Dynamic allocation of operators in a hybrid human-machine 4.0 context

Mémoire

Maude Beauchemin

Maîtrise en informatique - avec mémoire
Maître ès sciences (M. Sc.)

Québec, Canada

Dynamic allocation of operators in a hybrid human-machine 4.0 context

Mémoire

Maude Beauchemin

Sous la direction de:

Jonathan Gaudreault, directeur de recherche

Nadia Lehoux, codirectrice de recherche

Résumé

La transformation numérique et le mouvement « industrie 4.0 » reposent sur des concepts tels que l'intégration et l'interconnexion des systèmes utilisant des données en temps réel. Dans le secteur manufacturier, un nouveau paradigme d'allocation dynamique des ressources humaines devient alors possible. Plutôt qu'une allocation statique des opérateurs aux machines, nous proposons d'affecter directement les opérateurs aux différentes tâches qui nécessitent encore une intervention humaine dans une usine majoritairement automatisée. Nous montrons les avantages de ce nouveau paradigme avec des expériences réalisées à l’aide d’un modèle de simulation à événements discrets. Un modèle d'optimisation qui utilise des données industrielles en temps réel et produit une allocation optimale des tâches est également développé. Nous montrons que l'allocation dynamique des ressources humaines est plus performante qu'une allocation statique. L'allocation dynamique permet une augmentation de 30% de la quantité de pièces produites durant une semaine de production. De plus, le modèle d'optimisation utilisé dans le cadre de l'approche d'allocation dynamique mène à des plans de production horaire qui réduisent les retards de production causés par les opérateurs de 76 % par rapport à l'approche d'allocation statique. Le design d’un système pour l’implantation de ce projet de nature 4.0 utilisant des données en temps réel dans le secteur manufacturier est proposé.

Abstract

The Industry 4.0 movement is based on concepts such as the integration and interconnexion of systems using real-time data. In the manufacturing sector, a new dynamic allocation paradigm of human resources then becomes possible. Instead of a static allocation of operators to machines, we propose to allocate the operators directly to the different tasks that still require human intervention in a mostly automated factory. We show the benefits of this new paradigm with experiments performed on a discrete-event simulation model based on an industrial partner's system. An optimization model that uses real-time industrial data and produces an optimal task allocation plan that can be used in real time is also developed. We show that the dynamic allocation of human resources outperforms a static allocation, even with standard operator training levels. With discrete-event simulation, we show that dynamic allocation leads to a 30% increase in the quantity of parts produced. Additionally, the optimization model used under the dynamic allocation approach produces hourly production plans that decrease production delays caused by human operators by up to 76% compared to the static allocation approach. An implementation system for this 4.0 project using real-time data in the manufacturing sector is furthermore proposed.

Table of contents

List of figures

List of tables

Acknowledgements

This work was made possible thanks to many people to whom I would like to express my sincere gratitude. I must first thank my supervisors, Jonathan Gaudreault and Nadia Lehoux. Very obviously, this thesis would never have seen the light of day without you two unwavering allies. A huge thank you also to the precious collaborators of this project: Marc-André Ménard, Ludwig Dumetz and Claude-Guy Quimper. A special thank you to Caroline Cloutier, who assisted me in the redaction of this thesis in English. Thank you to the entire APN team but especially to Stéphane Agnard for his collaboration. Working with you has been a pleasure from day one. I would also like to thank those who have embellished my journey in the world of graduate studies and who have always been present with their many advice during the entirety of this project, the students of the PPC lab, especially Nicolas Leblanc and Anthony Deschênes. Finally, a special thank you to the evaluation board for their time reviewing this thesis.

The second part of these acknowledgements goes out to those who have allowed me to get to where I am today. You have listened to me change my career plan twenty times without ever panicking, and always believed in me even when I could not. Mom, Dad, Geneviève, Elisabeth and Édouard; you are the family everyone dreams of. Thank you to the one who shares my life and who cleaned our apartment and never complained when I woke him up at four in the morning to write. Pierre-Julien, I'd like to tell you that now that this thesis is over, I'm going to pull myself together, but we both know that would be a lie. Thank you, Mathilde and William, for listening to my roller coaster of emotions and helping me unravel the mess in my head every time of day and night I bolted in your apartment unannounced. A last special thanks to my dear friends and family who have always been by my side, I feel extremely lucky to have been surrounded by people like you in this journey.

Foreword

This work, titled "Dynamic allocation of operators in a hybrid human-machine 4.0 context", is presented in order to obtain the master's degree in computer science (M.Sc.) from Université Laval. It was carried out under the direction of Pr. Jonathan Gaudreault and the co-direction of Pr. Nadia Lehoux within the CRISI Research Consortium for Industry 4.0 Systems Engineering.

This thesis is written according to the principle of article insertion with a journal article entitled “Dynamic allocation of human resources: Case study in the metal 4.0 manufacturing industry” co-written with Pr. Jonathan Gaudreault, Pr. Nadia Lehoux, Marc-André Ménard, Stéphane Agnard and Pr. Claude-Guy Quimper. This article was submitted to Industrial Journal of Production Research (IJPR) in February 2022. I served as lead author responsible for all research, writing, work, and analysis related to the study.

Introduction

The machining industry has evolved along with the various industrial revolutions, and so has the relationship between humans and machines. While a machine was entirely dependent on a human operator following the first industrial revolution, the third revolution and the automation of processes allowed a machine to operate automatically, following a computer program. These specific types of machines are called computer-numerical control (CNC) machines and can be found in most metal manufacturing job-shops around the world. The use of CNC machines leads to different advantages, such as a reduced lead time, elimination of operator errors, and lower labour cost (Pabla and Adithan, 1994) since an operator may run two or more machines simultaneously. Nevertheless, a static allocation of an operator to a pre-determined set of machines may appear sub-optimal in terms of productivity as it typically leads to idle time for operators.

With the fourth industrial revolution upon us (Lasi *et al.*, 2014), which brings integration and interconnection of systems with the use of real-time data, the relationship between humans and machines may be redefined once again, with a set of operators dynamically allocated to a set of machines. While this proposition suggests a reduction of the important idle times and a possible productivity gain, many challenges may arise. The allocation decisions currently made by production managers are limited to the allocation of one operator to a set of machines for their work shift. A dynamic worker-machine allocation strategy would imply multiple allocation decisions per minute which a human could no longer be expected to do. Additionally, the task allocations cannot be planned much in advance since the production is constantly evolving, and unforeseen events happen frequently. When these events happen, the task allocation plan must be adapted, so the use of real-time data is necessary, as well as powerful algorithms which generate new optimal plans quickly.

In this master thesis, we propose and study a new paradigm of dynamic allocation of human resources in a 4.0 manufacturing context. Currently, an operator is allocated to a set of machines for which he or she is responsible during the work shift. With our industrial partner, APN Global, we have undertaken to implement this new dynamic allocation paradigm in their 4.0 high-precision metal manufacturing factory. Thus, we set the following objectives:

1. Analyze a metal manufacturing system, focusing on the different tasks necessary for the production of metal parts
2. Define scenarios allowing real-time allocation of operators for such a system
3. Measure the impact of this form of allocation in terms of number of parts produced in a week and delay caused by operators in the production
4. Facilitate the implementation of the dynamic allocation method in the real world

The study was carried out in four main phases. First, a literature review was conducted to explore the different available techniques and methods, as well as what had been previously done regarding dynamic allocation. The methods to use for this project, such as optimization and discrete-event simulation, were also chosen in this part.

In the second part, we focused on developing and running a discrete-event simulation model based on our industrial partner's factory, in order to study the dynamic allocation paradigm. The model included 22 CNC machines as well as four measurement machines and represented the entire machining section of the factory. The different tasks were modeled with extensive detail. Using this model, we were able to compare the static with the dynamic allocation and measure the impact of such a paradigm change on the factory's productivity, such as, for example, the number of parts produced during one week of production.

In the third part, a real-time optimization/allocation model was developed, which led to the submission of an article. The model uses Constraint Programming (CP) (Rossi, van Beek and Walsh, 2006) and was implemented using the MiniZinc language. The model was able to produce an optimal task allocation plan for the next hour of production using real-time data in a few seconds. Different scenarios of operators' allocation method (static or dynamic) and workforce skill levels (current skills or polyvalent operators) were compared based on productivity (i.e., tardiness of tasks) as well as human factors (i.e., operator occupation rates).

Finally, the implementation of the project in an industrial setting was tackled in the last part of the methodology. We designed a system to implement the real-time dynamic task allocation paradigm proposed in the factory. Future industrial developments for the project were also established.

The results obtained indicate that implementing a dynamic allocation paradigm of human resources can lead to important productivity gains in the manufacturing sector. Even when current operator training-levels are considered, this approach leads to important improvements. Indeed, there is no need to suppose perfectly versatile operators in order to profit from the productivity gains. When simulating a week of production, we showed a 30% increase in the quantity of parts produced when using a dynamic allocation method. Additionally, using the allocation algorithm with a dynamic allocation approach yields hourly production plans that decrease production delays caused by human operators by 76%.

This project contributes to research by proposing a new dynamic allocation paradigm in the manufacturing industry with a high-performance optimization model that allows real-time allocation of operators to the various tasks at hand. Different experiments conducted with the simulation model as well as with the optimization model show the strength of this new paradigm. This contribution is also considerable for the manufacturing sector, which can base their future decisions on the results of this study. Operator training is also addressed. Furthermore, an implementation framework is proposed, which may be followed by other industrials wishing to implement a similar project using real-time data in their factory.

This thesis is divided into five chapters. In the first chapter, a literature review highlights the different preliminary concepts of this study. The second chapter describes more thoroughly the objectives pursued and the methods used in this project. In the third chapter, the simulation model is described as well as different experiments performed, and results obtained. The fourth chapter introduces an article that was written in the context of this research, which presents the optimization model. Finally, the fifth chapter presents the design of system for the implementation of those concepts.

Chapter 1: Preliminary concepts

In this chapter, we introduce the main concepts used in this project with the existing literature. The concepts of simulation, optimization and Industry 4.0 are therefore explained.

1.1 Computer simulation

Computer simulation is a technique that consists of using computers to imitate a real-life system (Law and Kelton, 1991). The origins of computer simulation date back to the 1940s during the Second World War when John von Neumann and Stanislaw Ulam used it to solve neutron diffusion problems when designing a hydrogen bomb (Goldsman, Nance and Wilson, 2010). Simulation is used when systems cannot be modeled with analytical methods such as algebra. This can happen when systems are extremely complex, which often is the case in real-life systems. As first proposed by Schmidt and Taylor (1970), a system consists of entities which can be people or machines that interact together in order to reach a logical end. As shown in Figure 1, Law and Kelton (1991) presented the different ways to study a system.

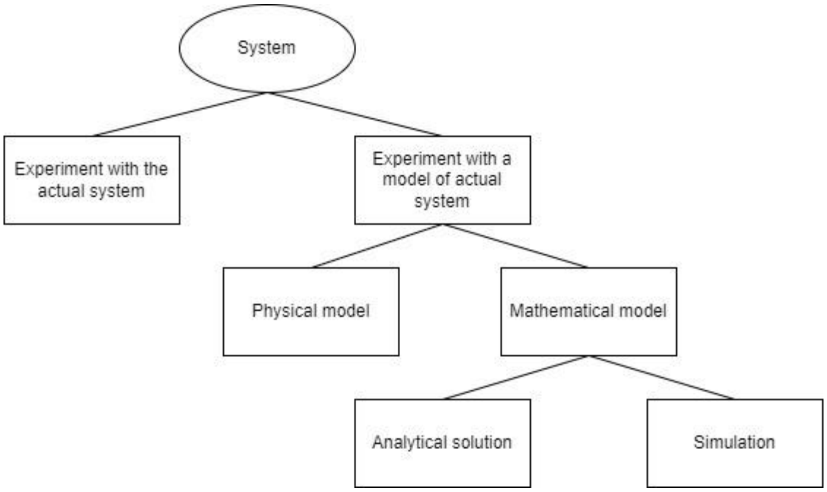

System
Experiment with the actual system
Experiment with a model of actual system
Physical model
Mathematical model
Analytical solution
Simulation

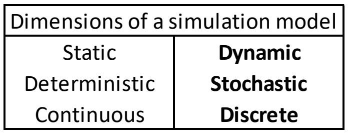
Figure 1: Ways to study a system, as represented by Law and Kelton (1991).

