compare their robustness at the variation of the most important system parameter: the system imbalance.

12.2 Design of experiment: the three research questions

Research question 1: Routing Flexibility

The table below shows the system parameters of the routing flexibility experiments and the experimental parameters with their respective levels (i.e. values). The rationale under the choice of figures of both parameters have been explained in the previous "Methodology" section.

Routing flexibility model	
System parameters	
PT mean	30
PT variance	900
Experimental parameters	
Level of interchangeability	0% 5% 10% 20% 50% 100%
Balance level	2-2 (balanced) 2.1-1.9 (unbalanced)
Performance to study	
GTT	
SFT	
Avg. queue lengths'	
Distribution of queue lengths'	
Machine saturation	

Table 18 – Routing flexibility model parameters and levels chosen

The parameter distinctive of the system (i.e. experimental parameters) in the case of routing flexibility is the *level of interchangeability,* represented by the percentage of jobs that can exploit routing flexibility. The system parameter studied is the *imbalance level*, considered as the two multipliers of the processing time of the parallel machines (see methodology for further explanations). In order to understand whether the routing flexibility is effective and under which model parameters, it has been done a "one-at-a-time" experiment. Such sets of experiments, shown in the table below, allow to understand to what degree the performance of the system have been affected by the level of interchangeability. Experiment 1 considers the system in the balanced scenario, while Experiment 2 repeats tests with the same levels of interchangeability but in the unbalanced scenario.

Experiment 1			
Code	Algorithm	Level of interchangeability	Balance level
1.0	Static	0%	2-2
1.1	Routing flexibility	5%	2-2
1.2	Routing flexibility	10%	2-2
1.3	Routing flexibility	20%	2-2
1.4	Routing flexibility	50%	2-2
1.5	Routing flexibility	100%	2-2
Experiment 2			
Code	Algorithm	Level of interchangeability	Balance level
2.0	Static	0%	2.1-1.9
2.1	Routing flexibility	5%	2.1-1.9
2.2	Routing flexibility	10%	2.1-1.9
2.3	Routing flexibility	20%	2.1-1.9
2.4	Routing flexibility	50%	2.1-1.9
2.5	Routing flexibility	100%	2.1-1.9

Table 19 – Routing flexibility design of experiments

Research question 2: Workers reallocation

The second research question aims at analyzing the system performance when the workers reallocation model has been applied. Similarly to the study on the first research question, also in this case different levels of the experimental parameters will be tested, and their effect on performances will be studied. The table below shows the system and model parameters of the workers reallocation simulation.

Workers reallocation model	
System parameters	
PT mean	30
PT variance	900
Experimental parameters	
Worker mode	Static Reactive
Efficiency	100% 75% 50% 25% 0%
Transfer time	0 10 15 20 30
Permanence time	0 30
When rule	Decentralised Centralised
Performance to study	
GTT	
SFT	
Workers' saturation	

Table 20 – Workers reallocation parameters and levels chosen

Given the many model parameters involved for a proper analysis of workers reallocation on parallel machines, the study has followed three steps.

STEP 1: it is the basic step which aims at understanding the impact of the reallocation of workers against the static case (where workers cannot be transferred), and the impact that the main experimental parameters of the model have on the results. The three main parameters tested have been:

- Transfer time

- Permanence time
- Workers' efficiency

First, an ANOVA analysis has been carried out in order to understand if all three of them had a statistically significant impact on the performance measured (GTT). To do that, it has been performed a "one-at-a-time experiment", considering two levels for each parameter (Experiment 3). The 3.1 refers to an ideal scenario, where all the levels have been set to their standard value. The 3.2 studies the effect of transfer time, the 3.3 of permanence time and 3.4 of the workers' efficiency. It has been chosen the one-at-a-time experiments in order to empirically assess the performances of the system at the variation of those parameters. To study the results, a Fractional Factorial experiment has been carried out through the software "Minitab".

Experiment 3					
Code	Algorithm	Transfer time	Permanence time	Efficiency	Balance level
3.0	Static	-	-	-	2-2
3.1	Workers reallocation	0	0	100%	2-2
3.2	Workers reallocation	**15**	0	100%	2-2
3.3	Workers reallocation	0	**30**	100%	2-2
3.4	Workers reallocation	0	0	**50%**	2-2

Table 21 – Workers reallocation first design of experiment: ANOVA of the three parameters

Finally, a further study focused on the most significant parameters has been performed to better address the sensibility of the system, considering in this case 5 different levels (Experiment 4).

Experiment 4					
Code	Algorithm	Transfer time	Permanence time	Efficiency	Balance level
4.0	Static	-	-	-	2-2
4.1	Workers reallocation	**0**	30	100%	2-2
4.2	Workers reallocation	**10**	30	100%	2-2
4.3	Workers reallocation	**15**	30	100%	2-2
4.4	Workers reallocation	**30**	30	100%	2-2
4.5	Workers reallocation	0	30	**25%**	2-2
4.6	Workers reallocation	0	30	**50%**	2-2
4.7	Workers reallocation	0	30	**75%**	2-2
4.8	Workers reallocation	0	30	**100%**	2-2

Table 22 – Workers reallocation second design of experiment: Transfer time vs Permanence time

STEP 2: once completed the STEP1 set of experiments and assessed the impact of workers reallocation parameters on parallel machines, the STEP2 aimed at assessing which *when rule* leads to better results. The when rule refers to the condition that triggers the transfer of workers between one machine to another. When the *decentralized* rule is considered, the model allows to move a worker to the other machine only when his machine is idle, so when his queue is empty. When the *centralized* is instead used, a worker can be reallocated to the other machine also when there are still jobs in his queues.

To carry out this experiment, different versions of the decentralized and centralized rule have been tested. The goal was to study how the GTT changed at the increase of the number of relocations occurring in the system. The difference between the different versions of centralized/decentralized,

regarded the threshold regarding the length of the two queues that made the algorithm activate and transfer the worker. The design of experiment is reported below:

Experiment 5						
Code	Algorithm	When rule	Threshold for algorithm activation*	Transfer time	Efficiency	Balance level
5.0	Static	-	-	-	-	2-2
5.1	Workers reallocation	Centralized	Q1>50 and Q1>1.5*Q2	10	75%	2-2
5.2	Workers reallocation	Centralized	Q1>20 and Q1>2.0*Q2	10	75%	2-2
5.3	Workers reallocation	Centralized	Q1>3.0*Q2	10	75%	2-2
5.4	Workers reallocation	Decentralized	Q2=0	10	75%	2-2
5.5	Workers reallocation	Decentralized	Q2=0 and Q1>100	10	75%	2-2
5.6	Workers reallocation	Decentralized	Q2=0 and Q1>500	10	75%	2-2
5.7	Workers reallocation	Decentralized	Q2=0 and Q1>900	10	75%	2-2
5.8	Workers reallocation	Decentralized	Q2=0 and Q1>1500	10	75%	2-2

Table 23 – Workers reallocation third design of experiment: Centralized vs Decentralized

*To understand this notation: the workers reallocation works between two machines. In case for example Q1>2.0*Q2, the worker from the second machine is reallocated to the first one whenever the queue of the first one is greater than the double of the second one. This means that if Q1=60 minutes, the worker from the second machine can help the worker of the first one only if Q2 is less than 30 minutes. The other notation, Q1>50, means that the reallocation is considered only after a certain threshold of queue length, in order not to make the system too nervous. For the decentralized case, instead, Q2 must be empty to consider reallocation, and the threshold refers to the minimum Q1 length that if surpassed it triggers the relocation (for example Q1>500).

The results in terms of GTT will be used to build a curve, which will consider the average GTT according to the different number of relocations performed.

STEP 3: the last step regarded instead to test the model robustness studying the performance obtained in two different scenarios: the balanced and the unbalanced scenario. In the balanced scenario, the two parallel machines work at the same speed, while in the unbalanced, one machine is 10% faster than the other (see methodology part for further description).

Experiment 6 refers to the comparison between the two scenarios:

Experiment 6						
Code	Algorithm	When rule	Transfer time	Permanence time	Efficiency	Balance level
6.0	Static	-	-	-	-	2-2
6.1	Workers reallocation	Decentralized	10	30	75%	2-2
6.2	Static	-	-	-	-	2.1-1.9
6.3	Workers reallocation	Decentralized	10	30	75%	2.1-1.9

Table 24 – Workers reallocation fourth design of experiment: Balanced vs Unbalanced scenario

Research question 3: Routing flexibility vs Workers reallocation

The last research question of the thesis aims at comparing the two models proposed, to study and to compare in detail their performance. The first experiment of this section (Experiment 7) aims at evaluating the improvement in GTT and SFT that are respectively brought by the two models. The configuration chosen is the one obtained throughout the two previous research questions. Then, a further study is performed to compare the robustness of the two models against the unbalanced case (Experiment 8). Although routing flexibility and workers reallocation have the same objective of balancing workload on the shop floor, they work in a different way and have their own model parameters. As a consequence, when comparing them it was not possible to change their specific model parameters. Rather, it has been changed the parameters that they have in common, which are

the ones that refer to the balance level. In such a way it is possible to compare how the two models react to strong imbalances in the system.

Experiment 7						
Code	Algorithm	Inter-changeability	When rule	Transfer time	Efficiency	Balance level
7.0	Static	-	-	-	-	2-2
7.1	Workers reallocation	-	Decentralized	10	75%	2-2
7.2	Routing flexibility	20%	-	-	-	2-2
Experiment 8						
Code	Algorithm	Inter-changeability	When rule	Transfer time	Efficiency	Balance level
8.0	Static	-	-	-	-	2.1-1.9
8.1	Workers reallocation	-	Decentralized	10	75%	2.1-1.9
8.2	Routing flexibility	20%	-	-	-	2.1-1.9

Table 25 – Workers reallocation vs Routing flexibility in balanced and unbalanced scenario

These two experiments will be deeply investigated by considering the performance in terms of GTT and SFT, but also the number of relocations per year required to obtain those results in finally the performance in terms of workers' saturation.

Part 3: Empirical results

13. Discussion of results

This chapter contains the empirical results that have been collected in the study of this thesis to answer the three research questions. The chapter is divided into three main sections:

- The first section aims at studying the model of *routing flexibility*. It starts with the analysis of how the different levels of interchangeability affect the results, and it considers the marginal improvements in performance given by the increase in interchangeability. The model is then tested in the unbalanced scenario. The goal of the first part is to give a complete perspective on how the *routing flexibility* performs according to the parameter of interchangeability and to the imbalance level present in the system.

- The second section aims at studying the model of *workers reallocation.* The main parameters (transfer time, permanence time, efficiency) are tested through the use of an ANOVA analysis. Then, the two "when" rules: decentralized and centralized, are studied. Finally, similarly to the first section, the model is tested in the unbalanced scenario. The goal of this second section is to understand how the different parameters of the workers reallocation affect the model.

- The last section, instead, contains a comparison between the *routing flexibility* and the *workers reallocation* model. The models are tested both in the balanced and unbalanced scenario, and their performances are deeply investigated under different perspectives. The goal of this last part is to assess and compare how the models behave, and to evaluate the number of relocations required by both of them to obtain the improvements.

13.1 Routing flexibility: the level of interchangeability

The aim of the first experiment is to study how the level of interchangeability (i.e. the percentage of jobs that can be moved from one parallel station to the other and vice versa) affects the performances of gross throughput time (GTT) and shop floor throughput time (SFT). Five levels of interchangeability are tested (5%, 10%, 20%, 50%, 100%) and they are compared against a control case (in which the interchangeability has been set to 0%, so no routing flexibility is applied). In order not to bias the results, the same release algorithm has been used in this experiment, which is the workload limiting algorithm, and the workforce has been considered static.

For transparency reasons, the first experiment is set in a *balanced* scenario. In this scenario the parallel machines are identical (they both have a processing time that is 2x the processing time of the others, see Chapter 11.3). The algorithm of routing flexibility will have to face the imbalances that occur due to the high variability of the jobs, solving the bottlenecks between the two parallel stations by re-allocating jobs between each other.

The same workload norms are tested for each case (2200, 2400, 2800, 3600, 4800). Generally, for low workload norms jobs are retained more time in the PSP before being released. In these cases, the GTT is going to be higher and SFT lower. Instead, with high workload norms, more jobs are immediately released, and they generally spend more time in queues in the shop floor, obtaining a lower GTT but higher SFT.

In the following graph, it is reported the results of the application of the routing flexibility algorithm to the different levels of interchangeability:

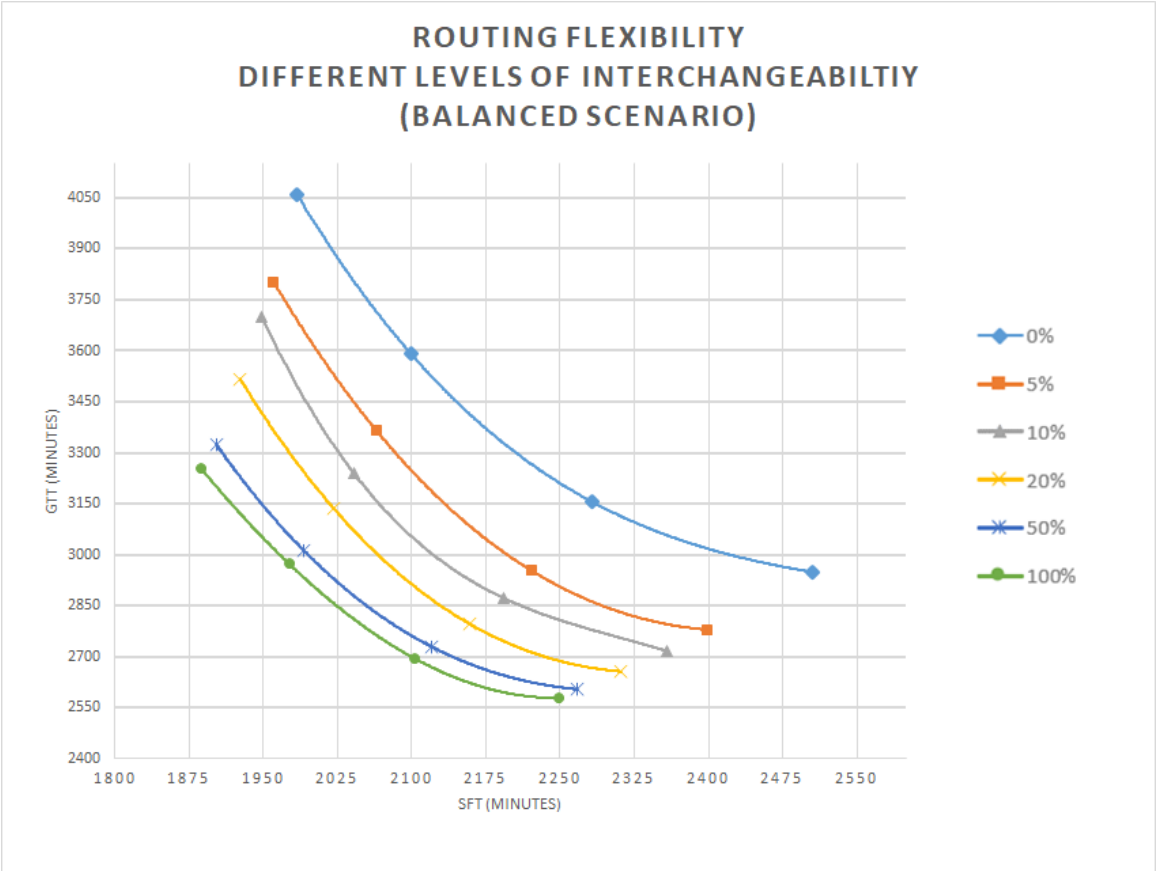


Figure 25 – Lead time performance of routing flexibility with different levels of interchangeability

The performance curves in Figure 25 are based on different degrees of interchangeability, going down from top to bottom: 0%, 5%, 10%, 20%, 50% and finally 100%. Obviously, a higher level of interchangeability always improve both the performances. What is interesting from the figure is that already a small level of interchangeability (5%) brings to a relatively large improvement in performance compared to the case of no routing flexibility (0%). Further increases in interchangeability lead to marginal improvements of overall performances, for example the distance between 50% and 100% is smaller than the one from 0% to 5% or from 5% to 10%.

Anyway, the improvements in performance in the figure due to the routing flexibility are evident. By simply re-allocating jobs between the two parallel machines, the algorithm is able to obtain better results both in terms of GTT and SFT. This comes from the fact that queues in the parallel stations are indeed more balanced, which reduces the occurrence of bottlenecks (as it will be demonstrated in Chapter 13.2). The other conclusion that can be drafted from these curves, is that the marginal benefits given by the routing flexibility is already relatively high from low levels of interchangeability, leading only to marginal improvements with high values of interchangeability.

Following, a table with the complete results of this experiment is reported (Experiment 1). Then, an analysis will be carried out regarding the different effects on performance that routing flexibility has when low or high norms are applied. Finally, the distribution and length of queues will be studied, in order to explain the reason behind the improvements obtained in the system.

Model: Routing flexibility					
Balance level	2-2 (balanced scenario)		Routing flexibility	Between machine 3A and 3B	
Release algorithm	Workload limiting		Workforce	Static	
Level of inter-changeability	Workload norm	Avg. GTT	Avg. SFT	Δ GTT with no RF (%)	Δ SFT with no RF (%)
0% (No Routing Flexibility)	2200	4058	1984	-	-
	2400	3590	2100		
	2800	3155	2283		
	3600	2949	2505		
	4800	2897	2626		
5%	2200	3801	1961	-6.3%	-1.2%
	2400	3365	2064	-6.3%	-1.7%
	2800	2952	2221	-6.4%	-2.7%
	3600	2779	2400	-5.8%	-4.2%
	4800	2750	2493	-5.1%	-5.1%
10%	2200	3701	1948	-8.8%	-1.8%
	2400	3240	2041	-9.8%	-2.8%
	2800	2872	2193	-9.0%	-3.9%
	3600	2716	2358	-7.9%	-5.9%
	4800	2692	2439	-7.1%	-7.1%
20%	2200	3517	1926	-13.3%	-2.9%
	2400	3135	2021	-12.7%	-3.7%
	2800	2797	2158	-11.3%	-5.5%
	3600	2656	2312	-9.9%	-7.7%
	4800	2634	2385	-9.1%	-9.2%
50%	2200	3325	1902	-18.1%	-4.1%
	2400	3012	1991	-16.1%	-5.2%
	2800	2729	2120	-13.5%	-7.1%
	3600	2603	2268	-11.7%	-9.5%
	4800	2580	2332	-11.0%	-11.2%
100%	2200	3250	1888	-19.9%	-4.9%
	2400	2972	1976	-17.2%	-5.9%
	2800	2693	2104	-14.6%	-7.8%
	3600	2578	2249	-12.6%	-10.2%
	4800	2558	2311	-11.7%	-12.0%

Table 26 – Complete results of Routing flexibility in the balanced scenario

To better investigate the benefits in terms of improvements in GTT and SFT, it has been decided to consider the effect that routing flexibility had on a low and a high norm (respectively 2400 and 3600). As it can be seen for Table 27, for *low norms* the percentage improvement in GTT is generally higher than for *high norms*. The difference in improvement increases at the increase of the interchangeability level, reaching with 100% interchangeability a -17.2% of GTT with low norms against a -12.6% with high norms. This difference between high and low norms is specularly reflected regarding the SFT, where for high norms the percentage of improvement is generally the double than the one obtained with low norms.

Improvements GTT		
	Av. GTT	
Interchangeability	Low norm	High norm
0%	0%	0%
5%	-6.3%	-5.8%
10%	-9.8%	-7.9%
20%	-12.7%	-9.9%
50%	-16.1%	-11.7%
100%	-17.2%	-12.6%

Improvements SFT		
	Av. SFT	
Interchangeability	Low norm	High norm
0%	0%	0%
5%	-1.7%	-4.2%
10%	-2.8%	-5.9%
20%	-3.7%	-7.7%
50%	-5.2%	-9.5%
100%	-5.9%	-10.2%

Table 27 – GTT and SFT improvements with Routing flexibility for low and high norms

The reason for this difference in improvements is that the GTT is generally high for *low norms* because orders are being held long time in the PSP not to exceed the workload norm (see Figure 4 for example). Hence the routing flexibility, by better balancing the two queues, decreases the occurrence of bottlenecks (i.e. the creation of a long queue before one of the two parallel machine). This allows jobs to wait a shorter time before being released from the PSP, improving in this way their GTT .

Regarding the SFT, instead, the improvement is greater in case of *high norms*. The reason for this is that with high norms the system is already releasing high amounts of orders into the system, which however end up queueing up before the stations. Due to long queues, jobs may take a long time to conclude their routing, increasing their SFT. Moreover, with high norms, the great number of jobs in the system causes frequent imbalances among machines. The positive fact of this is that having many jobs in the system in turn drives a higher probability to exploit the routing flexibility to balance the workload. In poor words, the higher the number of jobs within the system, the more possibilities have the routing algorithm to balance the queues. This is reflected by the higher improvement obtained in SFT in case of high norms compared to the low norms.

The other important aspect that emerges from Table 27 is that, at the increase of the interchangeability level, the rate of improvement in GTT and SFT marginally decreases. This means that, after a certain threshold, improving the interchangeability level brings only marginal results, as most of benefits have already been achieved. The following tables show the marginal improvement in GTT and SFT at the increase of interchangeability level.

Figure 26 – Marginal improvements of routing flexibility with low and high norms

With only the 20% of interchangeability, the model already achieves on average the 73% of the benefits achievable with a complete interchangeability. Especially for high norms, this level of interchangeability is sufficient to get most of the benefits (79.0% and 75.6% respectively in GTT

and SFT of total benefits achievable with 100% interchangeability). This occurs probably because few changes in routings are needed to balance the queues in the system, hence delivering a better SFT and GTT, as explained before.

In case of a realistic scenario, obtaining a 20% interchangeability between the orders is more practical than a 100%. This is because in the first case the machines do not have to be identical, but they can be different as one can be used for working smaller dimensions and the other greater dimensions. Moreover, there is less need to change the job's routings between one machine and the other, as only the 20% and not the 100% may be moved. This means that the system will be less nervous to small variations in the workload, obtaining a more stable production flow. Indeed, changing the 100% of the routings in a shop floor is extremely complex, as in a real industry many activities should be scheduled and prepared before (Henrich, Land, & Gaalman, 2007).

For all these reasons, a scenario of 20% of interchangeability appears to be more realistic, while still being able to deliver most of the benefits.

13.2 Routing flexibility model: queues' distribution and length

In the previous chapter it has been reported that with high norms (3600), the application of routing flexibility with a 20% level of interchangeability has led to a reduction in GTT and SFT respectively of -9.9% and -7.7% compared to the static case. The following analysis will investigate the reasons behind this reduction by studying the distribution and length of the queues before the two parallel machines.

Considering the model, the algorithm tries to balance the workload of the two parallel machines (3A and 3B) by re-allocating the interchangeable jobs between them. This is done according to the forecasted workload expected for the two machines, (see Chapter 10.1). An efficient routing flexibility would result in a better balance of the two queues before the two stations (queue 3A and 3B). Balancing the two queues means reducing their average difference in terms of queue length (calculated in minutes of workload). It is expected that the with the application of routing flexibility, the length of queues 3A and 3B will be more similar throughout the simulation time. Moreover, another expected result is that the system will improve its ability in solving the queues, hence that the new queues will have a lower average length and a lower variability.

This occurs because if for example there is no routing flexibility, there may be one queue in a bottleneck situation, while the other machine is instead empty. With the balancing done by the routing flexibility, instead, the queue in the bottleneck station would be split between the two machines, leading to a faster solution of the queues, thus decreasing the average queue length time. As a consequence of balancing, decreasing the occurrence of bottlenecks will also reduce the variability in the queues lengths', leading to a more stable system.

In Figure 27, it is reported the observed distribution of queue lengths in the experiment. The comparison is done between the case routing flexibility with 20% of interchangeability and the control case of no routing flexibility. To read the graph:

- Bar of color light blue: How many times it has been recorded a queue of such length in the control case.
- Bar of color light red: How many times a queue of such length in the routing flexibility 20% case.
- Bar of color purple: When the two bars overlap.

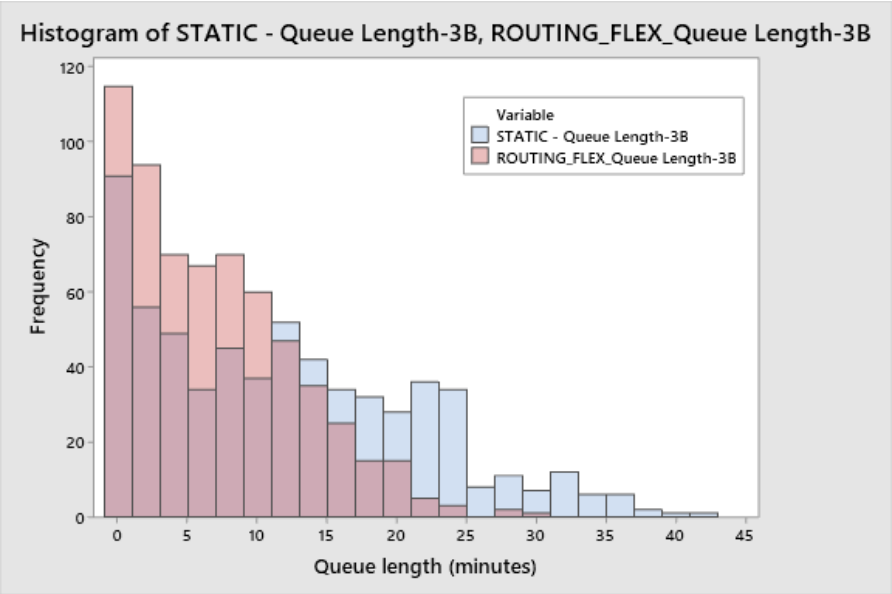


Figure 27 – Queue 3B distribution for the static (blue) and routing flexibility (red) case

As it can be seen from figure, the distribution of the queue lengths has moved from being spread with a high variability with several high values of queue lengths (light blue bars), to be more concentrated in the short lengths (light red bars). The results are significant, as the average queue length time has passed from being 11.92 minutes (with no routing flexibility) to 6.90 minutes (with 20% routing flexibility), and its standard deviation from 9.54 minutes to 5.86 minutes.

Furthermore, the graph clearly shows how the occurrence of peaks in queues (more than 20 minutes) is drastically reduced when routing flexibility is applied. Such decrease of peaks is relevant for the GTT, since those long queues are likely to be the ones that prevented jobs from being released from the pre-shop pool, causing a long GTT. Thus, the strong reduction of these long queues may also explain the consistent improvement of GTT achieved through routing flexibility (-9.9%).