In this master’s thesis project, different allocation policies need to be tested for operators in a factory with many complex processes. Conducting experiments with the actual system is therefore not conceivable. As for a physical model, it would be extremely complex and hard to simulate the system in such a way. Equally, an analytical solution would be too complex, and it would be harder to account for the stochasticity present in the system. Computer simulation was therefore selected for the project. As listed in Banks *et al.* (2005), simulation is an appropriate technique when studying the internal interactions of a complex system. Simulation can be used to test organizational changes and observe their effect on the system's behaviour.

1.1.1 Discrete-event simulation

Computer simulation can be characterized through three dimensions, as proposed by Law and Kelton (1991) and represented in Table 1. In a *static* simulation model, time plays no role. In our project, we therefore need to use *dynamic* simulation since time has an effect on the allocation policies. For a *deterministic* simulation model, the output is pre-determined according to the input and the relations established between the entities. Since this is not the case in our project as some aspects are *stochastic*, a stochastic simulation model will be used. A simulation model can finally be represented as *continuous* if the object of interest needs to be a flow. If we can represent the different events in the system as events occurring at discrete moments in time, we may then use *discrete*-event simulation to analyze it. For this project, the system is then represented as a chain of discrete events that have an effect on its state. In summary, we will propose a dynamic, stochastic, discrete-event simulation model to study the various worker allocation policies. This type of model will be referred to only as a discrete-event simulation model from now on, as is the case in the related literature.

Table 1: Dimensions of a simulation model, as proposed by Law and Kelton (1991).



Dimensions of a simulation model	
Static	**Dynamic**
Deterministic	**Stochastic**
Continuous	**Discrete**

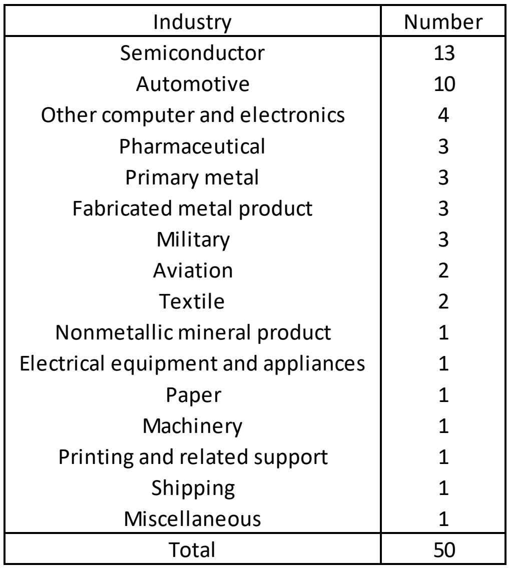
While the use of discrete-event simulation traces back to the 1950s, when it was written in machine code, it is not until the 1980s that it began being used as a decision-support tool in the manufacturing industry (Robinson, 2005). Many tools now exist to create discrete-event simulation models, such as Simio (Pegden, 2008), AnyLogic (Borshchev, 2014) and Arena (Hammann and Markovitch, 1995).

1.1.2 Discrete-event simulation in the manufacturing industry

Some areas of application of discrete-event simulation, as listed in Banks et al. (2005), consist of manufacturing applications, business processing, project management, health care, military, and transportation. In this project, we are interested in a case study in the manufacturing sector.

The first review on simulation in the manufacturing industry was published in 1992 and stated that awareness of discrete-event simulation in the manufacturing industry was very low (Hollocks, 1992). A decade later, Ingemansson, Bolmsjö and Harlin, (2002) found that of the 80 manufacturing companies they surveyed in Sweden, 12 were using discrete-event simulation in an active way, so they concluded that awareness of this technology in the industry was on the rise. Four years later, another survey looked at the different application sectors for discrete-event simulation as proposed in 52 papers of the *Winter Simulation Conference* proceedings of the previous years (Semini, Fauske and Strandhagen, 2006). Their results show that discrete-event simulation was already used in the manufacturing sector in the previous 15 years, by several sub-sectors, with a concentration in the semiconductor and automotive sectors.

Table 2: Use of discrete-event simulation in the manufacturing industry, based onSemini, Fauske and Strandhagen (2006).



Industry	Number
Semiconductor	13
Automotive	10
Other computer and electronics	4
Pharmaceutical	3
Primary metal	3
Fabricated metal product	3
Military	3
Aviation	2
Textile	2
Nonmetallic mineral product	1
Electrical equipment and appliances	1
Paper	1
Machinery	1
Printing and related support	1
Shipping	1
Miscellaneous	1
Total	50

In the last decades, several articles addressed the use of discrete-event simulation in the manufacturing industry. For example, Ferjani *et al.* (2017) proposed and compared different assignation heuristics with a simulation model when assigning human resources subject to fatigue in a manufacturing system. Lidberg, Pehrsson and Ng (2018) aimed at improving factory productivity with a discrete-event simulation model and multi-objective optimization. Allgeier *et* *al.* (2020) evaluated lot release policies in a power semiconductor facility, which is known to be one of the most complex manufacturing processes, by using discrete-event simulation.

In this project, the sub-sector we are interested in is the fabricated metal product sector. To the best of our knowledge, only a few articles have directly tackled this sector in a study using discrete-event simulation.

1.1.3 Discrete-event simulation of operators' tasks allocation

While it is quite common to model a production system and its different products in a discrete-event simulation model (Detty and Yingling, 2000; Zupan and Herakovic, 2015; Huynh, Akhtar and Li, 2020), we have a different approach in this project. Indeed, our need is to model the different operations or tasks performed by the machines and operators.

This approach has been used in the past. For instance, Keller (2002) modeled the human operators workload with a discrete-event simulation model. Jung *et al.* (2020) modeled a garment production line with accurate task times in order to measure and help improve productivity. In the hydraulic excavator sector, Hughes and Jiang (2010) proposed a task-network system imbedded in a discrete-event simulation model to help improve operator performances. In Nehme, Crandall and Cummings (2008), situational awareness (combination of perception of elements in the environment, the comprehension of their meaning, and the projection of their status in the future (Endsley, 1995)) of unmanned-vehicle operators was modeled using discrete-event simulation. Different operator strategies and their effect on situational awareness were tested. Authors suggested that situational awareness decreased with increased utilization rates of the operators, since they do not have the necessary time to assess their surroundings. Also, operator performance in a nuclear power-plant was predicted using discrete-event simulation to focus on operators' tasks (Yow *et al.*, 2005). Finally, the performance of a labour control strategy addressing the allocation of skilled and unskilled operators was tested with a discrete-event simulation-based manufacturing model (Le *et al.*, 2013). It was proven to increase productivity and utilisation of the resources.

1.2 Optimization

Optimization is a branch of mathematics that aims at finding the best possible solution to a problem using different techniques. Its foundation is based on two main articles, the Theory of Games and Economic Behavior (Deming, Neumann and Morgenstern, 1944) and the article presenting the discovery of the simplex algorithm (Dantzig and Wolfe, 1960). An optimization problem aims to find the best solution between all possible solutions. Optimization problems can be either discrete or continuous, with many sub-categories in each. A classification of the optimization problems is presented in Figure 2. In this thesis, the optimization was first formulated as a mixed-integer linear programming problem, which is a discrete optimization problem. Many problems can be written as mixed-integer linear programming problems with the use of different formulation techniques (Vielma, 2015).

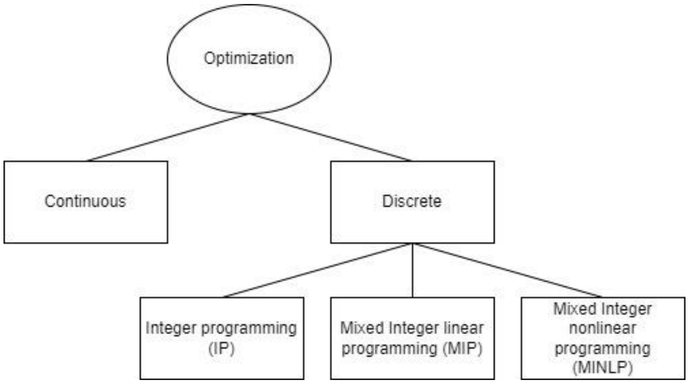

Optimization
Continuous
Discrete
Integer programming (IP)
Mixed Integer linear programming (MIP)
Mixed Integer nonlinear programming (MINLP)

Figure 2: Optimization problem classification, based on Rohde (2019).

Various software packages are available to solve linear-programming models, such as AMPL (Fourer, no date) and Lindo (Lin and Schrage, 2009).

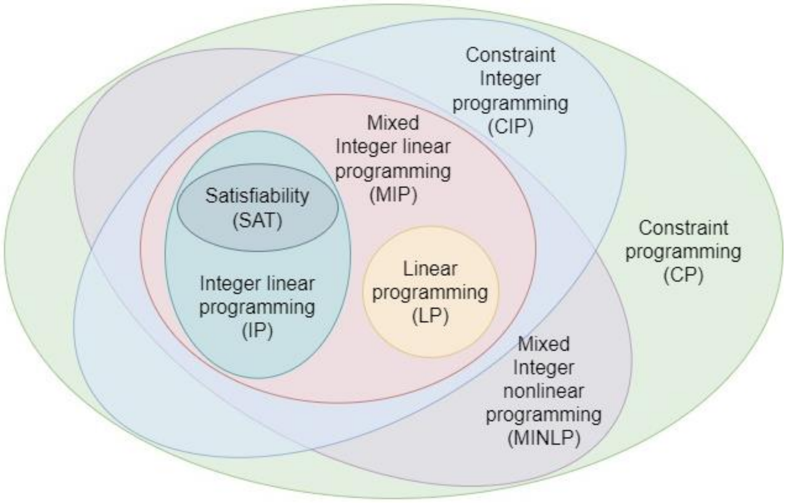


Figure 3: Optimization problem classification, based on Gleixner (2018).

In a broader sense, our optimization problem could also be formulated using constraint programming.

1.2.1 Constraint programming

Constraint programming is described in Rossi, van Beek and Walsh (2006) as a “powerful paradigm for solving combinatorial search problems”. More precisely, it consists of a user declaratively stating its constraints characterizing the feasible solutions of a given problem for a set of decision variables. Crucial concepts of constraint programming include backtrack search and constraint propagation which serve as a search strategy, which must be provided by the user. The first paper that provided a statistical and experimental evaluation of backtrack search was published in the *Artificial Intelligence Journal* in 1980 (Haralick and Elliott, 1980).

Constraint programming can be used to solve satisfaction problems as well as optimization problems by using a search tree. A search tree consists of branches representing partial solutions to the problem. The root of the tree represents the initial partial solution where no variables have any value. Each node below the root represents a possible value for a given variable, which inherits from the precedent partial solution. Branching consists of choosing a value for a variable. Backtracking occurs when no further branching is possible or if an incoherence is detected that does not respect the constraints. In a satisfaction problem, as soon as a complete solution that respects all constraints is found, the search with the tree is ended. In an optimization problem, it is possible that all nodes must be explored before the search ends.

The advantages of constraint programming compared to integer programming are the use of global constraints, non-linear constraints, and constraint propagation (Walsh, 2001). A global constraint is a constraint that manages to encapsulate the relation between multiple variables (van Hoeve and Katriel, 2006). It is especially helpful to deal with constraints that are common between different problems since it can use a specialized routine to handle it. Non-linear constraints are not acceptable in integer programming, which can make the formulation of constraints a hard task and imply linear relaxation techniques. In constraint programming, constraints are very flexible and can be expressed more directly to better represent the real world. Finally, constraint propagation is defined by (Rossi, van Beek and Walsh, 2006) as a "form of reasoning in which, from a subset of the constraints and their domains, we can infer more restrictive domains or constraints." It is considered to be one of the most important concepts in constraint programming since it is extremely helpful to finding solutions faster by examining a subset of all solutions.

1.2.2 The (flexible) Job-Shop Scheduling Problem

A common problem solved with optimization techniques is the job-shop problem, or its variant the flexible job-shop problem. In our project, the problem addressed shares many similarities with the flexible job-shop scheduling problem. This problem has been addressed by several authors (Fattahi, Jolai and Arkat, 2009; Ham and Cakici, 2016; Erming Zhou, Jin Zhu, and Ling Deng, 2017). Additionally, Demir and Kürşat İşleyen (2013) present different mathematical models to solve this problem. Heuristic approaches were also used to solve the flexible job-shop scheduling problem (Fattahi, Saidi Mehrabad and Jolai, 2007). A review on the flexible job-shop scheduling problem was proposed by Xie *et al.* (2019).

Scheduling using optimization techniques is a popular research subject in the manufacturing sector. Lan and Lan (2005) proposed a combinatorial manufacturing resource planning model applicable to the CNC machining industry. They attempted to schedule machines as well as human resources in order to optimize the production profit.

While a lot of research has been conducted on the job-shop scheduling problem and its variations for scheduling jobs in factories, we apply it in a different context, and focus on the allocation of real-time tasks to operators.

1.3 Industry 4.0

Industry 4.0 first was a German industrial program created in 2011 at the Hannover Fair in response to the changes occurring in the industry at the moment that led to believe we were experiencing a fourth industrial revolution (Devezas, Leitão and Sarygulov, 2017). Two years later, the Industry 4.0 Working Group presented the first report on Industry 4.0 at the Hannover Fair (Henning, 2013). In their report, they presented the fourth industrial revolution in a time scale. Figure 4 is taken from this report.

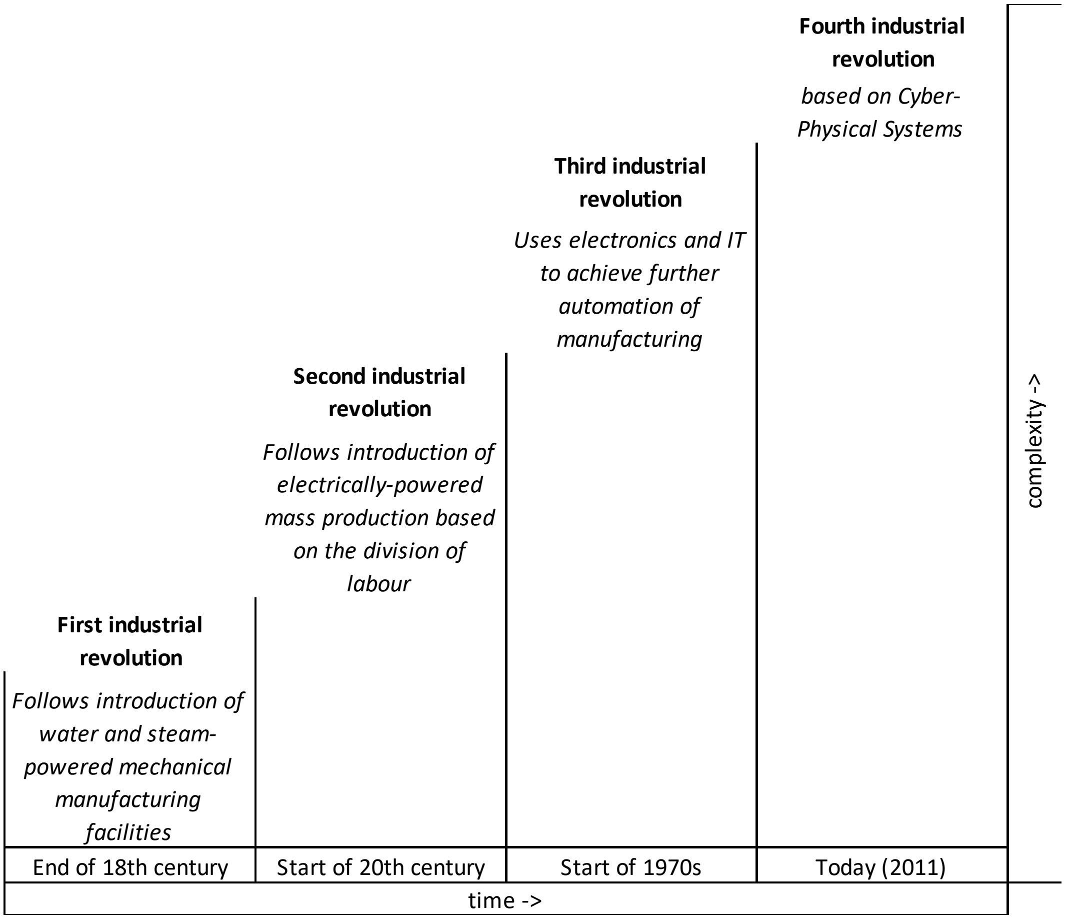

Fourth industrial revolution
based on Cyber-Physical Systems
Third industrial revolution
Uses electronics and IT to achieve further automation of manufacturing
Second industrial revolution
Follows introduction of electrically-powered mass production based on the division of labour
First industrial revolution
Follows introduction of water and steam-powered mechanical manufacturing facilities
End of 18th century
Start of 20th century
Start of 1970s
Today (2011)
time ->
complexity ->

Figure 4: The four industrial revolutions, based on Kagermann (2013).

While many technologies such as the internet-of-things, 3D printing, artificial intelligence, and cloud computing are part of this revolution, its defining characteristic is the interconnection and integration of systems and people in the entire value chain. Lasi *et al.* (2014) concluded that the term "Industry 4.0" described new IT advances primarily driven by changes in manufacturing systems with organizational implications. Popkova, Ragulina and Bogoviz (2019) suggested that this new revolution was driven in part by the economic crises of the early 2000s which led to an overproduction of industrial goods, suggesting that the current economic system was not compatible with the previous technological model.

Figure 5 presents the technological trends and the design principles of Industry 4.0 as discovered in a review of 178 papers by Ghobakhloo (2018). The technological trends used in our project are bolded, as well as the design principles followed.

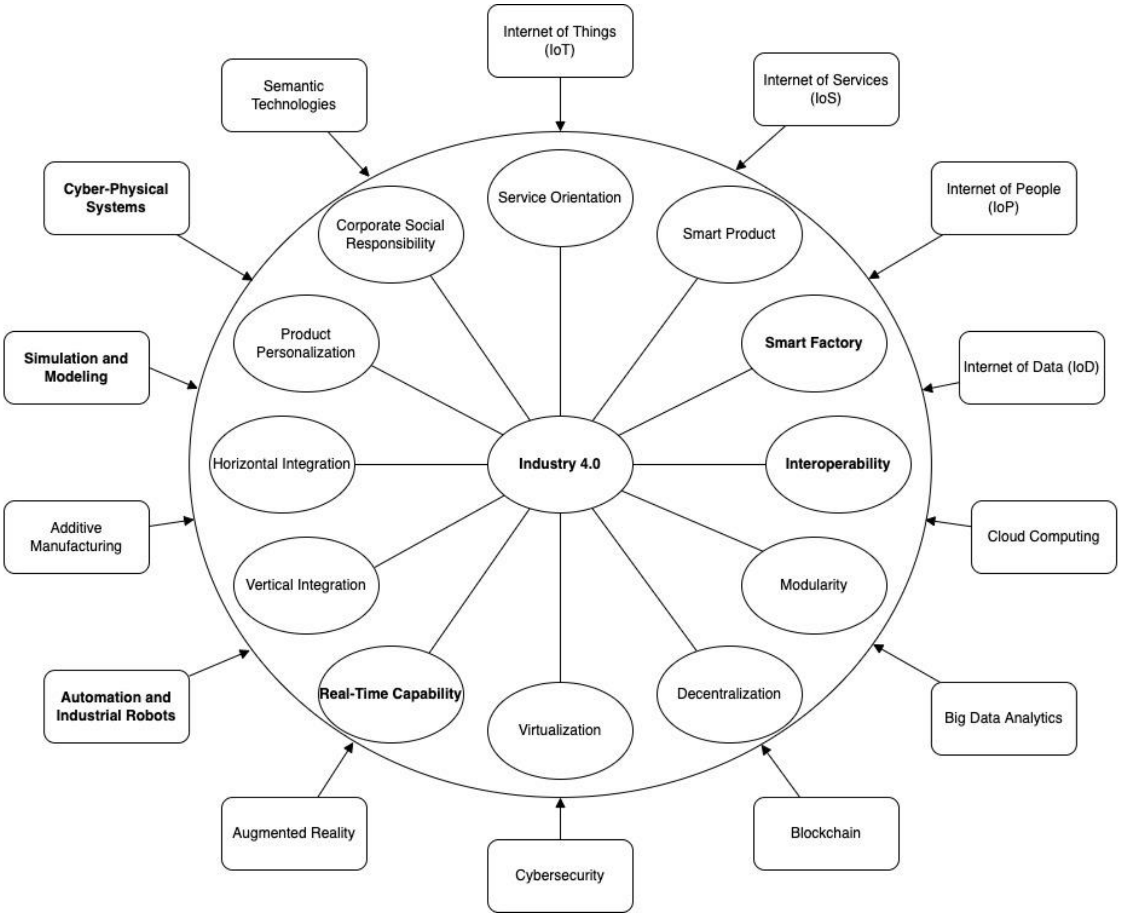


Figure 5: Design principles and technology trends of Industry 4.0, based on Ghobakhloo (2018).

Different reviews on Industry 4.0 were published during the last decade to monitor its evolution in the industrial world. Wichmann, Eisenbart and Gericke (2019) identified the different benefits of Industry 4.0 proposed in the 50 most relevant papers on Industry 4.0 at the time of their review. As shown in Figure 6, the most recurrent benefits expected from Industry 4.0 are a dynamic manufacturing system, an increased connectivity, and decentralised autonomous production systems.

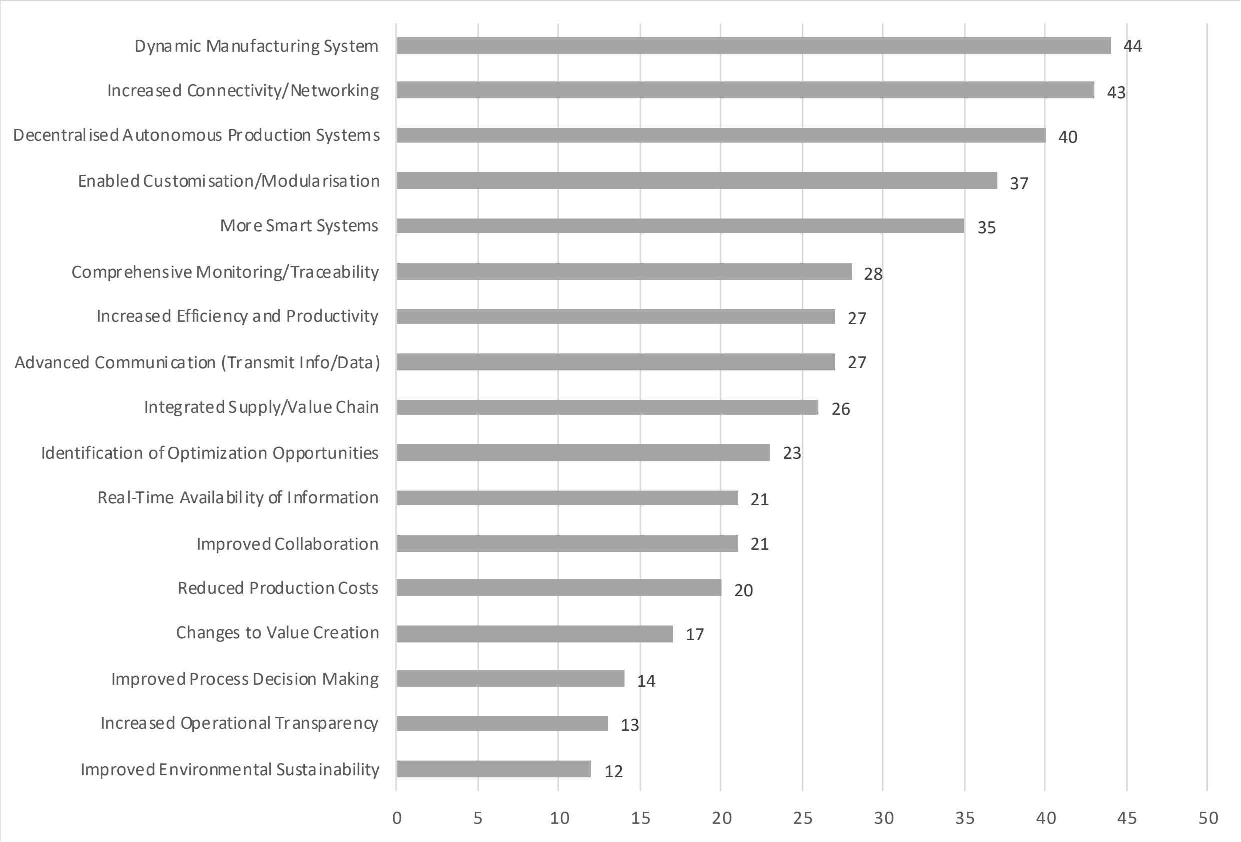


Figure 6: Industry 4.0 manufacturing benefit propositions, as presented in Wichmann, Eisenbart and Gericke (2019).