Figure 27 reported the queue before machine 3B, while for the queue of machine 3A the effect of the algorithm is similar and its graph is the following:

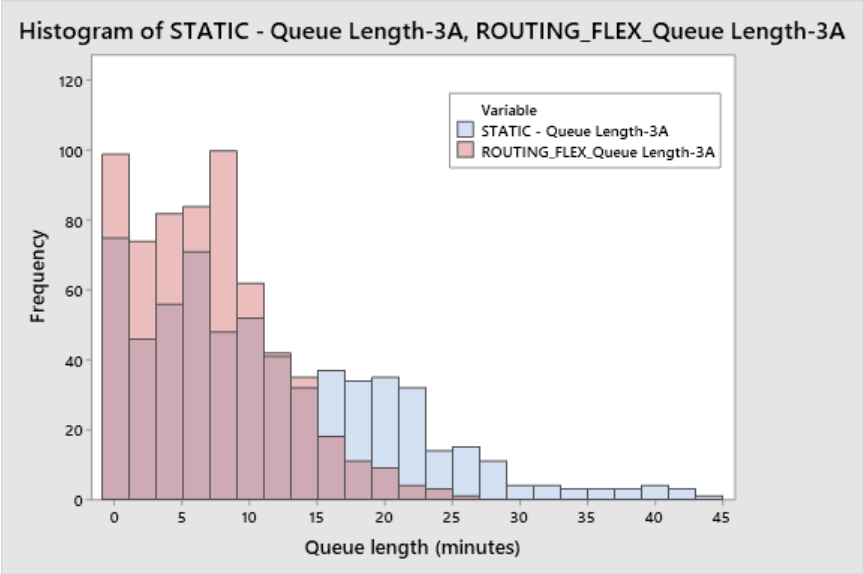


Figure 28 – Queue 3A distribution for the static (blue) and routing flexibility (red) case

The summary of the study of the queues length and distribution is in the following table (data is reported in minutes):

	No routing flexibility		Routing flexibility (20%)	
	Queue 3A	Queue 3B	Queue 3A	Queue 3B
Avg. queue length	13.7	14.3	6.8	6.9
Std. deviation of queue length	11.0	11.5	5.2	5.9
Avg. difference in queue length	-0.6		-0.1	
Std. deviation of difference	17.2		5.5	

Table 28 – Queues' statistics of Static vs Routing flexibility case

The results presented Table 28 show that not only the average queue length and std. deviation of both queues have decreased when applied routing flexibility, but also that the variability of the

difference between the two queue lengths has recorded a drop (from 17.2 minutes to 5.5 minutes). In other words, with the routing flexibility the two queues are reduced, but they are also more balanced. The increased balance is responsible for the decrease of SFT measured, as the two machines are more able to quickly solve the queues delivering jobs in a shorter time.

To conclude, it has been shown that routing flexibility has a clear and important effect on the performance of a system with parallel machines. Then it has been demonstrated that a level of interchangeability of 20% is sufficient to bring nearly the 80% of results, especially in case of high norms used. Finally, it has been shown how the distribution of the queues lengths varies thanks to the routing flexibility, leading to a reduction of both average length and variability in the queues' distribution, which explain most of the reduction of GTT and SFT observed.

13.3 Routing flexibility model: unbalanced scenarios

The last part of the analysis refers to create a system with intrinsic imbalances, and to observe the effects of the routing flexibility algorithm on performances. Systems with intrinsic imbalances are often found in real applications, for example production lines with machines working batches of different quantities and with different processing times. In a parallel shop configuration, an intrinsic imbalance can be due simply to the existence of new and old machines working on the same production flow. As stated by Miragliotta and Perona (2010) and by Henrich, Land and Gaalman (2006), models in literature often take into consideration machines with the same processing times, which is often a limit that may lead to derive guidelines that are far from real applications.

As one of the purpose of this thesis is to create a model as much as possible close to reality, it has been considered to implement in the model some intrinsic imbalances, and to test how the routing flexibility algorithm performs in this case. In this part, the *unbalanced scenario* will be tested. In this, the processing time of the two parallel machines will be x2.1 for the slower and x1.9 for the faster machine, which leads to an imbalance of 10% (see methodology part for further explanation). This is the modeling of a system with an intrinsic imbalance caused by a newer and hence faster machine, and an older hence slower one.

The aim of routing flexibility in this case is to solve the bottlenecks represented by the slower machine with the jobs' reallocation, in order to improve the time performance of the system. The results are presented in the following figure:

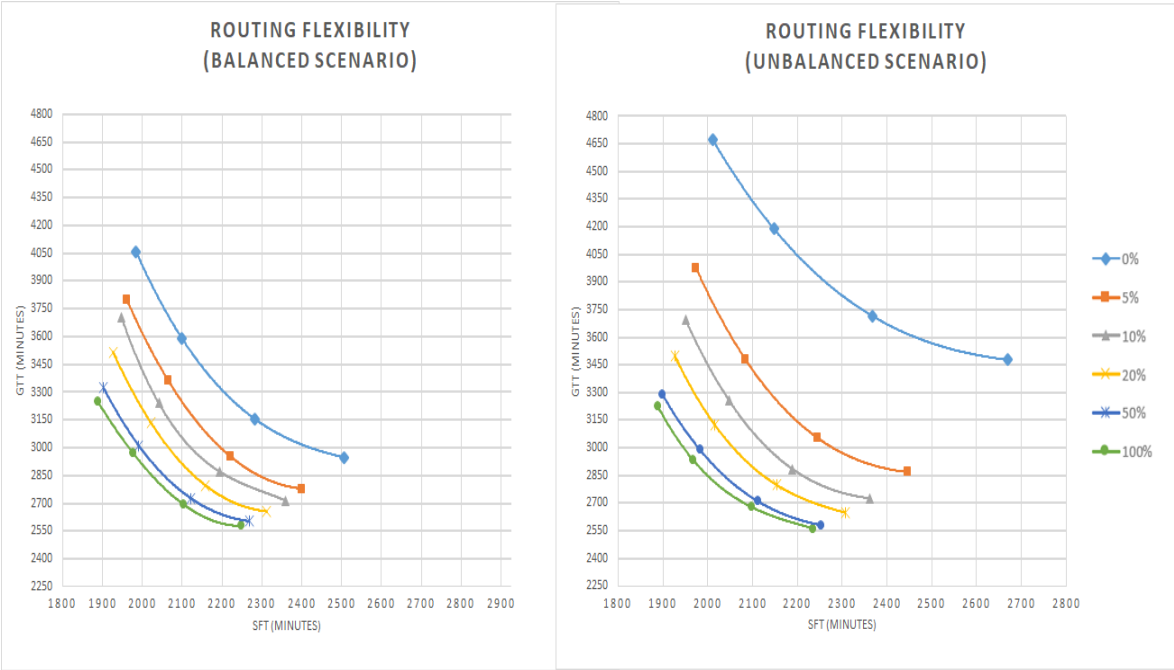


Figure 29 – Lead time performance of routing flexibility for the balanced and unbalanced case

As it can be seen from Figure 29, when no routing flexibility is applied (0%), the unbalanced scenario starts from a significantly worse point than the balanced. Anyway, routing flexibility has an important effect already for low levels of interchangeability. Indeed, in the unbalanced scenario it is sufficient to have a level of interchangeability of 10% to obtain respectively the 82% and and the 70% of the total benefits achievable with the complete routing flexibility. The first conclusion that can be drawn is that, the higher is the imbalance in the system, the lower is the level of interchangeability required to obtain most of the benefits.

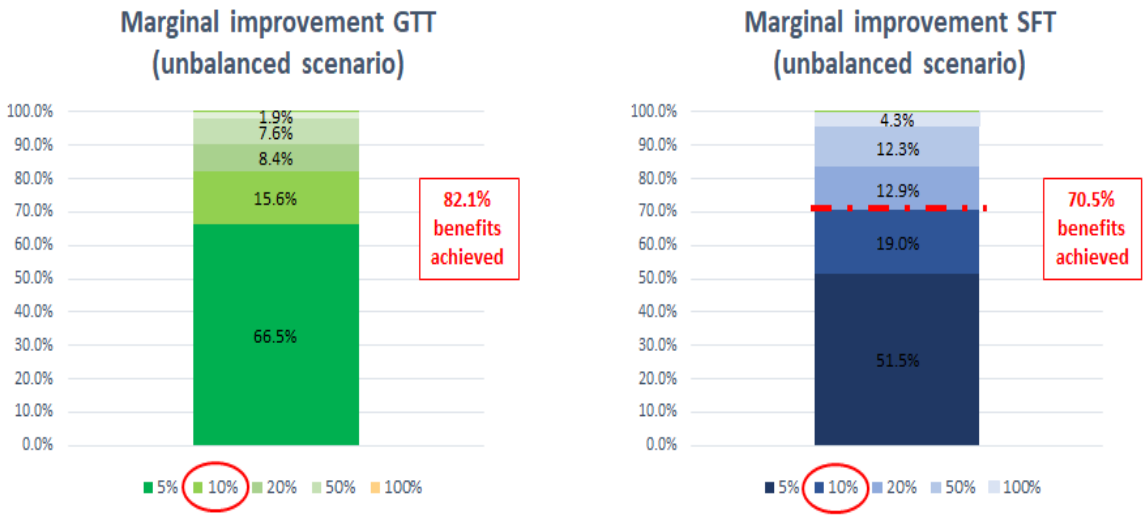


Figure 30 – Marginal improvements in GTT and SFT in the unbalanced scenario

Model: Routing flexibility					
Balance level	2.1-1.9 (unbalanced scenario)		Routing flexibility	Between machine 3A and 3B	
Release algorithm	Workload limiting		Workforce	Static	
Level of inter-changeability	Workload norm	Avg. GTT	Avg. SFT	Δ GTT with no RF (%)	Δ SFT with no RF (%)
0% (No Routing Flexibility)	2200	4674	2011	-	-
	2400	4188	2148		
	2800	3716	2367		
	3600	3477	2670		
	4800	3439	2940		
5%	2200	3976	1974	-14.9%	-1.8%
	2400	3481	2084	-16.9%	-3.0%
	2800	3055	2244	-17.8%	-5.2%
	3600	2868	2447	-17.5%	-8.4%
	4800	2828	2555	-17.8%	-13.1%
10%	2200	3697	1951	-20.9%	-3.0%
	2400	3259	2047	-22.2%	-4.7%
	2800	2885	2189	-22.4%	-7.5%
	3600	2725	2362	-21.6%	-11.5%
	4800	2705	2449	-21.3%	-16.7%
20%	2200	3497	1927	-25.2%	-4.2%
	2400	3124	2016	-25.4%	-6.1%
	2800	2798	2154	-24.7%	-9.0%
	3600	2648	2307	-23.8%	-13.6%
	4800	2631	2382	-23.5%	-19.0%
50%	2200	3292	1898	-29.6%	-5.6%
	2400	2989	1983	-28.6%	-7.7%
	2800	2710	2112	-27.1%	-10.8%
	3600	2580	2252	-25.8%	-15.6%
	4800	2565	2318	-25.4%	-21.2%
100%	2200	3225	1889	-31.0%	-6.1%
	2400	2935	1967	-29.9%	-8.4%
	2800	2679	2097	-27.9%	-11.4%
	3600	2561	2236	-26.3%	-16.3%
	4800	2546	2299	-26.0%	-21.8%

Table 29 – Complete results of Routing flexibility in the unbalanced scenario

By analyzing the machine saturation, similar conclusions can be drawn (Table 30) as most of the imbalance between the two parallel stations is solved with the 10% of interchangeability, while further increases lead to smaller improvements.

Interchangeability level	Machine 1	Machine 2	Machine 3A	Machine 3B	Machine 4	Machine 5
0%	93.4%	93.5%	**97.6%**	**88.6%**	93.2%	93.0%
5%	93.4%	93.5%	**95.7%**	**90.3%**	93.2%	93.1%
10%	93.5%	93.5%	**94.9%**	**91.0%**	93.2%	93.1%
20%	93.5%	93.5%	**94.2%**	**91.6%**	93.2%	93.1%
50%	93.5%	93.5%	**93.5%**	**92.2%**	93.2%	93.1%
100%	93.5%	93.5%	**93.3%**	**92.4%**	93.2%	93.1%

Table 30 – Machines' saturation in the unbalanced scenario

The conclusion of this last part of routing flexibility is that, in a system with different processing times, the higher the imbalances are the more effective the routing flexibility is, leading to significant results already for low levels of interchangeability. Indeed, in a real application, it would be worth investing to increase the level of interchangeability up to levels of 10-20%, while further levels would bring only limited additional improvements.

13.4 Workers reallocation

The second proposed model considers a reactive workforce, in which the workers of the two parallel machines (3A and 3B) can be transferred between the two machines. This is done to solve bottlenecks by temporarily shifting capacity from one machine to the other. The reallocation is governed by an algorithm, which was presented in Chapter 10.1 and has the same purpose of routing flexibility: creating a smooth and balanced production flow to improve system performances, in particular the gross throughput time (GTT) and shop floor throughput time (SFT).

In the proposed model of workers reallocation, there are several parameters that regulate its functioning: the *worker's efficiency* is the percentage that indicates the ability of a worker in a

certain machine (a worker with 50% efficiency staying 30 minutes in a machine will deliver 15 minutes of workload), the *transfer time* which is the time needed to move a worker from its current machine to another, the *permanence time* which is the minimum time for which a worker is obliged to stay in a machine when relocated.

The first part of the analysis aims to study the effect that these three parameters (efficiency, transfer time, permanence time) have on the worker reallocation model. The results will be tested with an ANOVA analysis, in order to obtain statistically significant indications whether the parameter impact the model. The second part of the analysis will instead be focused on the application of two different "when" rules, in particular the *decentralized* and *centralized*. In the first one, a worker can be reallocated in the other machine only when there are no jobs in his queue, while in the second one this constraint is released, in order to allow a higher number of relocations of workers. Finally, the last part of the chapter will be related to test the model in the unbalanced scenario, in order to verify its robustness in different scenarios.

13.5 Workers reallocation: the impact of efficiency, transfer time and permanence time

To test the effect of these three parameters, a one-at-a-time experiment has been designed. The workers reallocation has been applied leaving the three parameters to vary one at a time. The levels considered are 100% and 50% for *efficiency*, 0 and 15 minutes for the *transfer time*, 0 and 30 minutes for the *permanence time*. In these cases, the first value mentioned refer to an ideal scenario (full efficiency, no transfer time or permanence time), while the second is an important variation of the parameter.

As explained in the methodology section, the reallocation between workers is constrained only to the two parallel machines, allowing movements of workers 3A and 3B only between the two machines. Similarly to routing flexibility, the experiment has been conducted considering the workload norm of 3600.