In our project, the main benefits targeted are a dynamic manufacturing system, increased efficiency and productivity, identification of optimization opportunities, as well as improved process decision making.

1.3.1 Simulation and Industry 4.0

Since the arrival of new technologies characterizing the Industry 4.0 era, new possibilities regarding simulation have come to life. In their review, Turner *et al.* (2016) looked at the combined areas of discrete-event simulation with virtual reality in order to summarize future research directions. An entire 300-page book was also written on Simulation in the Industry 4.0 (Gunal, 2019). Their analysis of Google Scholar keyword search results for “simulation” and “Industry 4.0 has been on the rise for the last ten years, with no articles written in 2011 and exceeding 6 000 articles in 2018. A recent review (see Paula Ferreira *et al.* 2020) has also observed an increasing trend in simulation-based research in Industry 4.0 for the years 2016-2020.

The concept of a digital twin is defined by Boschert and Rosen (2016) as a “comprehensive physical and functional description of a component, product or system, which includes more or less all information which could be useful in all—the current and subsequent—lifecycle phases”. This can easily refer to a simulation model that represents the production system. A digital twin is updated based on real-world data and can be used periodically to make decisions, by using simulation. As described by (Boschert, Heinrich and Rosen, 2018), a digital twin consists of a collection of digital artefacts that include engineering and operation data as well as behaviour descriptions represented with simulation model(s) and it is used to extract solutions relevant for the real system, as is the case in our project.

1.3.1.1 Discrete-event simulation with real-time data

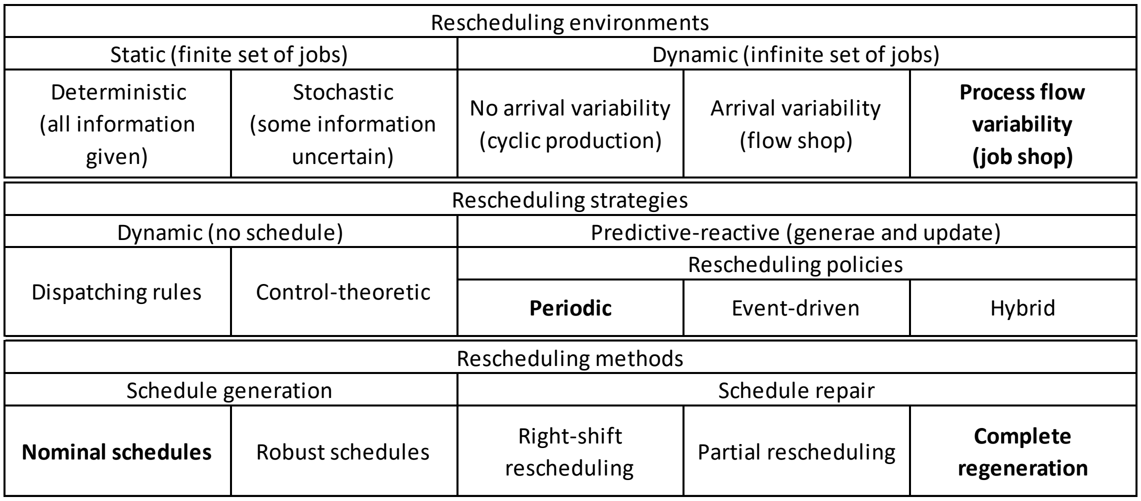
Discrete-event simulation has also been used with real-time data. In Turker *et al.* (2019), the authors used real-time data in order to test different classic dispatching rules to solve the job-shop scheduling problem. Brik *et al.* (2019) tackled task rescheduling in a flexible flow-shop with an optimization problem, solved using a meta-heuristic (tabu search) with localization information based on the Internet-of-things (IoT). Garrido and Sáez (2019) proposed a framework for the automatic generation of simulation models using real-time data applied to industrial transportation and warehouse systems.

1.3.2 Real-time or dynamic scheduling in manufacturing

While real-time scheduling using real-time data has only recently been made possible with the technologies characterizing Industry 4.0, scheduling and planning in the manufacturing industry has been a popular subject of research in the last decades.

Scheduling tasks or jobs in real-time brings its own challenges, since we must deal with stochastic events that may happen during the execution of the production plan. Waschneck *et al.* (2016) addressed the complex job-shop scheduling problem from an Industry 4.0 perspective. They identified different challenges from the literature and pointed to future directions that could be taken by the job-shop scheduling problem under Industry 4.0. They point out that rescheduling strategies can be enhanced under Industry 4.0 in order to increase flexibility in job-shop environments. Zhang *et al.* (2019) also reviewed job-shop scheduling research and its new perspectives under Industry 4.0 and concluded that the future of the job-shop scheduling problem is decentralized scheduling using real-time information. Finally, Leusin *et al.* (2018) also agreed that real-time data exchange is key to solving the job-shop scheduling problem in the Industry 4.0 era, based on tests using a simulation model of a real industrial case.

While real-time scheduling at large is not a new research area and has been studied for the last decades, its first classification was proposed by Vieira, Herrmann and Lin (2003) and shown in Figure 7. The technological advances that characterize Industry 4.0 have brought new perspectives for the resolution of this problem.



Rescheduling environments				
Static (finite set of jobs)		Dynamic (infinite set of jobs)		
Deterministic (all information given)	Stochastic (some information uncertain)	No arrival variability (cyclic production)	Arrival variability (flow shop)	**Process flow variability (job shop)**

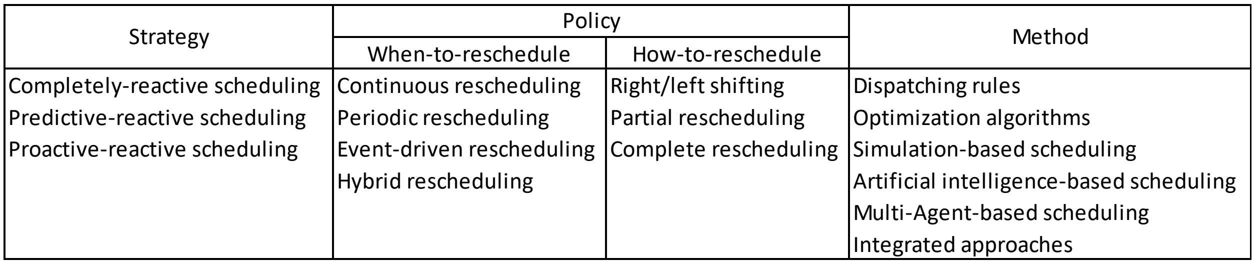
Rescheduling strategies				
Dynamic (no schedule)		Predictive-reactive (generae and update)		
Dispatching rules	Control-theoretic	Rescheduling policies		
		Periodic	Event-driven	Hybrid

Rescheduling methods				
Schedule generation		Schedule repair		
Nominal schedules	Robust schedules	Right-shift rescheduling	Partial rescheduling	**Complete regeneration**

Figure 7: Rescheduling framework as proposed by Vieira, Herrmann and Lin (2003).

Real-time scheduling was also classified in an exhaustive literature review (Ghaleb, Zolfagharinia and Taghipour, 2020) of real-time production scheduling in the Industry 4.0 context. The classification is presented in Figure 8 and is based heavily on the framework proposed by Vieira, Herrmann and Lin (2003). For more details on the classification, see Ghaleb, Zolfagharinia and

Taghipour (2020). Garrido and Sáez (2019) even proposed machine learning in order to determine the best when-to-reschedule policy in a flexible job-shop.



Strategy	Policy		Method
	When-to-reschedule	How-to-reschedule	
Completely-reactive scheduling	Continuous rescheduling	Right/left shifting	Dispatching rules
Predictive-reactive scheduling	Periodic rescheduling	Partial rescheduling	Optimization algorithms
Proactive-reactive scheduling	Event-driven rescheduling	Complete rescheduling	Simulation-based scheduling
	Hybrid rescheduling		Artificial intelligence-based scheduling
			Multi-Agent-based scheduling
			Integrated approaches

Figure 8: Real-time scheduling classification as proposed by Ghaleb, Zolfagharinia and Taghipour (2020).

Khayal (2018) proposed a review on dynamic scheduling in manufacturing. Uhlmann and Frazzon (2018) reviewed articles on production rescheduling in a dynamic environment and suggested the research subject was still of interest.

Rahmani and Ramezanian (2016) proposed a stable reactive approach in dynamic flexible flow shop scheduling when dealing with unexpected disruptions, aiming to provide stable rescheduling against disruptions with a variable neighborhood search. Zhu *et al.* (2019) proposed an adaptive method to solve the real-time job-shop scheduling problem with the use of a multi-agent system. Sreekara Reddy *et al.* (2018) focused specifically on machine breakdowns as real-time events in order to propose a hybrid multi-objective meta-heuristic algorithm to solve the flexible job-shop real-time scheduling problem. Palombarini and Martínez (2019) proposed closed-loop rescheduling using deep artificial intelligence in order to handle unplanned disturbances in a manufacturing system. Another paper adopted different methods such as constraint programming, mixed-integer programming and dispatching rules in order to address the flexible assembly job-shop scheduling problem in a dynamic manufacturing environment (Zhang and Wang, 2018). Production rescheduling has also been studied under the angle of sustainability, aiming at choosing the best rescheduling method to minimize energy consumption (Salido *et al.*, 2017).

Pfitzer *et al.* (2018) proposed an event-driven rescheduling method in job-shop environments to deal with unpredictable incoming orders. They evaluated the performance of their method with simulation experiments.

Although simulation promises many benefits, the collection of input data can prove to be a challenge. Onggo, Hill and Brooks (2013) conducted a survey revealing that data collection problems are common and have significant impacts on simulation projects in the manufacturing industry. Barlas and Heavey (2016) reviewed different papers on automated input data in discrete-even simulation models. They believed automated data collection could help manage data collection challenges in this type of project.

1.3.2.1 Mathematical optimization with real-time data

With the technologies of Industry 4.0 such as smart monitoring, optimization can be executed in real-time. Ghaleb, Taghipour and Zolfagharinia (2020) developed a real-time integrated optimization model to schedule operations and maintenance in a manufacturing system. When used in a case study, they showed that their advanced method outperformed a simple right- shifting method currently used by the company.

1.3.2.2 Robot-human collaboration

In these dynamic environments under Industry 4.0, human operators may also be brought upon collaborating with robots. This type of environment is addressed in Evangelou *et al.* (2021), who proposed an approach for task planning using artificial intelligence and real-time rescheduling. Another review on human-robot interaction focusing on task planning was published by Tsarouchi, Makris and Chryssolouris (2016). Al-Behadili, Ouelhadj and Jones (2019) proposed a multi-objective optimization model that considers different real-time events in order to produce a robust plan for the permutation flow shop scheduling problem. Stochastic and dynamic disruptions were addressed in a robust and stable flow shop scheduling proactive-reactive approach (Liu *et al.*, 2017). Human attention in a robot and human collaboration environment was also addressed in Yao and Zhang (2020) in which a human receives collaboration requests from many robots simultaneously.

1.3.3 Implementation under Industry 4.0

While the Industry 4.0 revolution promises new interesting benefits for the manufacturing sector, its implementation faces many challenges. Butt (2020) proposed a strategic roadmap for the manufacturing industry to implement Industry 4.0 based on the Lean Six Sigma framework. The author justified this choice by stating that the manufacturing industry is already familiar with this framework. The different phases proposed in the framework, as presented in Figure 9, are to define the problem and to identify the existing limitations (1), to collect data (2) to clarify the requirements, to evaluate the Industry 4.0 tools already established in the company (3), to optimize the processes with different tools such as simulation and mathematical optimization (4), to develop a detailed plan by including more people (5), to validate the prototype created (6), and to implement a full-scale pilot (7). The implementation is executed following the DMAIC framework which consists of an iteration between the five phases: define, measure, analyze, improve, and control, which then leads to the final step which is the digitization.