The results are the summarised in table:

Workforce	Parameters	Avg. GTT	Std. dev	Confidence level (90%)
Static	-	2949	487	[3575, 2323]
Reactive (Ideal scenario)	EFF 100% TT 0 PT 0	2583	401	[3098, 2069]
Reactive	**EFF 50%** TT 0 PT 0	2716	417	[3252, 2181]
Reactive	EFF 100% **TT 15** PT 0	2687	408	[3211, 2163]
Reactive	EFF 100% TT 0 **PT 30**	2580	392	[3085, 2076]

Table 31 – Parameters and levels chosen for the ANOVA analysis

As it can be seen from Table 31, the performance of GTT appears to be strongly affected by both the reduction of *efficiency* (case with efficiency 50%) and the increase in *transfer time* (case of 15 minutes) as by changing the level of these parameters from the ideal case it has been obtained an important worsening of the GTT. The parameter of *permanence time*, instead, seems not to influence considerably the result. Anyway, since the standard deviation is high for all these cases and the confidence levels are partially overlapping, an ANOVA analysis has been carried out:

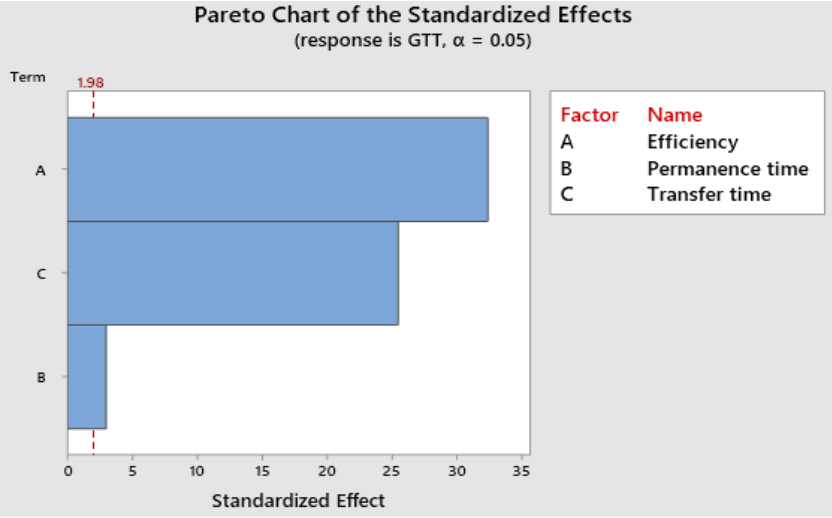


Figure 31 – Pareto chart of the effects of the three parameters (from Minitab)

In Figure 31 it is shown the result of the ANOVA analysis. The model obtained has an R^adjusted of 99.58%, which means that the parameters well explain the variations in the model. As shown in Figure 31, all three parameters are statistically significant. However, from the graph it is clear that the influence of *efficiency* and *transfer time* is significantly higher than the one of *permanence time* (first two bars against the third).

Following, the three hypothesis of ANOVA are reported. The normality of residuals and the independence of residuals (Figure 32 and 33 on the left) are both respected. The last one, the equal variances of residuals is not respected (Figure 33 on the right), which is probably given by the high variability that there is between each run. Anyway, this does not have a considerable effect on the conclusions drafted from the analysis regarding the three parameters.

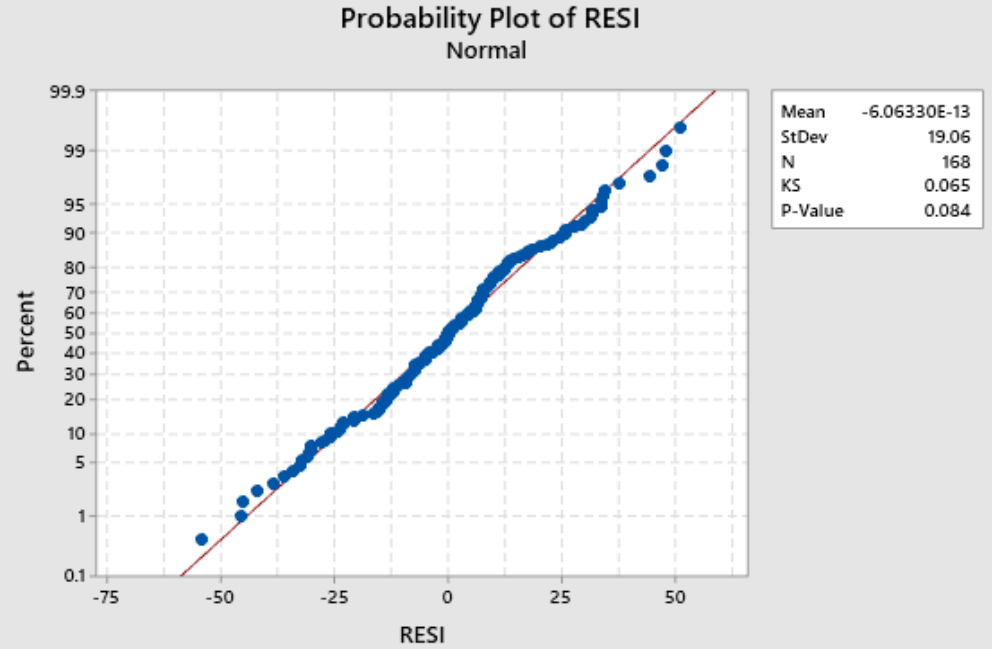


Figure 32 – The hypothesis of normality test of residuals (from Minitab)

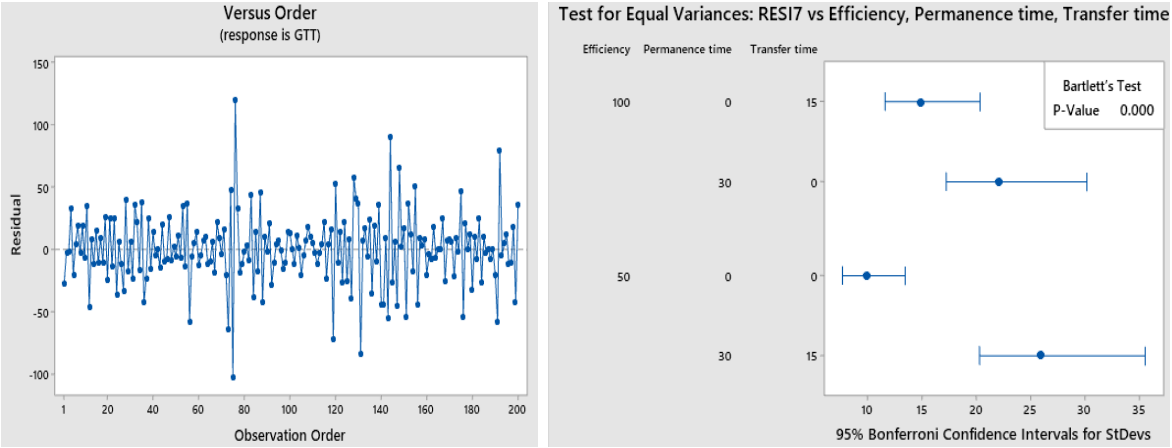


Figure 33 – The independence of residuals (left), and the equal variances of residuals (right)

To conclude the first part of analysis, all three parameters are statistically significant. Anyway, the *permanence time* affects the result to a lower degree than the *efficiency* and the *transfer time* of the reallocated workers.

For the two most significant parameters, the efficiency and the transfer time, a further analysis has been performed studying the effect on GTT with five different levels. Regarding efficiency, it has been tested starting from the static case of 0%, to 25%, 50%, 75%, 100%. For the transfer time, from the static case to 0, 10, 15, 20, 30 minutes. To summarize, five levels of the two parameters have been studied separately, and their result has been grouped in 5 cases in order to compare the respective effect on performance. The results are shown in the following table:

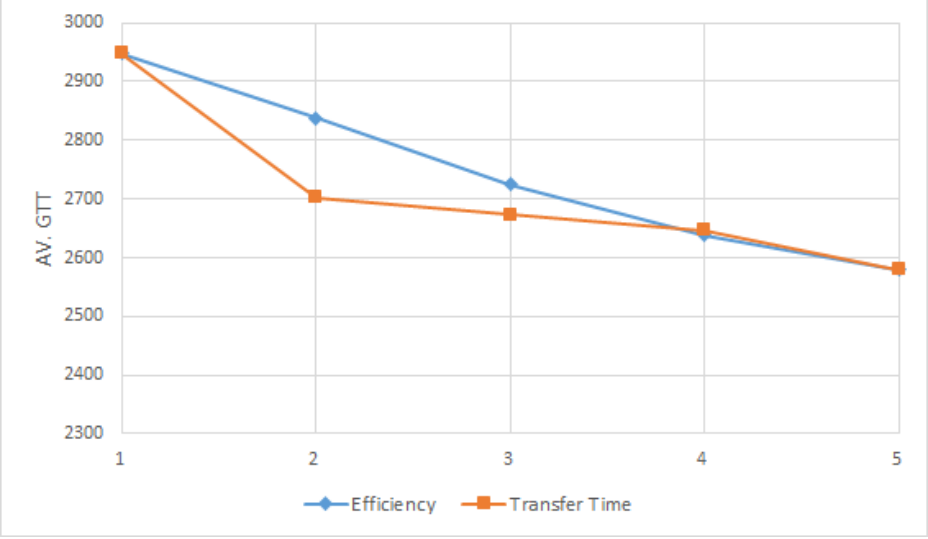


Figure 34 – Average GTT versus different levels of Efficiency (blue) and Transfer time (orange)

Case	Efficiency	Avg. GTT	Case	Transfer time	Avg. GTT
1	Static	2949	1	Static	2949
2	25%	2839	2	30	2703
3	50%	2725	3	15	2674
4	75%	2638	4	10	2646
5	100%	2580	5	0	2580

Table 32 – Results of the 5 cases of Efficiency vs Transfer time

As it can be seen from Table 32, in both cases at the increase of the efficiency or at the decrease of transfer time the system records an improvement. However, the system is more sensible to the workers' efficiency, obtaining lower improvements for levels of efficiency under 50%.

With the same approach used for the routing flexibility part, to carry on with the analysis for the worker's reallocation model it will be considered the parameters chosen for the two experiments of case 4, i.e. an efficiency of 75% and a transfer time of 10 minutes.

An efficiency of 75% is generally high, meaning that a worker when relocated will be almost as efficient as he is in his default machine. Anyway this is justified by the fact that workers can be reallocated only between the two parallel machines. Indeed, these two machines perform similar activities, so it is likely that the efficiency of a worker does not change considerably between one another. The transfer time of 10 minutes, instead, is justified by the fact that two parallel machines, being in the same station, will likely not be placed far from each other, thus allowing a short time for the transfer. These are the reasons why the following analysis will be performed considering a level of efficiency of 75% and a transfer time of 10 minutes as the model parameters.

13.6 Workers reallocation: centralized vs decentralized

The second part of the analysis refers to the study of the effect that the two when rules have on performance. The main difference between the *decentralized* and the *centralized* is that the first one allows a worker to be transferred to another machine only in case his queue is empty, while the latter does not consider this constraint. Indeed, with the *centralized* rule a worker will be transferred at any time, whenever the imbalance between the two queues will have reached a certain threshold.

To make a comparison between the two rules, it has been decided to study their effect on GTT and to compare it with a control case of a static system with no allowed reallocation. For this analysis it has been recorded the performance of the system in terms of GTT, considering the workload norm of 3600, a transfer time of 10 minutes, a permanence time of 30 minutes and an efficiency of 75%. In the graph it is reported the performance of the static case (red part in the right), then the performance of the worker's reallocation where the *decentralized* rule (green part) has been applied and finally the *centralized* rule (blue part). To obtain these data, different thresholds for the activation of the rules have been tried, for example regarding the maximum level of imbalance between the two queues that triggers the relocation of a worker, as explained in the Design of Experiment chapter. This has been done in order to study the level of GTT at the variation of the number of relocations.

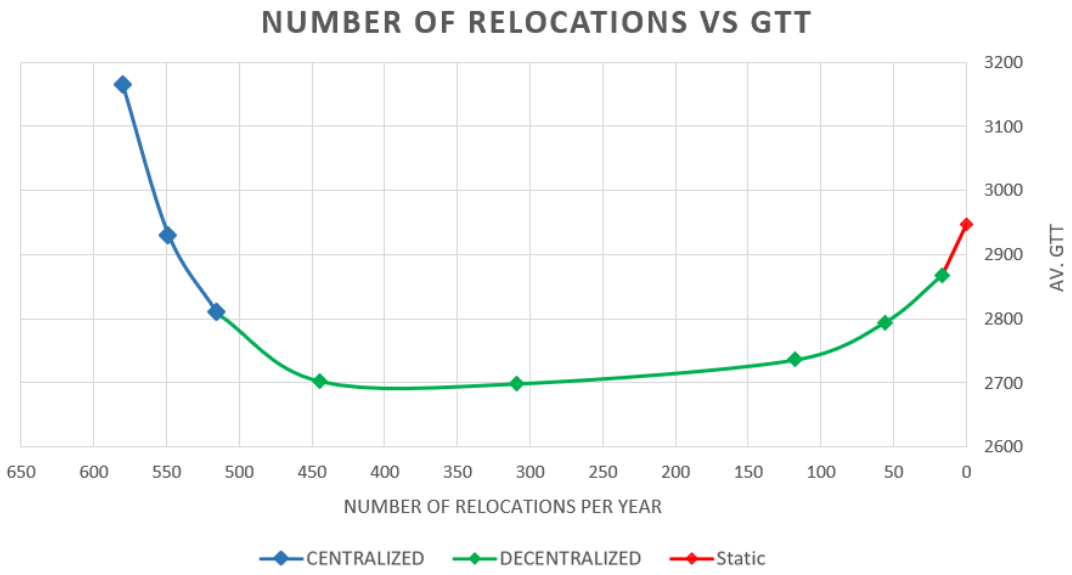


Figure 35 – Number of relocations vs Av. GTT for Centralized, Decentralized and Static rule

As it can be seen from Figure 35, the *decentralized* rule outperforms the *centralized*. Indeed, increasing the total number of relocations brings improvements only up to a certain threshold, reached with the *decentralized* (around 450 relocations per year), above which the performance in terms of GTT starts to worsen. The reason behind this is that every relocation means that the worker needs to spend time transferring to the other machine, and also due to the fact that their efficiency is slightly lower (75%), as they are less productive when relocated. Anyway, reallocating workers means dynamically adjusting capacity to solve bottlenecks, and the best trade-off is obtained with a number of relocations ranging from 300 to 450 per year (which is from 1.30 to 1.96 per day on average).