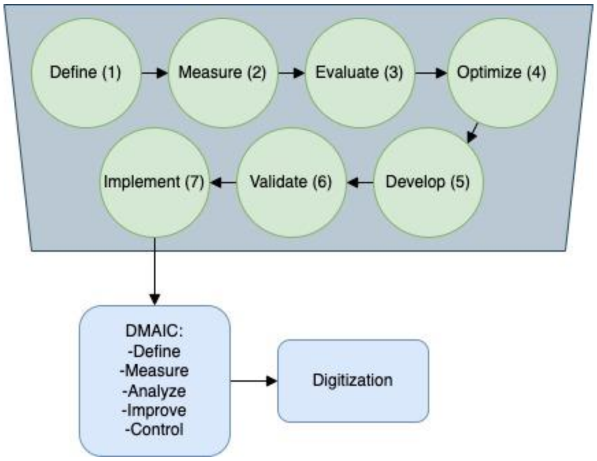
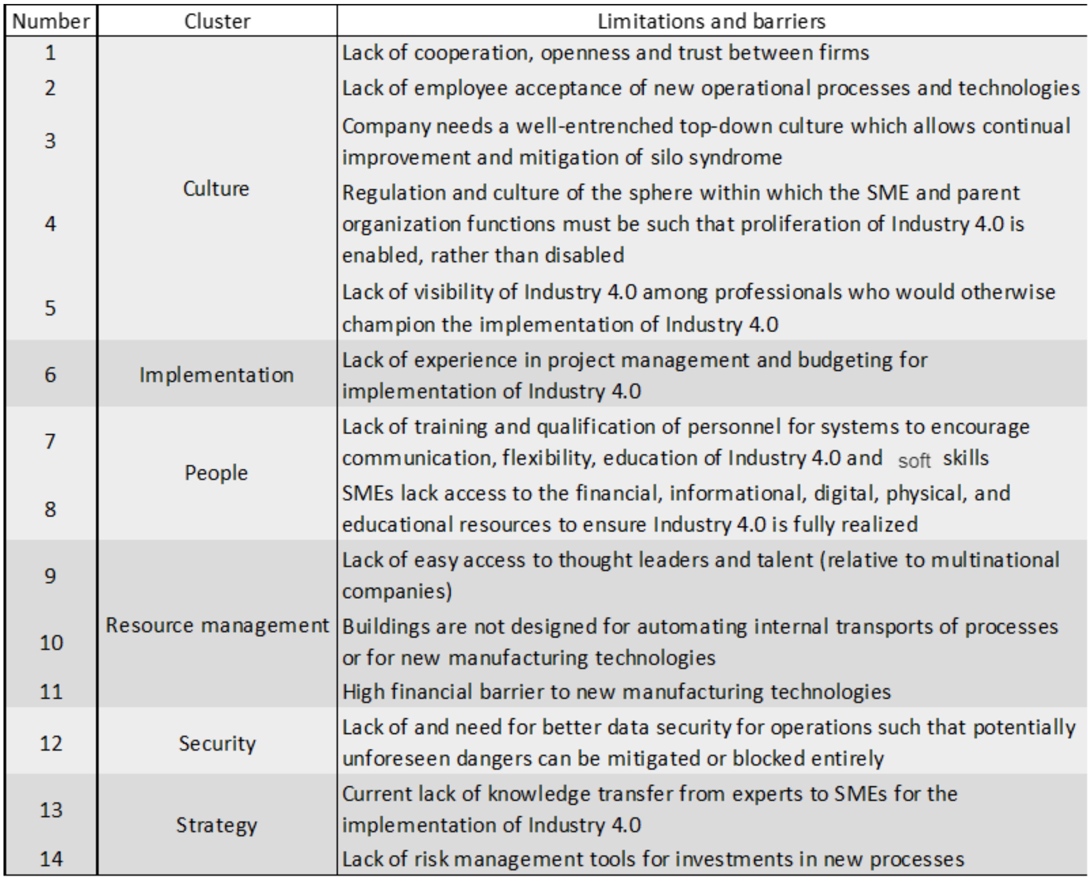


Figure 9: Strategic roadmap for Industry 4.0 implementation Butt (2020).

1.3.4 Implementation under Industry 4.0 in SMEs

Following an established framework is even more important in a small-and-medium enterprise (SME) since they face specific challenges related to the implementation of Industry 4.0 projects. As seen in Figure 10, Matt, Modrák and Zsifkovits (2020) stated that specific challenges faced by SMEs can be a “lack of employee acceptance of new operational processes and technologies, a lack of training and qualification of personnel for systems to encourage communication, flexibility, education of Industry 4.0 and soft skills”, etc.



Number	Cluster	Limitations and barriers
1	Culture	Lack of cooperation, openness and trust between firms
2		Lack of employee acceptance of new operational processes and technologies
3		Company needs a well-entrenched top-down culture which allows continual improvement and mitigation of silo syndrome
4		Regulation and culture of the sphere within which the SME and parent organization functions must be such that proliferation of Industry 4.0 is enabled, rather than disabled
5		Lack of visibility of Industry 4.0 among professionals who would otherwise champion the implementation of Industry 4.0
6	Implementation	Lack of experience in project management and budgeting for implementation of Industry 4.0
7	People	Lack of training and qualification of personnel for systems to encourage communication, flexibility, education of Industry 4.0 and soft skills
8		SMEs lack access to the financial, informational, digital, physical, and educational resources to ensure Industry 4.0 is fully realized
9	Resource management	Lack of easy access to thought leaders and talent (relative to multinational companies)
10		Buildings are not designed for automating internal transports of processes or for new manufacturing technologies
11		High financial barrier to new manufacturing technologies
12	Security	Lack of and need for better data security for operations such that potentially unforeseen dangers can be mitigated or blocked entirely
13	Strategy	Current lack of knowledge transfer from experts to SMEs for the implementation of Industry 4.0
14		Lack of risk management tools for investments in new processes

Figure 10: : Limitations and barriers for the design of smart manufacturing systems in SMEs introducing Industry 4.0 projects as proposed by Matt, Modrák and Zsifkovits (2020).

Focusing on these specific challenges, the authors proposed an implementation toolkit to support SMEs in the implementation of 4.0 technologies. This toolkit consists of four phases (organizational analysis, gap analysis, economical analysis, and implementation guideline) as shown in Figure 11. The proposed toolkit has many similarities with the DMEODVI framework proposed by Butt (2020).

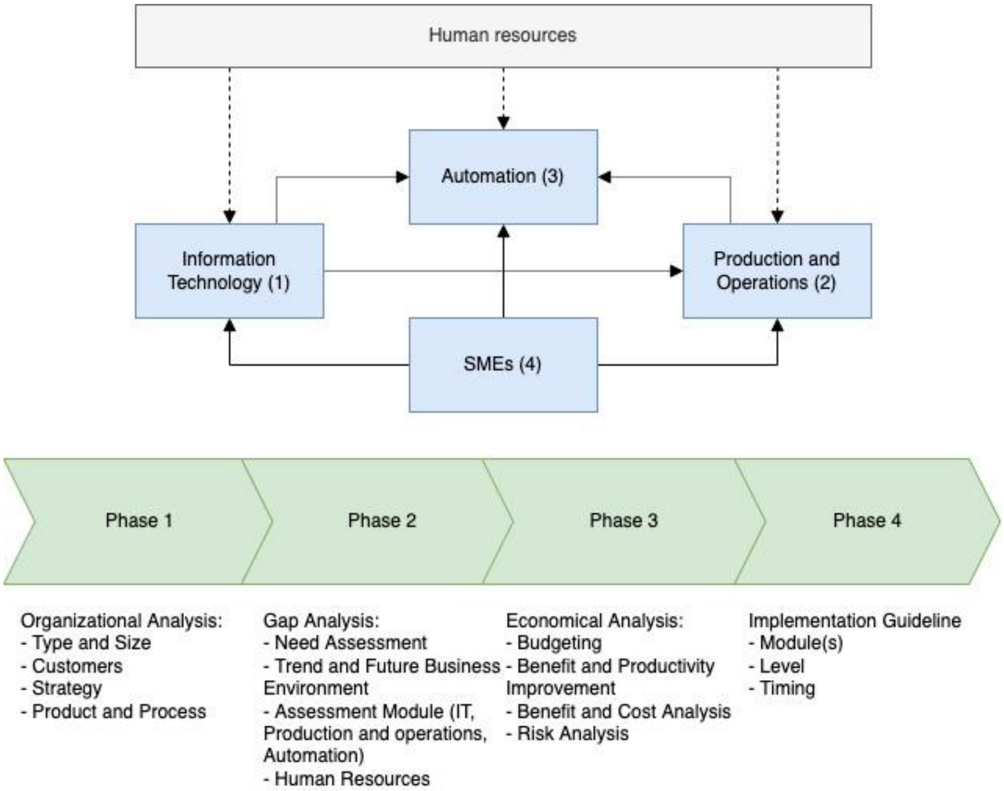


Figure 11: Implementation toolkit for introducing Industry 4.0 projects as proposed by Matt, Modrák and Zsifkovits (2020).

Several case studies exist in the literature as to the implementation of Industry 4.0 technologies in SMEs. Grieco *et al.* (2017) presented an application of Industry 4.0 technologies in an Italian luxury goods manufacturing company, with a focus on the production process throughout the supply chain. Bär, Herbert-Hansen and Khalid (2018) also focused on the supply chain in an SME while considering Industry 4.0 aspects.

We reported preliminary concepts for carrying out the project as well as the research that has been carried out in these different research areas. In the rest of this thesis, we will propose a dynamic allocation paradigm in the industry 4.0 manufacturing sector that will combine all these different research areas.

Chapter 2: Objectives and methods

The goal of this research is to propose and study new methods for the human resources allocation made possible through Industry 4.0 digitalization. This idea can be seen as a paradigm shift (Figure 12) from a static allocation from one operator to Y machines (middle in Figure 12) to a dynamic allocation from X operators to Y machines (bottom of Figure 12). Indeed, before the arrival of automation in the manufacturing sectors, each machine required the presence of an operator 100% of the time (top of Figure 12). However, this is not the case anymore. Currently, in the manufacturing industry, each operator is typically allocated to a set of machines (middle of Figure 1).

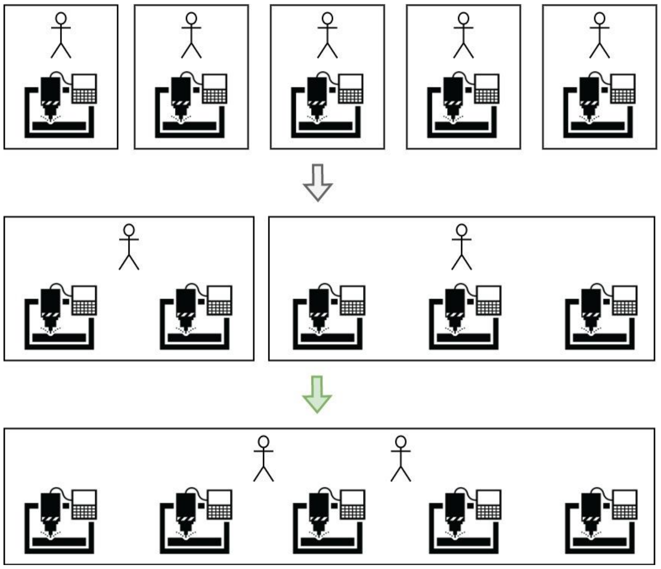


Figure 12: Paradigm shift

This dynamic allocation requires real-time data and interconnection of systems, which was not possible before the arrival of technologies that characterize the Industry 4.0 revolution. In order to evaluate the possible benefits of this paradigm shift, we conducted a case study with an industrial partner, APN. We aimed at providing insights into what this shift from a static allocation to a dynamic allocation could provide in terms of productivity and the implications of such a shift. We also wanted to provide the manufacturing sector with results using real industrial data in order to motivate and inform them about this possible shift to a dynamic allocation of human resources.