To conclude, in these first two chapters it has been seen that two parameters, *transfer time* and *efficiency* have an important impact on the system performance. In particular, the model is particularly sensible to the level of efficiency. For reasons of applicability to a real context, it has been decided to consider a level of transfer time of 10 minutes and an efficiency of 75%, instead of considering at their ideal level. The second part of the analysis has been focused on the two different when rules, the *decentralized* and the *centralized.* It has been demonstrated that the first outperforms the latter, and a curve displaying the trade-off between number of relocations and GTT improvement has been drafted. The best trade-off appears to be from 300 to 450 relocations per year, a number which is sufficient to bring significant benefits in the GTT (from 2930 to 2699, -8%) without losing too much productivity of the workers due to the transfer times and the lower efficiency in the relocated machine.

13.7 Workers reallocation: balanced and unbalanced scenario

The last part of the analysis on the workers reallocation refers to test the model both in a *balanced* scenario (in which the bottlenecks to solve come only from the variability of the demand and no intrinsic bottlenecks are present in the system) and in an *unbalanced* scenario (in which the two parallel machines have a different speed). It will be tested the capacity of the worker's reallocation model to solve the queues in order to deliver better time performances (GTT and SFT).

The simulation has been carried out with the norms defined in the methodology section (2200, 2400, 2800, 3600, 4800), and the parameters are set as it has been defined in the last chapter (transfer time 10 minutes, permanence time 30 minutes, efficiency 75%, when rule decentralized). The processing time of the two parallel machines system is 2.1-1.9 (see methodology section for further explanation) which brings to an intrinsic imbalance of 10%.

The results are the following:

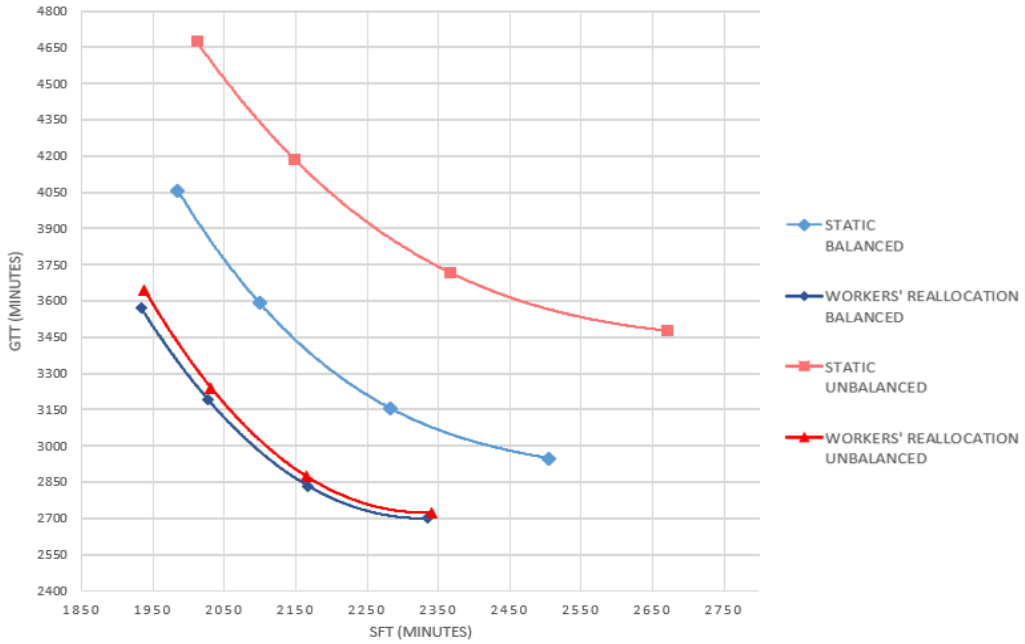


Figure 36 – Lead time performance of workers reallocation vs static in both scenarios

As it can be seen from Figure 36, the unbalanced scenario (2.1-1.9) with static workforce records significantly higher GTT and SFT if compared to the balanced scenario (2-2), see the two curves on the right in the graph. Anyway, the algorithm shows to work well in both cases (balanced and unbalanced), being able to obtain similar results of reduction of GTT and SFT in both the scenarios (the two curves on the bottom left of the graph). This means that, although it has been implemented into the system an intrinsic imbalance of the 10%, the model of workers reallocation has been successful in readjusting capacity to solve the bottlenecks, obtaining a result in the unbalanced scenario similar to the balanced one.

In the following table the results are presented:

Model: Workers reallocation					
Balance level	2 - 2 (balanced scenario)		Permanence time	30 min	
Release algorithm	Workload limiting		Transfer time	10 min	
Routing Flexibility	None		Efficiency	75%	
Workforce	Workload norm	Avg. GTT	Avg. SFT	Δ GTT with static (%)	Δ SFT with static (%)
Static	2200	4058	1984	-	-
	2400	3590	2100		
	2800	3155	2283		
	3600	2949	2505		
	4800	2897	2626		
Reactive	2200	3570	1933	-12.0%	-2.6%
	2400	3191	2027	-11.1%	-3.5%
	2800	2834	2167	-10.2%	-5.1%
	3600	2700	2335	-8.4%	-6.8%
	4800	2675	2419	-7.7%	-7.9%

Table 33 – Complete results of Workers reallocation in the balanced scenario

Model: Workers reallocation					
Balance level	2.1 – 1.9 (unbalanced scenario)		Permanence time	30 min	
Release algorithm	Workload limiting		Transfer time	10 min	
Routing Flexibility	None		Efficiency	75%	
Workforce	Workload norm	Avg. GTT	Avg. SFT	Δ GTT with static (%)	Δ SFT with static (%)
Static	2200	4674	2011	-	-
	2400	4188	2148		
	2800	3716	2367		
	3600	3477	2670		
	4800	3439	2940		
Reactive	2200	3646	1938	-22.0%	-3.6%
	2400	3240	2031	-22.6%	-5.4%
	2800	2874	2165	-22.7%	-8.5%
	3600	2724	2341	-21.7%	-12.3%
	4800	2693	2427	-21.7%	-17.5%

Table 34 – Complete results of Workers reallocation in the unbalanced scenario

The application of the workers reallocation model has brought important improvements for the both balanced and unbalanced case, obtaining in each case similar results in terms of GTT and SFT (see Figure 36). To further investigate the reasons behind this improvement, the following analysis considers the workers' average idleness and the number of reallocations for the four cases studied. This analysis is done considering the results obtained in the before mentioned simulation, considering in particular the workload norm of 3600 (similarly as it has been done for the routing flexibility model).

The worker idleness is the non-productive time spent by a worker during his working hours. It can be accounted as a percentage of his time. The following table reports the percentage of workers' idleness for the cases before mentioned:

Scenario	Model	W1 idleness (%)	W2 idleness (%)	W3A idleness (%)	W3B idleness (%)	W4 idleness (%)	W5 idleness (%)
Balanced	Static	6.6%	6.9%	**7.1%**	**7.0%**	6.8%	6.6%
Unbalanced	Static	6.6%	6.5%	**2.4%**	**11.5%**	6.8%	7.0%
Balanced	Workers reallocation	6.5%	6.5%	**5.7%**	**5.6%**	6.8%	7.0%
Unbalanced	Workers reallocation	6.5%	6.5%	**4.8%**	**6.4%**	6.8%	6.9%

Table 35 – Workers idleness in the balanced vs unbalanced scenario

As it can be seen from Table 35, for the two *balanced* scenario, passing from static to workers reallocation the idleness of the workers of the two parallel machines (3A and 3B) decreases (from around 7.1% to 5.7%). Indeed, the system has become more efficient in solving bottlenecks that are created between the parallel machines, and it has increased the two workers' saturation.

In the *unbalanced* scenario, instead, the initial idleness of worker 3A and 3B are very different (respectively 2.4% and 11.5%). This is due to the fact that machine 3A is slower, hence for most of the time it is occupied solving the queues that are generated. The intrinsic bottleneck is thus represented by machine 3A. The idleness of the two workers becomes more balanced once the workers reallocation is activated, obtaining respectively 4.8% and 6.4%.

13.8 Routing flexibility vs workers reallocation: a comparison

The last research question regards the comparison between the two models. For this last analysis, the two models will be considered with the parameters determined in the previous analyses, this means the routing flexibility with a 20% level of interchangeability, and the workers reallocation with a transfer time of 10 minutes, permanence time of 30 minutes and an efficiency of 75%.

Models	Pre-set parameters			Experimental parameters
	Inter-changeability	Transfer time	Efficiency	Balance level
Static	-	-	-	2-2 (balanced) 2.1-1.9 (unbalanced)
Workers reallocation	-	10	75%	2-2 (balanced) 2.1-1.9 (unbalanced)
Routing flexibility	20%	-	-	2-2 (balanced) 2.1-1.9 (unbalanced)
Performance studied				
Gross Throughput Time (GTT)				
Shop Floor Throughput Time (SFT)				
Number of relocations per year				
Workers' saturation				

Table 36 – Parameters and levels for the comparison of the two models

The goal of this analysis is to compare the two models and to test their performances both in the balanced and in the unbalanced scenario. The first comparison regards the performances obtained in terms of GTT and SFT. Then, the idleness of the workers will be studied and finally the number of relocations.

The results in terms of GTT and SFT are the followings:

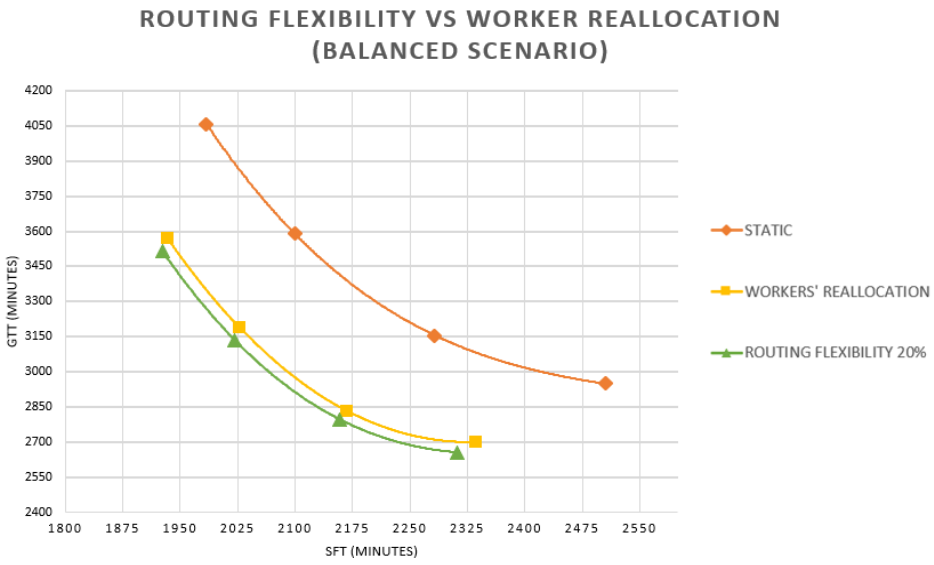


Figure 37 – Lead time performance of workers reallocation vs routing flexibility

As it can be seen from the graph, in both cases the system is able to solve bottlenecks and to improve considerably both GTT and SFT. The routing flexibility model slightly outperforms the workers reallocation model, however their difference is very low compared to the total improvement. The marginal percentage improvement for low and high norms (norm of 2400 and 3600) is reported in the next table:

MARGINAL IMPROVEMENTS (LOW NORM)		
MODEL	**Avg. GTT**	**Avg. SFT**
STATIC	0%	0%
WR	-11.1%	-3.5%
RF 20%	-12.7%	-3.7%

MARGINAL IMPROVEMENTS (HIGH NORM)		
MODEL	**Avg. GTT**	**Avg. SFT**
STATIC	0%	0%
WR	-8.4%	-6.8%
RF 20%	-9.9%	-7.7%

Table 37 – Marginal improvements of low and high norms for the two models

As it can be seen from table, for low norms the improvement in GTT is wider than in SFT, while for high norms the two improvements are more balanced. For low norms, indeed, by applying the models the system can release more orders into the system, reducing the GTT, and bottlenecks are less frequent. With high norms, instead, the two models allow the system not only to release more orders (reducing GTT), but also to consistently solve queues, leading also to a shorter SFT. The two models, as it can be seen also from the Figure 36, appear to work better with high norms, obtaining a better tradeoff between GTT and SFT.

By comparing the two models' efficacy in the unbalanced scenario, the results are the following:

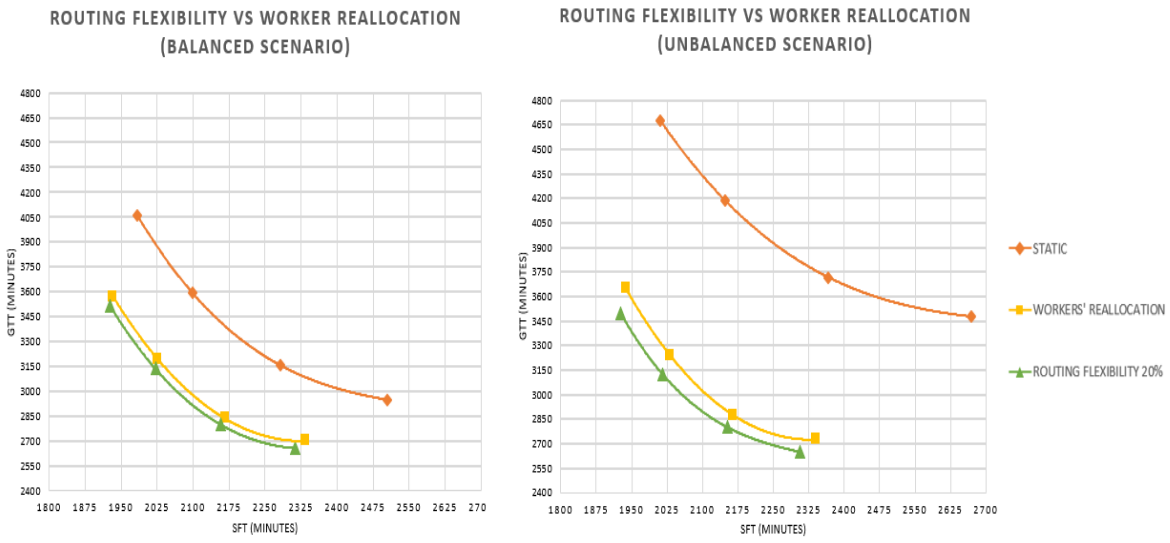


Figure 38 – Lead time performance of the models in the balanced and unbalanced scenarios

The two models appear to be both robust to solve intrinsic imbalances, and they obtain similar results in the unbalanced case. In this case, however, the routing flexibility algorithm slightly outperforms the workers reallocation algorithm.

One peculiar characteristic of the system is that by applying the model in the balanced and the unbalanced scenario, the total number of reallocations does not vary significantly between the two scenarios, as it can be seen from Figure 39.

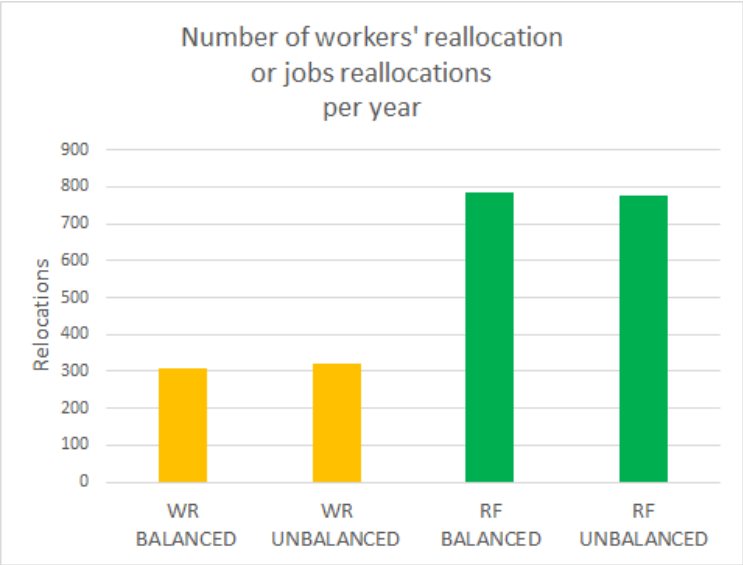


Figure 39 – Comparison of reallocations for the two models

The system is indeed able to adapt to the changes regarding the intrinsic imbalances (parallel machine 3B being 10% faster than machine 3A) without changing considerably the total number of relocations that are done to balance the system. The other significant conclusion that can be drafted from data reported in Figure 39, is that the routing flexibility algorithm needs to carry out almost three times the number of relocations of the other model to deliver similar results (around 800 relocations a year against 300). Anyway, it must be considered that the relocations of the two models are of different nature: in the WR, reallocating means transferring workers from one station to another. This indeed requires a worker to collect his tools, physically move to the other station and start helping the other worker. This dynamic change in capacity requires of course to spend time transferring to the other station and implies that the worker will have a lower efficiency in the other machine. For the RF model, instead, changing the routing of a job should be instead much simpler. Indeed, as the change occurs while the job is still queueing in the previous machine (see methodology section for complete description), it means that the job simply needs to change its routing indications. Which may be done in the Kanban (physically or digitally) or in the production

planning system. Indeed, after being processed by prior to the parallel station (machine 2), the job would be immediately directed towards its new destination, without requiring any additional movement or change from one queue to another. In conclusion, the number of relocations in the RF model are almost three times the ones in the WR model, but due to their different nature they are also way simpler to carry out and they allow a smooth redistribution of the workload between the stations.

In Table 38, it is reported the distribution of the relocations in the balanced and in the unbalanced scenario. As it can be seen, the machine that in the unbalanced scenario is the slowest one (3A) is also the one the requires the highest number of relocations. Anyway, the total number of relocations does not change considerably between the balanced and the unbalanced scenarios.

Model	Scenario	M3A	M3B	TOTAL
Workers' reallocation	Balanced	155	154	309
	Unbalanced	213	106	319
Routing flexibility	Balanced	390	395	785
	Unbalanced	442	336	777

Table 38 – Number of jobs or workers reallocated in the two models (data refers to one year)

Finally, the last part of the analysis considers the workers' idleness. In the left column, the results in the balanced case are presented, while in the right column the unbalanced.

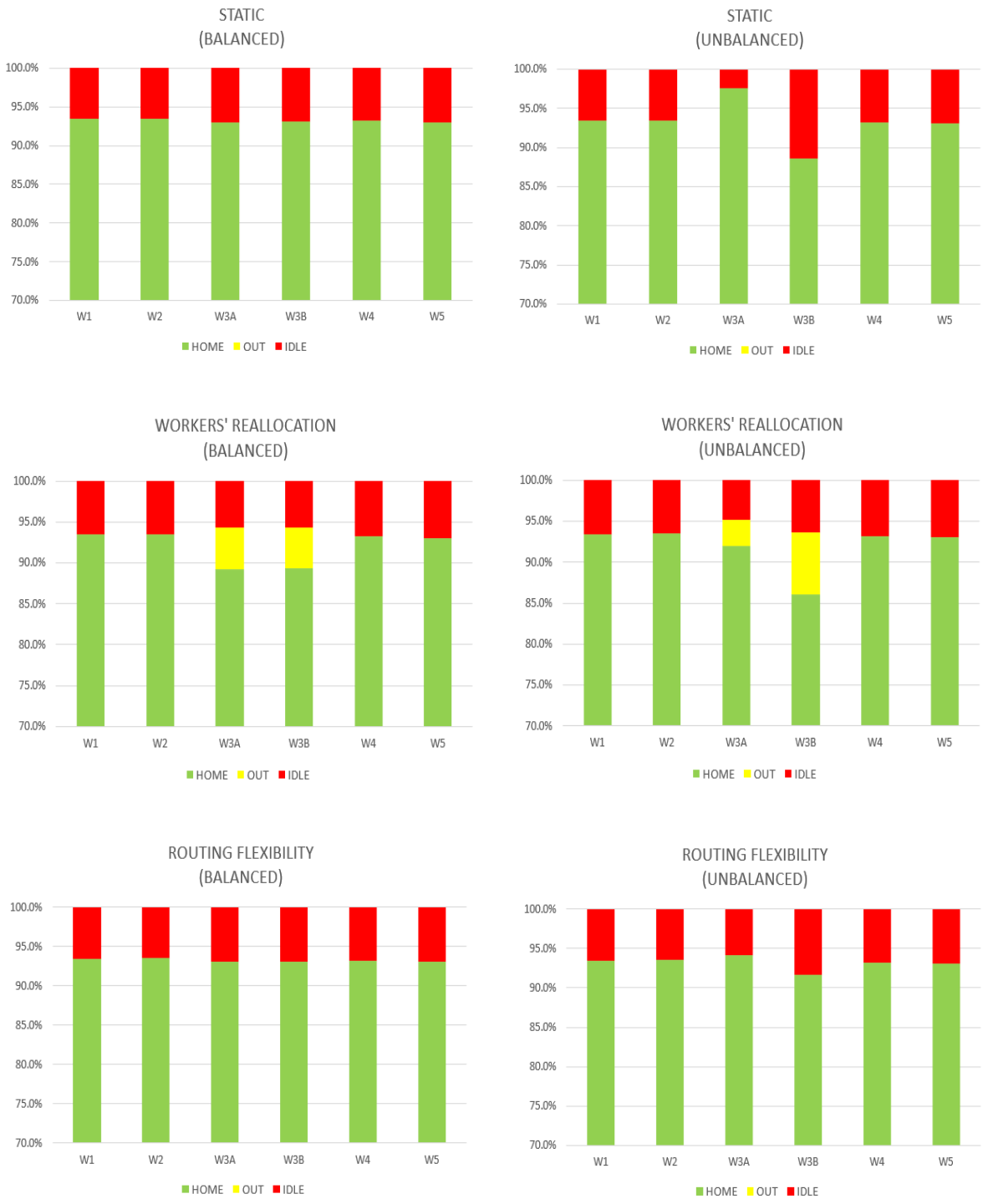


Figure 40 – Comparison of workers saturation for the two models

Comparing the two scenarios, it can be seen how the different models respond to the imbalance created regarding the workers' time. The *static* model, of course, has no means to defend itself from the imbalance, and the result is that the worker 3A (the one working in the slower parallel machine) is saturated for the 97.6% of his time, while the worker 3B in the faster machine for the 88.5%

(graph at top right). The *workers reallocation* model solves this unbalance by transferring in the other machine the worker 3A for the 3.2% of his time, and worker 3B for 7.5% (time out, in yellow). This difference enables to solve the intrinsic imbalance. The *routing flexibility* model, instead, is successful in solving the unbalance in workers' idleness thanks to the reallocation of jobs between the two machines. The complete results are reported in the next table:

Model	Scenario	Time	W1	W2	W3A	W3B	W4	W5
Static	Balanced	Home	93.4%	93.5%	92.9%	93.1%	93.2%	93.0%
		Out	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
		Idle	6.6%	6.5%	7.1%	6.9%	6.8%	7.0%
	Unbalanced	Home	93.4%	93.5%	97.6%	88.5%	93.2%	93.0%
		Out	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
		Idle	6.6%	6.5%	2.4%	11.5%	6.8%	7.0%
WR	Balanced	Home	93.5%	93.5%	89.2%	89.4%	93.2%	93.0%
		Out	0.0%	0.0%	5.1%	5.0%	0.0%	0.0%
		Total (H+O)	93.5%	93.5%	94.3%	94.4%	93.2%	93.0%
		Idle	6.5%	6.5%	5.7%	5.6%	6.8%	7.0%
	Unbalanced	Home	93.5%	93.5%	91.9%	86.1%	93.2%	93.1%
		Out	0.0%	0.0%	3.2%	7.5%	0.0%	0.0%
		Total (H+O)	93.5%	93.5%	95.1%	93.6%	93.2%	93.1%
		Idle	6.5%	6.5%	4.8%	6.4%	6.8%	6.9%
RF	Balanced	Home	93.5%	93.5%	93.0%	93.1%	93.2%	93.1%
		Out	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
		Idle	6.5%	6.5%	7.0%	6.9%	6.8%	6.9%
	Unbalanced	Home	93.5%	93.5%	94.2%	91.6%	93.2%	93.1%
		Out	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
		Idle	6.5%	6.5%	5.8%	8.4%	6.8%	6.9%

Table 39 – Complete results of workers time for the two models in balanced vs unbalanced case

To conclude, in this chapter the models of routing flexibility and workers reallocation have been compared with respect to the balanced and the unbalanced scenario. The first aspect analyzed was the reduction in GTT and SFT. Both models are successful in improving the system performance, obtaining GTT and SFT reductions between 7 and 10% for high norms. Analyzing the unbalanced scenario, the models appeared both to be robust against the intrinsic imbalances created. In both cases the two models have been able to solve the bottlenecks and to deliver results similar to the

ones obtained in the balanced case. Anyway in the unbalanced the routing flexibility slightly outperformed the workers reallocation.

By studying more in deep the two models, it has been reported how the workers reallocation model required nearly 300 relocations per year, while the routing flexibility around 800. However, the different nature of these relocations has been explained: in one case it means a worker changing his station, walking to the other station and helping the other, while in the other case it is simply a change in routing before the job enters one queue or the other. Thus, the higher number of relocations in the routing flexibility model are so explained, and they do not appear to bring important issues from the organizational point of view. The surprising result, instead, is that in the unbalanced scenarios the total number of relocations for both the two models did not change compared to their balanced case, although the distribution of the relocations was moved more towards the slower machine (3A).

Finally, the workers' idleness has been studied, and it has been demonstrated how the two models different cope with the imbalances in the system, being both successful in reducing the difference in idleness between the workers of the two parallel stations.

14. Conclusion

The most important field of application of the Workload Control system is the high-variety and low-volume manufacturing environment typical of MTO firms. By controlling the release of jobs to the shop floor, Workload Control aims to obtain a smooth production flow to avoid shop congestion and excessive queues. In a Workload Control system, jobs are held in a Pre Shop Pool (PSP) before being released to the shop floor. The Order Review and Release (ORR) algorithm regulates this decision, trying to obtain a system in which from one side jobs are not held too much time in the PSP, and from the other side no significant congestion is created. This first method of workload control is called input control. Out of these, one of the most important input control method studied in literature is the Workload Limiting.

- *Workload limiting algorithm*: jobs are released in the shop floor until a pre-set norm (level of workload) is not violated. The norms serve as an upper bound, in order to avoid the creation of shop congestion.

Anyway, the variability in a MTO system is generally quite consistent, a precise prediction on the machines' saturation and station imbalances is rather difficult to achieve, and frequent bottlenecks can occur, limiting the system capacity to deliver jobs in short time. To face this, another important kind of leverage can be used to dynamically balance the production flow: the output control. This different kind of control is used to adjust production capacity to the current necessities of the system, or to redirect the flow between the shop floor stations. This method it has been widely proven in literature to be very effective in balancing the production flow, helping to solve bottlenecks and consequently improving the time performance of the system. Two different kinds of output control are considered:

- *Routing flexibility*: when it is possible to move jobs between one station and another, an algorithm decides the routing of the interchangeable jobs, with the aim to obtain a smoother production flow to avoid the creation of bottlenecks.