2.1 Objectives

The project's objectives are defined as follows:

1. Analyze a metal manufacturing system, focusing on the different tasks necessary for the production of metal parts
2. Define scenarios allowing real-time allocation of operators for such a system
3. Measure the impact of this form of allocation in terms of number of parts produced in a week and delays caused by operators in the production
4. Facilitate the implementation of the dynamic allocation method in the real world

2.2 Industrial case study presentation

This project was realized in collaboration with the manufacturing company APN. APN is a high-precision metal parts manufacturing company. Their principal sectors of sales are the aeronautical sector and the military sector. In Figure 13, we see different types of metal parts produced by APN.

Figure 13: Metal parts produced by APN, our industrial partner

APN employs some 135 employees located in their two factories in Québec City, as well as in their factory in California. As part of this project, we limited ourselves to the study of one of the factories in Québec City, located on Boulevard du Parc Technologique.

Our industrial partner is considered a leader of Industry 4.0 in Québec City. During the last ten years, they developed a solution in the form of a Computer Integrated Manufacturing (CIM) software. This solution, shown in Figure 14 (left), is the central piece that allows for integration and interconnection of all their systems. Employees, administrative as well as operators, use this solution constantly to support their work (right).

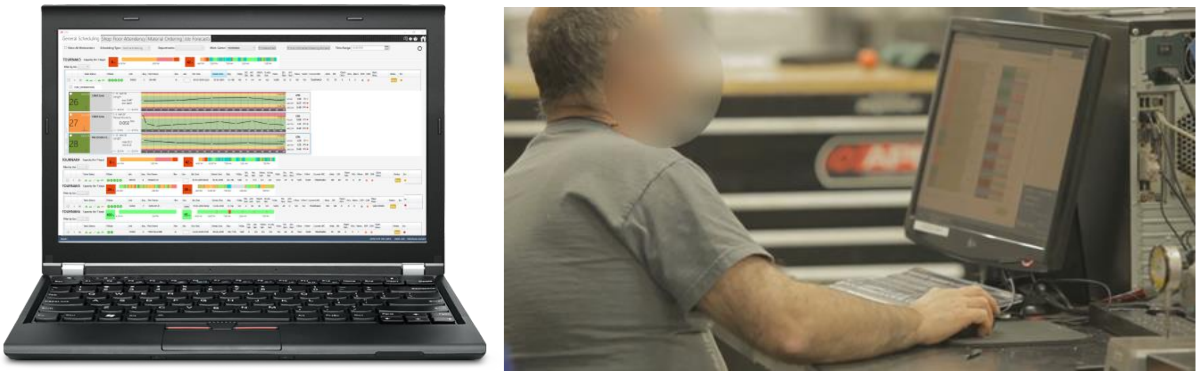


Figure 14: Computer Integrated Manufacturing (CIM) solution (left) and CIM in use by an operator (right)

APN also has high degrees of automation in the factory. For example, in Figure 15, we see a collaborative robot transporting a metal part (left) and a mechanical arm moving a metal part (right) when it exits a machine.

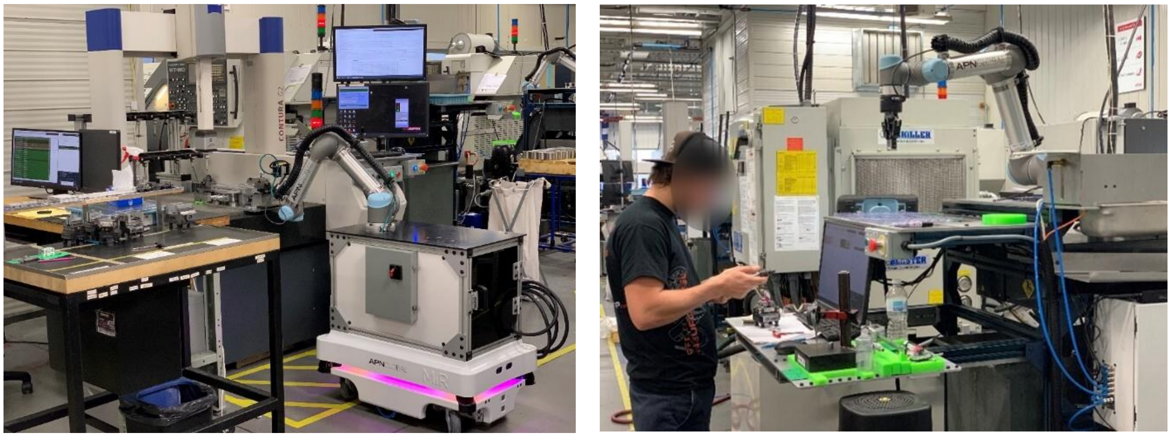


Figure 15: Robots in use in the factory

2.2.1 Metal production with CNC machines

The metal production line can include different steps, such as polishing and assembly. The main step, required for every single metal part, is to be machined on a Computer Numerical Control (CNC) machine. This step is the most important and central part in metal production since it allows for the parts to be produced. While all parts require to be processed on CNC machines, the other steps are all optional. CNC machines possess rotating tools that cut and shape the raw metal fed in order to produce a part according to its computer program.

The processing of parts on the CNC machine is the main focus of our project. In Figure 16, we see how this step is carried out. Raw material is provided to a CNC machine, which then automatically, with a computer program, produces a number of metal parts.

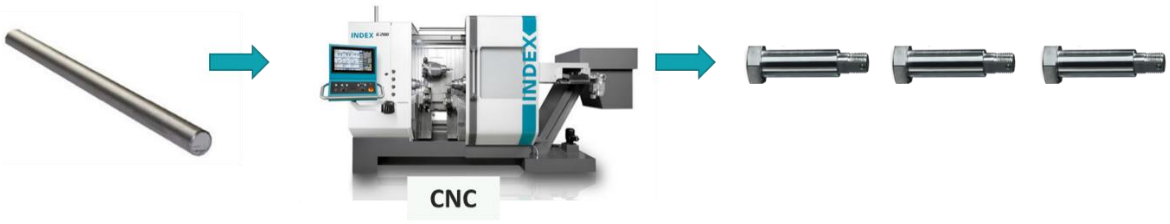


Figure 16: Metal parts production

2.2.2 Production sectors

Our industrial partner currently has 25 CNC machines in their factory located in Quebec City. While the production on each of these machines is similar to what is shown in Figure 6, there are some distinctions to be made and the CNC machines are divided into three sectors: the lathes, the milling machines, and the combitec machines. In Table 3, we show a complete inventory of the CNC machines at the beginning of the project. Italicized numbers account for available CNC machines including acquisitions by APN during the project. The simulation model experiments were performed on the initial number of CNC machines while the optimization model experiments were performed on the updated number of CNC machines.

Table 3: CNC machines inventory by production sector

CNC machines inventory		
Total: 22 (*25*)		
Lathes	Milling machines	Combitec
Total: 12 (*14*)	Total: 6 (*7*)	Total: 4
NAK-3	GROB-1	COMBITEC-1
NAK-4	GROB-2	COMBITEC-2
NAK-5	GROB-3	COMBITEC-3
NAK-6	GROB-4	COMBITEC-4
NAK-7	GROB-5	
NAK-8	GROB-6	
NAK-9	*GROB-7*	
NAK-10		
INDEX-1		
INDEX-2		
INDEX-3		
INDEX-4		
TRAUB-1		
TRAUB-2		

There are some distinctions between the machines present in the different production sectors. In Figure 17: Different types of CNC machines in each sector – lathe (left), milling machine (center) and combitec (right), we see an example of all three types of machines that are located in each production sector of the factory.

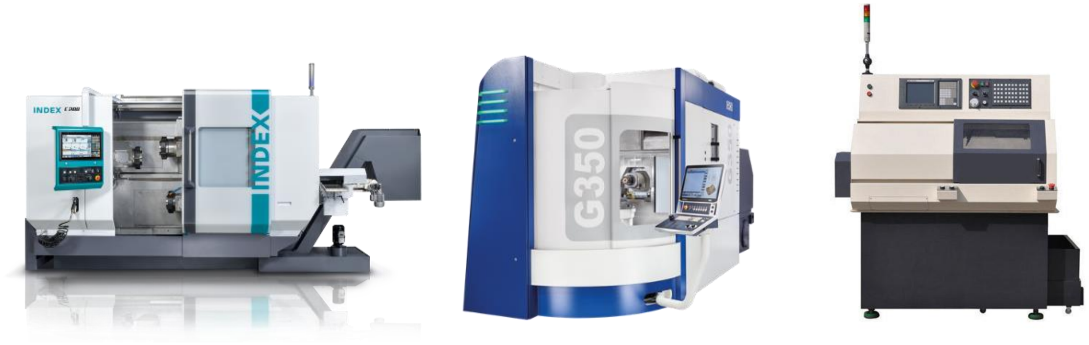


Figure 17: Different types of CNC machines in each sector – lathe (left), milling machine (center) and combitec (right)

2.2.2.1 Lathes

The biggest sector in the factory is the lathes sector, in number of machines, floor surface, and typical number of operators. CNC machines in this sector have a 4-foot bar feeder, except for the TRAUB-1, INDEX-1, and INDEX-2 machines which have an extra long 12-foot feeder. A mechanical arm is located at the exit of all the CNC machines in the lathes sector, in order to grab the machined metal part, clean it, and insert it in a plastic tray. This sector is the one with the highest degree of automatization.

2.2.2.2 Milling machines

The milling machines possess a palletizer and they do not have a bar feeder nor a mechanical arm. The palletizer in the CNC machine means that while a part is being produced, an operator can install the metal raw material for the next part. Indeed, in this sector, machines require a metal puck for each machined part since they do not have a bar feeder. This sector has a production process that is also a bit different. Parts may need to go through the machine multiple times in order to be completed. For example, a given part may need to be processed in three different steps. In order to model the machines in this sector, many hypotheses were required, such as limiting each part processing to a single step. Because of this particularity, production scheduling is typically the most difficult for this sector.

2.2.2.3 Combitec

Finally, the combitec sector is composed of small precision machines that require a metal puck for each part, but they do not possess a palletizer, contrary to the milling machines. This means that between each part, the machine needs to be opened to remove the machined part and install the next metal puck. Machines in this sector are also the oldest in the factory, which require experienced operators to work them.

2.3 Methods

The four phases of the project's methodology are presented in Figure 18.

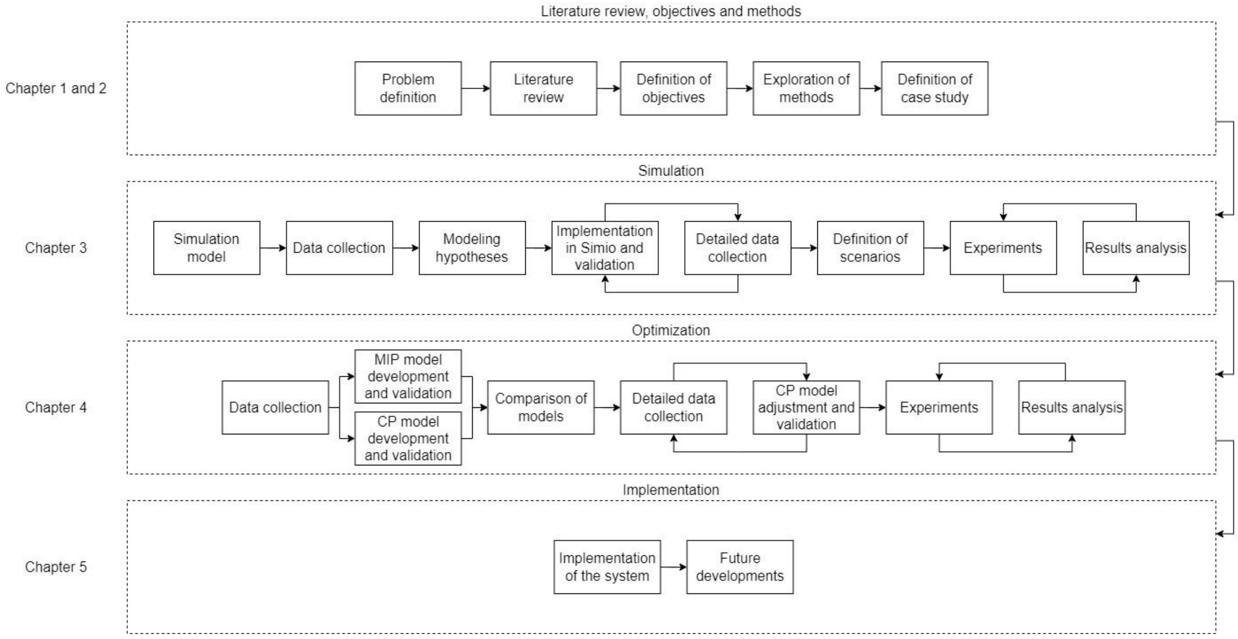


Figure 18: Methodology followed

2.4 Literature review, objectives and methods

At the beginning of the project, different meetings with the industrial partner and a visit of the factory led to the definition of the problem to investigate during the research.