- *Workers reallocation*: workers can be transferred between one station and another, to help each other to process jobs whenever a strong imbalance between the stations has occurred.

The aim of this thesis has been to study the different application of the two output control methods, the *routing flexibility* and the *workers reallocation* in a flow-shop with parallel machines. In literature, few studies on workload control have concentrated in configurations with parallel machines, and in only one case (Bokhorst, Slomp &, Gaalman; 2006) the methods of routing flexibility and workers reallocation have been compared in such a configuration. In this thesis it has been built a model of flow-shop with a total of six machines with two working in parallel (five stations, six machines) to study the performance of the two output control models in such a configuration.

One issue often found in literature is that generally in the models many simplifications and assumptions are done, some of which place the constraints for the replicability of the studies in real manufacturing contexts. As stated by Miragliotta and Perona (2010), the lack of considerations of parallel machines in the shop configurations often done by researchers is one of these reasons. A further reason is that the model simulated with parallel machines in literature generally assumes that the latter have the same processing time. As one objective of the thesis is to create and study a system which is as close as possible to reality, it has been introduced in the model the possibility to create an intrinsic imbalance between the two parallel machines, considered as a percentage difference of their respective processing time. Systems with intrinsic imbalances are indeed often found in real applications. For example, in a parallel shop configuration, an intrinsic imbalance can be simply due to the existence of new and old machines working on the same production flow. The lack of consideration of both parallel machines in a shop, and the simplification of using the same processing time for all the stations are some of the reasons stated by Miragliotta and Perona (2010) for which the simulated models are often far from real scenarios. Then, to the best of the authors knowledge, no research has studied the application of these two output control algorithms in a system with parallel machines that have different processing times.

To summarize, it has been studied the performance of the two output control models, the *routing flexibility* and the *workers reallocation* in a flow shop with parallel machines. Then the two models have been tested in a scenario containing intrinsic imbalances. The main objective indeed has been to compare the model of routing flexibility and the of workers reallocation in a system that is affected not only by a high variability of demand, typical of MTO companies, but that also entails intrinsic imbalances which are typical of real shops applications.

14.1 Research question 1

- What is the contribution of the routing flexibility to performances and how these are affected by the level of interchangeability? What is the minimum level of interchangeability that leads to most of the result?

The results regarding the first part of this question have been obtained testing the routing flexibility model with different levels of interchangeability (0%, 5%, 10%, 20%, 50%, 100%). For example, 20% of interchangeability means that the routing flexibility can be applied to the 20% of jobs in the system. The results have been studied in terms of average gross throughput time (GTT) and average shop floor throughput time (SFT) of the jobs worked by the system. The first conclusion is that, as expected, increasing the level of interchangeability leads to an improvement in the results. Anyway, what is interesting is that the marginal improvement rapidly decreases. Indeed, with the 20% of interchangeability, the system is already able to obtain the 79.0% and the 75.6% of the total improvement, respectively in GTT and SFT, that it would achieve with an interchangeability of 100%. This entails that most of the benefits brought by the application of routing flexibility is obtainable with an interchangeability of 20%, and the further increases in this level bring only to limited marginal benefits.

The improvement for high norms in case of 20% of interchangeability is of -9.9% in GTT and -7.7% in SFT. The reason for the GTT improvement is due to the fact that, thanks to the reallocation of jobs between the two queues of the parallel machines, the occurrence of bottlenecks (queues with high amounts of workload) is reduced. Whenever a queue gets too long, the input control algorithm does not allow the release of new jobs into the system, as the workload norm would be violated. This makes the algorithm hold jobs for long time in the PSP, thus increasing their average GTT. As a matter of fact, when the system is static (no routing flexibility is applied), the only way for the system is to wait that queues are solved by themselves. The algorithm of routing flexibility, instead, rebalances the two queues, changing the routing of the jobs to the less saturated queue. In this way, queues are solved faster. The better balancing leads also to jobs spending less time queueing, which explains the other reduction obtained in the SFT.

To investigate the results obtained, the distribution and length of the queues before the two parallel machines have been studied. It has been noticed that, thanks to the routing flexibility, the system has actually become more efficient in decreasing the queues' length. The results obtained in this analysis confirm this statement, as the average queue length of the two parallel machines (3A and 3B) has passed from 14.0 minutes to 6.85 minutes. Anyway, this decrease has also brought to a higher stability, as the average standard deviation of the lengths of the two queues has passed from 11.2 to 5.5 minutes. Thus, the system has obtained both shorter (lower average) and stable (lower variability) queues. Finally, to understand how much the two queues have become also more similar, the standard deviation of the difference of their length has been studied. As their difference has been reduced from 17.2 to 5.5 minutes, it proves that the two queues have become more similar, leading to the objective of obtaining a more balanced production flow.

Finally, it has been decided to study the performance of the routing flexibility model in a scenario with an imbalance of 10% between the two parallel machines. This means that one machine was 10% faster than the other, while the average of the two together was the same as the other machines. This was called the unbalanced scenario. In this case, the algorithm has brought results that were close to the balanced scenario, being able to successfully solve the intrinsic imbalances created. In the unbalanced scenario, already the 10% of interchangeability was sufficient to reach around the 80% of the total benefits achievable with the complete interchangeability. In this latter scenario, the GTT has improved by -21.6%, passing from 3477 to 2725 minutes, and the SFT by -11.5%, passing from 2670 to 2362 minutes with respect to the static case - the static case in the unbalanced had worse performance than the static case in the balanced. The conclusion is that, the more the system is intrinsically unbalanced, the more the routing flexibility has effect on performances.

The most important result is that with the 10-20% of interchangeability the routing flexibility can already obtain most of the results, making less marginally efficient investing in higher percentages of interchangeability in the system.

14.2 Research question 2

- What is the contribution of workers reallocation to performances and how these are affected by the efficiency of the workers, the permanence time and the transfer time between machines? Which when rule (decentralized, centralized) leads to most of the benefits in a parallel flow-shop configuration?

The first part in the study of the model of workers reallocation was to conduct an ANOVA analysis to assess whether the three main parameters of this model had a significant effect on the result (the performance studied was the GTT). The parameters considered have been the *efficiency* of the workers, the *transfer time* and the *permanence time.* Results from the ANOVA indicate that *efficiency* and *transfer time* both have a relevant impact on the GTT, being the model highly sensible to the level of these two parameters, while the *permanence time* was instead less significant.

After further investigating the effect of *transfer time* and *efficiency*, for reasons of applicability to a real context it has been decided to consider a level of transfer time of 10 minutes and an efficiency of 75%. These levels have been chosen instead of their ideal levels (which would have been 0 minutes of transfer time and 100% of efficiency) because the latter would have probably led to results not strictly applicable in real industrial contexts. A transfer time of 10 minutes is justified by the fact that parallel machines in the same station will likely be placed not far from each other. Whilst the efficiency of 75% that one worker would have in the other machine when reallocated is justified by the fact that two parallel machines usually perform similar activities, hence the worker when reallocated to the other machine would probably be almost as effective as he is in his default machine.

The second part of the analysis has been focused on the performance of the model when two different when rules were applied: the *decentralized* and the *centralized.* A practical difference between these two rules is that the centralized allows workers to be transferred more times to the other machines than the other. This means that the total number of relocations will be higher when the centralized is applied. To evaluate the performance of these two rules, a curve displaying the trade-off between number of relocations and GTT improvement has been built. The result demonstrates that the decentralized rule outperforms the centralized, and that the best tradeoff is between the 300 and 450 total relocations per year (1.3 and 1.96 per day). The benefits obtained in GTT with such a number of relocations were significant, decreasing it by -7.9% from 2930 to 2699 minutes (the curve flattens around this number of relocations, obtaining similar performance of GTT). The conclusion that can be drafted is that the centralized rule is outperformed by the decentralized rule, which delivers better performance in terms of GTT and will hence taken as reference for the remaining part of the study.

The third and last part studied the performance of the model in the balanced and unbalanced scenario. The first notable fact is that in the static case (when the model is not applied), the unbalanced scenario yields highly worse results, with the GTT going to 3477 from 2949 minutes (+17.9%) compared to the static balanced case, and the SFT going to 2670 from 2505 minutes

(+6.6%). Thus, there is a higher presence of bottlenecks, which is mostly due to the intrinsic imbalance created in the system. In this case, the workers reallocation model has proven to be successful in solving these bottlenecks, obtaining similar results in both the two scenarios (balanced and unbalanced). Indeed, in the balanced case it records a -8.4% reduction of GTT, passing from 2949 to 2700 minutes. While, in the unbalanced scenario, the reduction of GTT is -21.7%, from 3477 to 2727 minutes. It is worth noting that the results in GTT obtained in the two scenarios (2700 and 2727 minutes) are very close, although they started from very different performances in the static case (2949 and 3477 minutes). The results in terms of SFT are similar, passing from 2505 to 2335 minutes in the balanced case (-6.8%) and from 2670 to 2341 minutes (-12.3%) for the unbalanced one. Therefore, it can be concluded that the workers reallocation model is generally robust against the intrinsic imbalances, being able to successfully solve them and to obtain consistent reduction of both GTT and SFT.

14.3 Research question 3

- What is the respective contribution of the two methods (routing flexibility and workers reallocation) to performances, to the variation of system parameters such as stations' imbalance?

The last research question regards the comparison between the two models. In particular, it has been considered the data obtained to answer the two previous research questions, and the performances achieved by the two models have been compared, considering both the balanced and the unbalanced case.

The two models have obtained similar improvements in the balanced case: for high norms, the routing flexibility improves by -9.9% the GTT and by -7.7% the SFT compared to the static case, while the workers reallocation improves by -8.4% and -6.8% respectively the GTT and SFT. In the unbalanced case, also, the routing flexibility model (with 20% interchangeability) slightly outperforms the other model, reducing by -23.8% and -13.6% against a reduction of -21.7% and -12.3% of the workers reallocation model. However, on the whole, the difference between the two models is so small that it might be due more to the choice of the levels of the parameters, than to an effective better capability of one model compared the other.

The most important consideration refer to the number of relocations of the two models. Comparing this measurement, the routing flexibility has obtained the results mentioned in GTT and SFT by reallocating nearly 800 jobs a year (this means that 800 times one job's routing has been changed). While the workers reallocation obtained similar results in terms of GTT and SFT, but just by reallocating workers around 300 times a year. Anyway, this difference in quantity of relocations (which is 3.5 relocations a day for the first model against 1.3 for the second) should not generate a problem for the routing flexibility model, as this kind of transfers are generally easier to manage than the ones in the workers reallocation model. Indeed, when the routing of a job is changed, in the model this occurs while the job is still queueing in the machine before. Hence, changing the routing could mean simply modifying the Kanban of the job, or to record the change in the production planning and control (PPC) software. From the other side, instead, transferring a worker from one machine to the other requires of course alerting the worker in time to allow him to prepare, and then there is the time needed for the worker to physically move to the other station. To conclude, the higher number of relocations in the routing flexibility model would likely not bring important issues, as from the organizational point of view they should be simpler to manage.

The other important fact discovered in this analysis is that, in case of important intrinsic imbalances, the two models achieve results similar to the balanced case, without varying much of the total number of relocations. The latter pass from 309 to 319 for the workers reallocation model and they even decrease from 785 to 777 for the routing flexibility model. What changes in the unbalanced case is only the percentage of relocations requested by the machine 3A (the slowest one) compared to the other, which passes for worker reallocation from 50.1% to 66.8% and for the routing flexibility from 49.7% to 56.9% of the total relocations. The conclusion that can be drafted is that the two models, when the system is highly unbalanced, are indeed more efficient in solving the bottlenecks, starting from lower performances of the static case and being able to obtain similar results to the balanced scenario with the same number of relocations.

14.4 Managerial implications

Regarding routing flexibility, the first indication that can be useful for practitioners is that an interchangeability between the 10 and the 20 percent has proven to be sufficient to bring most of the benefits in the application of this model. Indeed, investing to increase this level above 20 percent would only bring to marginal improvements. This can be important as the desired level of

interchangeability could be a driver in the decision on which machine to buy or on how to program the production line. Hence, the suggested trade-off is around 20 percent, which should be also sufficient to reach a stability in the system that allows to obtain a smooth and balanced production flow, as demonstrated in the thesis.

The second indication regards the fact that, the more unbalanced the system is, the more efficient is the model of routing flexibility. Indeed, the number of relocations (changes in routing) was similar for both the balanced and the unbalanced case, although in the second case the percentage improvement has been considerably higher compared to the respective static case. In other words, applying routing flexibility would bring the strongest benefits in an unbalanced line.

Regarding the workers reallocation model, the transfer time and the efficiency were proven to be the most important parameters. Indeed, when in a real context the workers reallocation is considered, these two parameters should carefully be evaluated. A low efficiency in the other parallel machine or a high transfer time due to the distance of the parallel machines may strongly inhibit the benefits brought by reallocating workers.

Another indication is that, in order not to make the system too nervous and to better exploit the trade-off, the thesis suggests that is better to restrain the transfer of workers to maximum 1-1.5 changes per day. This was sufficient in the model to obtain important improvements in both the gross throughput time and the shop floor throughput time, without making the system too nervous. The proper number of relocations would probably vary significantly depending on the day and on the configuration of the system, but the maximum threshold of 1.5 per day may still be valid.

Finally, both models have proven to be robust against a scenario containing strong imbalances, and hence they both have demonstrated to be more efficient in that case. Generally, the routing flexibility has brought to important results, and it is expected not to bring too many managerial complications. Anyway, although the choice between the two models needs to be done regarding the configuration of the system (machines placed far from or near each other and possibility to split the workload by two workers or to effectively change the routing), the thesis has demonstrated that, in case these parameters are respected, both the models can bring to significant improvements both in GTT and SFT balancing the workload in a flow-shop with parallel machines.

14.5 Limitations and future research

A limit that strongly affects most of simulation-based studies is the strong dependence of the results on the levels chosen for the parameters. Indeed, the configuration of the model and the parameters impact significantly on the conclusions drafted from workload control studies.

To face this limit, it has been tried to keep the model as neutral and simple as possible, for example considering a simple configuration with only one station with parallel machines, and studying in deep the proper level of the parameters to set, so that they could be considered as much realistic as possible. This has been done in order to be able to study the single effect on the model of the different parameters and the assessment of the performance obtained in the different scenarios. Anyway, it is inevitable that, to a certain degree, the conclusions depend on the levels chosen of the parameter.

However, the general indications obtained in the thesis seem to be feasible, like for example that a low percentage of interchangeability is sufficient to deliver most of the benefits, that the workers reallocation should not exceed a certain threshold, or finally that the efficiency of these two models increases the more unbalanced the system is. Regarding the different percentages of improvements, instead, the result may be significantly affected by the choice of the level of the parameters.

The thesis sets the basis and provides indications on how the model of a system with parallel machines can be created with the simulations and studied, applying to it different methods of output control. A future research could increase the level of complexity of the model, considering for example more stations with parallel machines or more than two machines in parallel in a station. This would open up to the possibility to study more deeply the interactions and possible synergies between different parallel stations or parallel machines in a station. It is suggested also to experiment new algorithms both of routing flexibility and workers reallocation and to replicate the model not under a workload limiting but under other workload control algorithms. Finally, this thesis has studied the effect of routing flexibility and workers reallocation in a flow-shop configuration. A future study could extend also the analysis to other configurations such as job shops or to hybrid solutions.

References

Baykasoğlu, A., & Göçken, M. (2011). A simulation based approach to analyse the effects of job release on the performance of a multi-stage job-shop with processing flexibility. *International Journal of Production Research, 49(*2), 585-610.

Bechte, W. (1988). Theory and practice of load-oriented manufacturing control. *The International Journal of Production Research, 26(*3), 375-395.

Bergamaschi, D., Cigolini, R., Perona, M., & Portioli, A. (1997). Order review and release strategies in a job shop environment: A review and a classification. *International Journal of Production Research, 35*(2), 399-420.

Bertolini, M., Romagnoli, G., & Zamora, F. (2015). Assessing performance of Workload Control in High Variety Low Volumes MTO job shops: A simulative analysis. In 2015 *International Conference on Industrial Engineering and Systems Management (IESM)* (pp. 362-370). IEEE.

Bertrand, J. M., & Van Ooijen, H. P. G. (2002). Workload based order release and productivity: a missing link. *Production Planning & Control, 13(*7), 665-678.

Bokhorst*, J. A. C., Slomp, J., & Gaalman, G. J. C. (2004). On the who-rule in Dual Resource Constrained (DRC) manufacturing systems. *International Journal of Production Research, 42*(23), 5049-5074.

Bokhorst, J. A., & Gaalman, G. J. (2009). Cross-training workers in Dual Resource Constrained systems with heterogeneous processing times. *International Journal of Production Research, 47(*22), 6333-6356.

Breithaupt, J. W., Land, M., & Nyhuis, P. (2002). The workload control concept: theory and practical extensions of Load Oriented Order Release. *Production Planning & Control, 13*(7), 625-638.

Campbell, G. M. (1999). Cross-utilization of workers whose capabilities differ. *Management Science, 45(*5), 722-732.

Chan, F. T. (2001). The effects of routing flexibility on a flexible manufacturing system. *International Journal of Computer Integrated Manufacturing, 14*(5), 431-445.

Davis, D. J., Kher, H. V., & Wagner, B. J. (2009). Influence of workload imbalances on the need for worker flexibility. *Computers & Industrial Engineering, 57*(1), 319-329.

Felan, J. T., & Fry, T. D. (2001). Multi-level heterogeneous worker flexibility in a Dual Resource Constrained (DRC) job-shop. *International Journal of Production Research, 39*(14), 3041-3059.

Fernandes, N. O., Thürer, M., Silva, C., & Carmo-Silva, S. (2017). Improving workload control order release: Incorporating a starvation avoidance trigger into continuous release. *International Journal of Production Economics, 194*, 181-189.

Fernandes, N. O., & Carmo-Silva, S. (2011). Workload control under continuous order release. *International Journal of Production Economics, 131*(1), 257-262.

Fredendall, L. D., Ojha, D., & Patterson, J. W. (2010). Concerning the theory of workload control. *European Journal of Operational Research, 201*(1), 99-111.

Fruggiero, F., Fera, M., Iannone, R., & Lambiase, A. (2015). Work control in balanced DRC systems supported by negotiation procedures between autonomous agents. *IFAC-PapersOnLine, 48*(3), 733-740.

Fry, T. D., Kher, H. V., & Malhotra, M. K. (1995). Managing worker flexibility and attrition in dual resource constrained job shops. *International Journal of Production Research, 33*(8), 2163-2179.

Hendry, L. C., & Wong, S. K. (1994). Alternative order release mechanisms: a comparison by simulation. *The International Journal of Production Research, 32*(12), 2827-2842.

Henrich, P., Land, M. J., & Gaalman, G. J. C. (2007). Semi-interchangeable machines: implications for workload control. *Production Planning and Control, 18*(2), 91-104.

Henrich, P., Land, M., & Gaalman, G. (2006). Grouping machines for effective workload control. *International Journal of Production Economics, 104*(1), 125-142.

Hottenstein, M., & Bowman, S. (1998) Cross-training and worker flexibility: a review of DRC system research, *The Journal of High Technology Management Research*, Volume 9, Number 2, pages 157-174

Huang, Y. (2017). Information architecture for effective Workload Control: an insight from a successful implementation. *Production Planning & Control*, *28*(5), 351-366.

Kher, H. V., & Malhotra, M. K. (1994). Acquiring and operationalizing worker flexibility in dual resource constrained job shops with worker transfer delays and learning losses. *Omega*, *22*(5), 521-533.

Kingsman, B. G., Tatsiopoulos, I. P., & Hendry, L. C. (1989). A structural methodology for managing manufacturing lead times in make-to-order companies. *European journal of operational research*, *40*(2), 196-209.

Kingsman, B., & Hendry, L. (2002). The relative contributions of input and output controls on the performance of a workload control system in make-to-order companies. *Production Planning & Control*, *13*(7), 579-590.

Košťál, P., Velíšek K. (2011). Flexible manufacturing system In: World Academy of Science, Engineering and Technology 77 ISSN 2010-376X. pp 825-829

Land, M., & Gaalman, G. (1996). Workload control concepts in job shops a critical assessment. *International journal of production economics*, *46*, 535-548.

Land, M., Stevenson, M., & Thürer, M. (2014). Integrating load-based order release and priority dispatching. *International Journal of Production Research*, *52*(4), 1059-1073.

Land, M., & Gaalman, G. (1996). Workload control concepts in job shops a critical assessment. *International journal of production economics*, *46*, 535-548.

Land, M. (2006). Parameters and sensitivity in workload control. *International journal of production economics*, *104*(2), 625-638.

Land, M. J., Stevenson, M., Thürer, M., & Gaalman, G. J. (2015). Job shop control: In search of the key to delivery improvements. *International Journal of Production Economics*, *168*, 257-266.

Land, M., Stevenson, M., & Thürer, M. (2013). Integrating load-based order release and priority dispatching. *International Journal of Production Research, 52* (4), 1059–1073. doi:10.1080/00207543.2013.836614

Ma, Y., Chu, C., & Zuo, C. (2010). A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering, 58*(2), 199-211.

Małachowski, B., & Korytkowski, P. (2016). Competence-based performance model of multi-skilled workers. *Computers & Industrial Engineering, 91,* 165-177.

Malhotra, M. K., Fry, T. D., Kher, H. V., & Donohue, J. M. (1993). The impact of learning and labor attrition on worker flexibility in dual resource constrained job shops. *Decision Sciences, 24*(3), 641-664.

McCreery, J. K., & Krajewski, L. J. (1999). Improving performance using workforce flexibility in an assembly environment with learning and forgetting effects. *International Journal of Production Research, 37*(9), 2031-2058.

Melnyk, S. A., & Ragatz, G. L. (1989). Order review/release: research issues and perspectives. *The International Journal Of Production Research, 27*(7), 1081-1096.

Miragliotta, G., & Perona, M. (2000). Workload control: A comparison of theoretical and practical issues through a survey in field. *In Eleventh international working seminar on production economics* (pp. 235-248).

Mitzner, K. Introduction to Design for Manufacturing, Editor(s): Kraig Mitzner, Complete PCB Design Using OrCAD Capture and PCB Editor, Newnes, 2009, Pages 71-96, https://doi.org/10.1016/B978-0-7506-8971-7.00005-6.

Moreira, M. D. R. A., & Alves, R. A. F. (2012). Input-output control order release mechanism in a job-shop: how workload control improves manufacturing operations. *International Journal of Computational Science and Engineering, 7*(3), 214-223.

Oosterman, B., Land, M., & Gaalman, G. (2000). The influence of shop characteristics on workload control. *International journal of production economics, 68*(1), 107-119.

Park, P. S. (1991). The examination of worker cross-training in a dual resource constrained job shop. *European Journal of Operational Research, 52*(3), 291-299.

Park, P. S., & Bobrowski, P. M. (1989). Job release and labor flexibility in a dual resource constrained job shop. *Journal of operations management, 8*(3), 230-249.

Portioli-Staudacher, A., & Tantardini, M. (2012). A lean-based ORR system for non-repetitive manufacturing. *International Journal of Production Research, 50*(12), 3257-3273.

Pürgstaller, P., & Missbauer, H. (2012). Rule-based vs. optimisation-based order release in workload control: A simulation study of a MTO manufacturer. *International Journal of Production Economics, 140*(2), 670-680.

Ramirez, A., Zhu, S. C., & Benhabib, B. (1999, November). Moore automata for flexible routing and flow control in manufacturing workcells. *In Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*. CIRA'99 (Cat. No. 99EX375) (pp. 119-124). IEEE.

Ruiz-Torres, A. J., & Mahmoodi, F. (2007). Impact of worker and shop flexibility on assembly cells. *International journal of production research, 45*(6), 1369-1388.

Sabuncuoglu, I., & Karapınar, H. Y. (1999). Analysis of order review/release problems in production systems. *International Journal of Production Economics, 62*(3), 259-279.

Sagawa, J. K., & Land, M. J. (2018). Representing workload control of manufacturing systems as a dynamic model. *IFAC-PapersOnLine, 51*(2), 825-830.

Sammarco, M., Fruggiero, F., Neumann, W. P., & Lambiase, A. (2014). Agent-based modelling of movement rules in DRC systems for volume flexibility: human factors and technical performance. *International Journal of Production Research, 52*(3), 633-650.

Sirikrai, V., & Yenradee, P. (2006). Modified drum–buffer–rope scheduling mechanism for a non-identical parallel machine flow shop with processing-time variation. *International Journal of Production Research, 44*(17), 3509-3531.

Stevenson*, M., Hendry, L. C., & Kingsman, B. G. (2005). A review of production planning and control: the applicability of key concepts to the make-to-order industry. *International journal of production research, 43*(5), 869-898.

Stevenson, M. (2006). Refining a workload control (WLC) concept: a case study. *International Journal of Production Research, 44*(4), 767-790.

Thürer, M., Silva, C., Stevenson, M., & Land, M. (2012). Improving the applicability of workload control (WLC): the influence of sequence-dependent set-up times on workload controlled job shops. *International Journal of Production Research, 50*(22), 6419-6430.

Thürer, M., Stevenson, M., & Land, M. J. (2016). On the integration of input and output control: Workload Control order release. *International Journal of Production Economics, 174*, 43-53.

Thürer, M., Stevenson, M., & Silva, C. (2011). Three decades of workload control research: a systematic review of the literature. *International Journal of Production Research, 49*(23), 6905-6935.

Thürer, M., Stevenson, M., Land, M. J., & Fredendall, L. D. (2019). On the combined effect of due date setting, order release, and output control: an assessment by simulation. *International Journal of Production Research, 57*(6), 1741-1755.

Thürer, M., Stevenson, M., Silva, C., Land, M. J., & Fredendall, L. D. (2012). Workload Control and Order Release: A Lean Solution for Make-to-Order Companies. *Production and Operations Management, 21*(5), 939-953.

Thürer, M., Stevenson, M., Silva, C., Land, M. J., Fredendall, L. D., & Melnyk, S. A. (2014). Lean control for make-to-order companies: Integrating customer enquiry management and order release. *Production and Operations Management, 23*(3), 463-476.

Thürer, M., Land, M. J., Stevenson, M., & Fredendall, L. D. (2017). On the integration of due date setting and order release control. *Production Planning & Control, 28*(5), 420-430.

Uzun Araz, Ö., & Salum, L. (2010). A multi-criteria adaptive control scheme based on neural networks and fuzzy inference for DRC manufacturing systems. *International Journal of Production Research, 48*(1), 251-270.

Valeva, S., Hewitt, M., Thomas, B. W., & Brown, K. G. (2017). Balancing flexibility and inventory in workforce planning with learning. *International Journal of Production Economics, 183*, 194-207.

Wang, S. Y., Wang, L., Liu, M., & Xu, Y. (2013). An enhanced estimation of distribution algorithm for solving hybrid flow-shop scheduling problem with identical parallel machines. *The International Journal of Advanced Manufacturing Technology, 68*(9-12), 2043-2056.

Weng, M. X., Wu, Z., Qi, G., & Zheng, L. (2008). Multi-agent-based workload control for make-to-order manufacturing. *International Journal of Production Research, 46*(8), 2197-2213.

Yue, H., Slomp, J., Molleman, E., & Van Der Zee, D. J. (2008). Worker flexibility in a parallel dual resource constrained job shop. *International Journal of Production Research, 46*(2), 451-467.

Zäpfel, G., & Missbauer, H. (1993). New concepts for production planning and control. *European Journal of Operational Research, 67*(3), 297–320.

Zhao, B., Gao, J., Chen, K., & Xu, A. (2015). Workload control-related Workload Route Decision. *IFAC-PapersOnLine, 48*(3), 1422-1427.