Once the problem was defined, a set of keywords was established in order to conduct an exhaustive literature review. The literature surrounding the problem and the different methods used to solve it were explored and reviewed. Different scientific databases (Scopus, IEEE Xplore, ScienceDirect, Google Scholar, ResearchGate) were used, as well as popular conferences proceedings (Winter Simulation Conference, Manufacturing Modelling, Management and Control Conference). Abstracts of the articles corresponding to the keywords were read in a first iteration. If the abstract seemed relevant to our study, the article was read in a second iteration with a focus on the introduction and the conclusion sections. The articles that proved to be especially relevant to our study were read entirely during a third iteration.

With a definition of the problem and a review of the literature, we were able to define the objectives of the project, that were presented above. The objectives were established in order to advance scientific and industrial knowledge as well as help our industrial partner.

The different available methods to tackle the problem were then explored. Since the literature had been reviewed, we knew what methods were available to us. We decided to first simulate the factory processes. The second method chosen was to establish an optimization model. We knew that with a system as complex as ours, a discrete-event simulation technique could be an interesting way to study it thoroughly. We also figured that the allocation and scheduling problem could be tackled with an optimization model.

In order to better understand the environment and processes at our industrial partner’s site, many meetings were held with different resources, such as engineers, operators, directors, etc. During each meeting, the comprehension of the case study increased. Multiple observation periods with operators also took place during different working hours in the factory. This way, we were able to directly observe the different processes and refine the case study to investigate. The results are reported in chapter 3.

2.5 Evaluating the effects of dynamic allocation with discrete-event simulation

Once we had a good comprehension of the system and its processes, we began modeling the system in terms of simulation. Having worked with simulation before, we knew how to define the different aspects of the case study in order to include them in a simulation model.

The simulation model was presented to the R&D director at APN's, which helped clarify which data could be used for the project. Then, meetings with developers were held in order to access these data. Data were collected in a JSON format and validated in multiple iterations. Once data were reliable and validated, several transformations were required in order to use the data as an input in the simulation model (in CSV).

After collecting the available data, we were able to clarify and formulate modeling hypotheses. These hypotheses were all validated with the R&D director and the project team to have the most reliable representation of the system possible. Collaboration with a project engineer in charge of collecting processing times for the different operations was also required.

Then, the model was implemented in Simio. Simio was chosen for its interesting visual representation that would allow us to show the model to the industrial partner. Implementation in Simio was coupled with the validation of the model. The model was validated by presentations and interviews with different process engineers and the R&D director of the company.

When errors or imprecisions were detected in the validation phase, we reviewed and modified the data with developers. These multiple iterations of validation and detailed data collection led to a precise model that represented closely the system of production under study. Then, a final detailed data collection was executed, ready for the different experiments.

In order to reach our objectives, three scenarios were defined for the simulation: one representing the current situation in the factory (the status quo), another one allowing for operators to be dynamically allocated to tasks based on their current skill levels, and a third one with dynamic allocation and perfectly polyvalent operators.

Once we had our data and scenarios, we were able to run different experiments in Simio. Experiments represented a challenge since they were resource-intensive for the computer. We had to proceed to a change of computer in order to be able to run all the experiments and replications required.

The results obtained in the experiments were analyzed individually and then additional experiments were executed, in order to push the analysis further and provide different performance indicators.

2.6 Proposing an optimization model for the real-time allocation of tasks to operators

Having already collected data as part of the simulation phase of the project, we had a starting point for the optimization model. Different and more specific data were nonetheless required for certain parts of the optimization model.

Two models were developed, one using mixed-integer programming (MIP) and another one using constraint programming (CP). Lindo was the tool employed for the MIP model and MiniZinc was the one exploited for the CP model. The models were validated by interviews with the industrial partner, similarly as with the simulation model.

Experiments were conducted in order to compare the performance of the two models. The conclusion was that the CP model outperformed the MIP model.

Once the model had been chosen, further data were necessary in order to execute a real-time optimization model. We collected data from additional data sources (e.g., the CNC machines) with the support of the APN’s development team.

Iteratively, the CP model was adjusted to fit the additional input data received. Every time the model was modified, further validation was necessary. Once we knew the collected data and the CP model were valid and accurate, an official data collection was executed for the experiments.

Experiments were conducted to test and compare the same scenarios defined for the simulation phase of the project.

Once the results were analyzed, an additional experiment was put in place in order to provide further analysis concerning the different scenarios investigated.

Finally, the implementation of the real-time task allocation system began. An implementation roadmap was put in place and presented to the industrial partner. Different aspects of the implementation were discussed. A first artefact was provided to the industrial partner.

While this thesis in being written, tests on the production floor are about to begin. The next steps have already been defined by our project team.

Chapter 3: Evaluating the effects of dynamic allocation with discrete-event simulation

In order to analyze the impact the proposed paradigm shift might have on the manufacturing job-shop, we developed a discrete-event simulation model. It represents the production floor of our industrial partner. Firstly, we needed to model the different tasks and processes (Section 3.1). This allowed us to request the necessary input data in order to run experiments with the simulation model. The model was verified with experts in the shop. Then, the model was implemented using *Simio* simulation software (Section 3.2). A validation phase was also conducted based on some guided hypotheses. Following the validation, we were given access to additional data regarding the scheduled jobs as well as the production floor layout. These new data were used in a simulation experiment. In the experiment, three different scenarios are compared regarding how they affect the number of parts that can be produced in a week. The first scenario uses static allocation (1), the second uses dynamic allocation with their current set of skills (2), and the third one uses dynamic allocation with polyvalent operators (3). These aspects will be presented in Section 3.3.

3.1 Model

While most simulation models focus on product flow, our need was to model sequences of *tasks*: some performed by operators, some by operators and machines, others simply by machines. A classic simulation model approach is to have *entities* representing the physical products/items moving from one machine to another. Since our aim was heavily reliant on the tasks surrounding these parts, this type of modeling would have required a lot of workarounds to correctly model the various behaviours of interest throughout the production process. Instead, we opted for an alternative modeling approach. In our model, an *entity* represents a single task. This task might need to be performed by an operator, by a machine, by a coordinates measurement machine (CMM) or by an optical comparator (CO). Combinations are also possible, where an operator is needed in addition to another type of resource.

A conceptual representation of the model is given in Figure 19: Simplified conceptual representation of the simulation model. The plant currently has 22 CNC machines (e), 31 operators with different sets of skills (d), four measurement machines (f) including two coordinates measurement machines (CMM), one automatic optic comparator, and one manual optic comparator.

From the APN production schedule (a), we could extrapolate a predictable sequence of tasks to perform: parts machining (blue), measurement (red), and others (yellow). Parts machining is executed on CNCs. Simultaneously, other tasks may be accomplished by an operator on a worktable (b), although some require the CNC to be inactive. After leaving the workcenter (e), some parts need to be measured for quality control. This is a complex process that may involve different measuring instruments and measurement machines. If a measure is outside the specification limits, we begin a non-conformity process and a corrective action task (e.g., tool change) might be created and inserted in first place into the waiting queue of tasks.

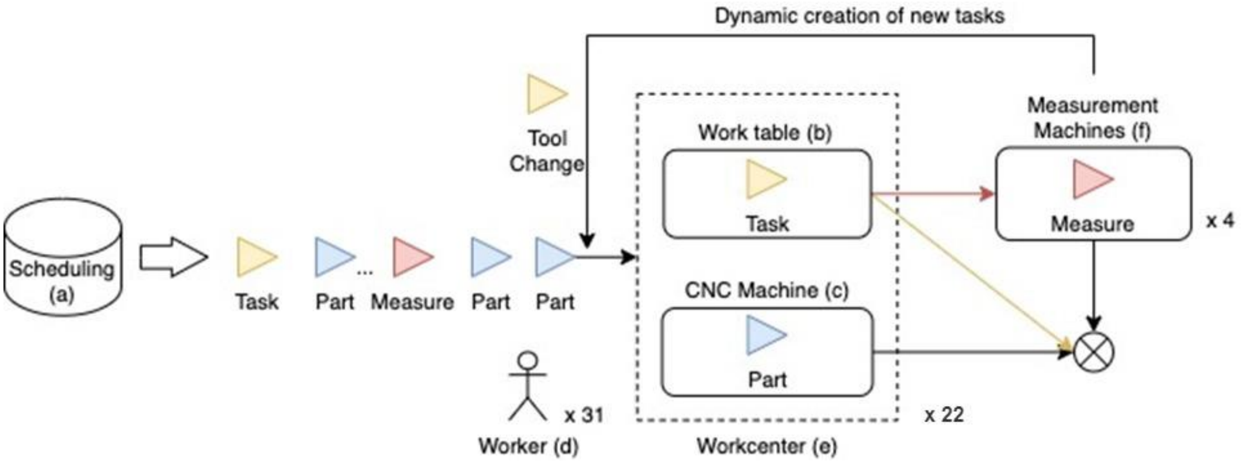


Figure 19: Simplified conceptual representation of the simulation model

3.1.1 Task types

Because the main focus of the model is on the different tasks at hand, this subsection summarizes the main types of tasks that may occur in the workcenter (e).

*Table 4: Task types and their predictability and required resources. *The tasks requiring an operator may also require measurement machines.*

	Predictable		Non-predictable
	With schedules	With frequencies	With probabilities
Operators*		Parts measuring	Offset (corrective action) Parts measuring (backtracking/forwardtracking)
Operators + CNC machines	Setups	Tool changes Raw material addition	Tool changes (corrective action)
CNC machines	Parts machining		

We will first describe the *predictable* tasks such as parts machining, tool changes, raw material addition, setup, and parts machining. Then, the *non-predictable* tasks which occur in the non-conformities process will be presented.

3.1.1.1 Parts machining

Parts machining is considered the most important task since it has the most direct impact on the company's productivity. Nevertheless, in the context of this study, we are more interested in the surrounding tasks since parts machining does not need an operator to be performed. Indeed, once their setup is over, CNC machines may machine parts on automatic mode. Parts machining tasks require a certain processing time, depending on the complexity of the product being produced. The product's specifications are directly linked to the job in progress on the CNC machine. Parts machining is included in the simulation model in order to pace the other tasks requiring operators. It is also useful in order to extract performance indicators.

3.1.1.2 Tool change

Tool changes consist of removing one of the tools placed inside the CNC machine with a new one. Each CNC machine uses different tools to machine the metal. Since tools are used to remove metal matter on the metal parts, they can only be used for a certain number of parts before they are worn out and lose their initial shape, which would result in parts outside the specification range. Therefore, the tool changes occur according to a certain part frequency depending on the current job on a given workcenter. For example, a certain tool may require to be changed after every 10 parts machined on the CNC machine. The part frequencies are determined by the quality control department and are known in advance when a job begins.

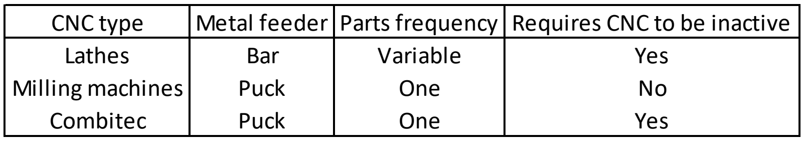
Tool changes may also occur after an unexpected event occurs. For example, if a tool breaks during production, it needs to be changed immediately, before resuming the machining process.

When a tool change occurs, the CNC machine must be stopped so it can be opened for the old tool to be removed and the new one to be added.

3.1.1.3 Raw material addition

Metal must be provided to the CNC machines in order to produce parts. Our industrial partner possesses two different types of machines for raw material addition: one that uses a metal bar with a feeder to produce many parts per bar, and another that needs a metal puck for each part. For the first type of machine, the length of the machined part is known, as well as the length of the metal bar. There is also a certain quantity of metal lost, which we call the cut-off, and its length is also known. With this information, we are able to calculate the part frequencies at which a new metal bar must be provided to a CNC machine. For the second type of machine, we simply use a frequency of one, since after every part, a metal puck must be provided to the machine. When raw material addition occurs, the CNC machine must be stopped so it can be opened for the material to be added. However, there is an exception to this rule. A certain type of machine, the milling machines, possesses a pallet system that allows for a raw material puck to be added while the machine is working. Table 5 shows the characteristics of the raw material addition for each of the three types of CNC machines.

Table 5: Raw material addition characteristics for the different CNC machine types



CNC type	Metal feeder	Parts frequency	Requires CNC to be inactive
Lathes	Bar	Variable	Yes
Milling machines	Puck	One	No
Combitec	Puck	One	Yes

3.1.1.4 Setup

The setups are the longest tasks. Although in practice they are composed of multiple subtasks, they were modeled as one long task since their total processing time is known by our industrial partner and typically allocated to a single operator. Alternatively, the multiple subtasks could also be modeled separately if more data were available for each one. A setup consists of preparing a machine for a new job. It occurs between different jobs. Setups often last many hours. In a setup, tools must be prepared and changed, computer programs must be loaded in the machine, and then the first parts produced need to be controlled at 100% before letting the machine run on automatic mode. Setups are always the first task to occur when a new job arrives. The order of the jobs to arrive is known, as the priorities are determined by the scheduling team, which directly gives us the setup task.

3.1.1.5 Parts measuring

Parts measuring is the most complex task to model, and also the most critical one from the point of view of the industrial partner. Indeed, since their parts are used in the aeronautical sector, an elaborate quality control system is used with many processes, machines, and instruments. Every type of part is characterized by many defining characteristics, for example the diameter of a cylindrical part. Every characteristic of a machined part must be measured with frequencies determined in advance by the quality control team. For example, a given part might have up to a hundred different characteristics that each need to be measured at certain frequencies. The quality control team normally tries to determine frequencies that have a common multiple, so that when a given part is being controlled, many of its characteristics are measured, minimizing the number of distinct parts that need to be controlled for different characteristics. Some characteristics might need to be measured every 6 parts, while others need measuring every 12 or 24 parts. Some parts even have extremely critical characteristics that must be measured on every single part (so with a frequency of one).

Adding to the complexity of the system is the fact that different characteristics need to be measured using different instruments, or machines. Many characteristics are measured manually using instruments such as a caliper, while others are measured on CMMs or optical comparators. Manual measures are taken directly at the workcenter, on a table next to the CNC machine. Every table possesses the necessary instruments to measure the parts on the current job. A given parts measuring task may then include multiple different characteristics, meaning that it may need to be processed at up to three different places (the worktable, an optical comparator, and a CMM).

The two following tables show how the quality control is planned for a given job producing a certain part. In Table 3, we see all the characteristics (A through E) that need to be controlled (measured) on this part number (A7D2J4), their measuring frequency in terms of number of parts, as well as the tool required to conduct the measure. In Table 4, we see how this quality system is translated into tasks for a job producing 10 of those parts. Since D has a measuring frequency of one in this example, we need to measure certain characteristics for every part. Typically, not every part needs to be measured, although, a critical characteristic that needs to be controlled for every part may occur in real life. MOC stands for *Manual Optical Comparator* and AOC for *Automatic Optical Comparator*. Manual represents all manual tools that operators may use on the worktable. Table 6 data, provided by our industrial partner, is converted to Table 7 data when generating the list of predictable tasks, before the simulation begins.

Table 6: Characteristics measured on a given part with their measuring frequency (in number of parts) and the required tool to take the measurement.

Part A7D2J4		
Characteristics	Measuring frequency	Tool
A	2	CMM
B	4	MOC
C	2	AOC
D	1	Manual
E	3	Manual

Table 7: Example of a sequence of machined parts on a given job with their characteristics that require to be measured as well as the list of tools required to take the measurements.

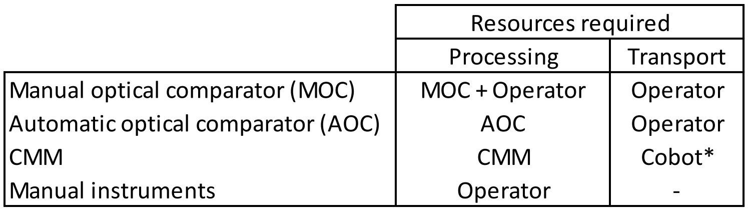
Job A7D2J4-1		
Part number	Characteristics measured	Tools required
1	D	Manual
2	A, C, D	CMM, AOC, Manual
3	D, E	Manual
4	A, B, C, D	CMM, MOC, AOC, Manual
5	D	Manual
6	A, C, D, E	CMM, AOC, Manual
7	D	Manual
8	A, B, C, D	CMM, MOC, AOC, Manual
9	D, E	Manual
10	A, C, D	CMM, AOC, Manual

The processing times for the characteristics measured on the CMM comes from the company's database. The data needed to be adjusted since the processing times recorded by the CMM were from a set of characteristics and not for a specific characteristic. We needed to match the sets of measures from the tasks in the simulation with a set of measures recorded in the CMM database. When no match was found in the database (meaning that this particular set of characteristics was measured for the first time), processing times were extrapolated from similar sets of characteristics.

The processing time for the characteristics measured on the optical comparator as well as with manual instruments were obtained from a previous project regarding the automatization of the measuring process in which the individual time to measure a characteristic with each of the different instruments was approximated by the engineering team.

The handling of parts between stations (transport) is included in the measuring process. While the worktable is directly next to the CNC machine as part of a single workcenter, the measuring machines are shared between all CNC machines in the factory and are located remotely from the CNC machines. Table 8 shows the *resources* required for the processing and the transport of the different tasks.

*Table 8: Resources required for the different types of parts measuring machines/instruments. *Collaborative robot.*



	Resources required	
	Processing	Transport
Manual optical comparator (MOC)	MOC + Operator	Operator
Automatic optical comparator (AOC)	AOC	Operator
CMM	CMM	Cobot*
Manual instruments	Operator	-

The manual optical comparator needs to be reached on foot by the operator wanting to measure its part. Since the measure is taken manually with this machine, the operator must move to the optical comparator with the part, wait in line if the optical comparator is not currently available, measure the part with the optical comparator, and return to the original workcenter with the measured part.

The automatic optical comparator also needs to be reached on foot, but once the measuring step starts, the operator may leave the station to do something else in the factory. The operator must return to pick up the part once the measuring process is over and bring it back to the original workcenter.

The parts measured at the CMM need to be transported with a special collaborative robot (*cobot*). When a part is ready to be measured, the operator places it in a vice in a special location next to its machine, and the cobot comes and picks it up, transports it to the CMM machine, and brings it back once it is measured. The operator must then remove the measured part from the vice and place it with the other completed parts. Every machine has its own pickup location, except for the Combitecs. The Combitecs only have one common location where the vice with the part must be installed before being picked up by the cobot. The operator must then move the part from the workcenter to the pickup location and move it back to the workcenter from the pickup location once it is back from the CMM.

3.1.1.6 Tasks generated by the non-conformities management process

Simulation to analyze processes that has some stochastic effects. In our case, stochasticity comes directly from the fact that parts measured may be outside the specification range. When we determined a value is out of specifications, we start a process dealing non-conformities, which may generate additional tasks for the operators.

Figure 20 presents the non-conformity management process. A non-conformity occurs when a given characteristic is out of the specified range. The quality system of the company has two different triggers, *warning* (yellow) and *critical* (red). A non-conformity may issue only a warning (yellow), meaning that the machined part will still be accepted by the client, but a corrective action needs to be taken to bring the next parts back to the *accepted* range (green). A non-conformity may also be critical (red), meaning that the machined part needs to be discarded. In this case, previous machined parts must also be measured to ensure that they are acceptable with a process called *backtracking*. In backtracking, the problematic characteristic must be measured on some previous parts, given that they were not previously measured, until the characteristic is yellow or green. Following the backtracking (if the characteristic was in the red zone), or immediately after the characteristic is measured (if it was in the yellow zone), a corrective action must be taken. Different corrective actions exist, the two most frequent being a premature tool change or an offset, which are the ones modeled in the simulation model. Following this action, the given characteristic must be measured on the next machined part under a process called forward tracking. If the machined part’s characteristic is in the green zone, the non-conformity is considered dealt with, and the automatic production may resume. If the machined part’s characteristic is either in the yellow or red zones, it means that the corrective action was not successful, and we must correct the machine once again. We execute this process until we are able to produce a part with its characteristic in the green zone.

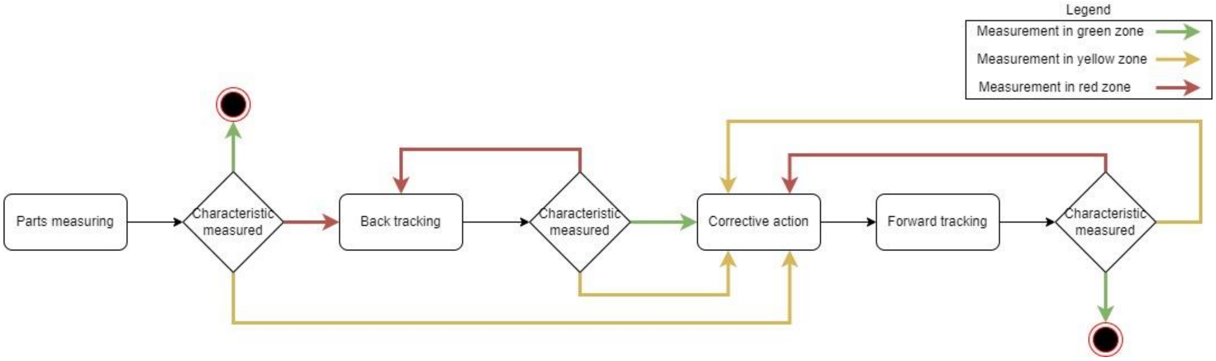


Figure 20: Non-conformity management process

3.2 Implementation in Simio simulation software

In our project, Simio was chosen for its visualization capabilities. It has been used in other publications such as in Munasingha and Adikariwattage (2020) to model passenger processing at an airport and in Costigliola *et al.* (2017) to model quality control laboratories in the pharmaceutical industry.

In order to understand the implementation of a model in Simio, some basic concepts are defined in Table 9.

Table 9: Basic concepts of Simio simulation software as defined by Pegden (2009).

Name	Definition
Entity	Base object.
Source	Creates entities that arrive to the system.
Sink	Destroys entities and records statistics.
Server	Fixed object that models a service process with input and output queues.
Node	Entry and exit point of a server.
Resource	Models a resource that can be used by other objects.
Vehicle	Carries entities between fixed objects.
Connector	A zero-time connection between two nodes.
Path	A pathway between two nodes where entities travel based on speed.
Add-on process	Set of user-defined actions that take place over time that may change the state of the system.

Figure 21 shows the entire factory represented in the simulation model as implemented in Simio. Scheduling (a) is the *source* where all the initial *entities* are created. They are created following an exhaustive data table, that is imported in Simio at the beginning of the simulation run. The data table will be presented in detail in the section Input data. They are routed to the corresponding workcenter (e) depending on their properties. Workers are included in the model with their full name, schedules, and skills. Task entities are routed to the appropriate measurement machine depending on their initial data, following some *add-on processes*. As stated, the different *entities* created in Simio represented the different tasks (g) of interest that could occur during production. The four measurement machines (f) are also included in the simulation model. Finally, the cobot (h) is represented by a transporter that can transport parts from their pickup to the CMMs and back. *Entities* may only move in the simulation model when transported either by a cobot or an operator, with the exception of when they are initially created by the *source* and routed to their workcenter. In the model, operators act as *resources*, as well as *vehicles*.

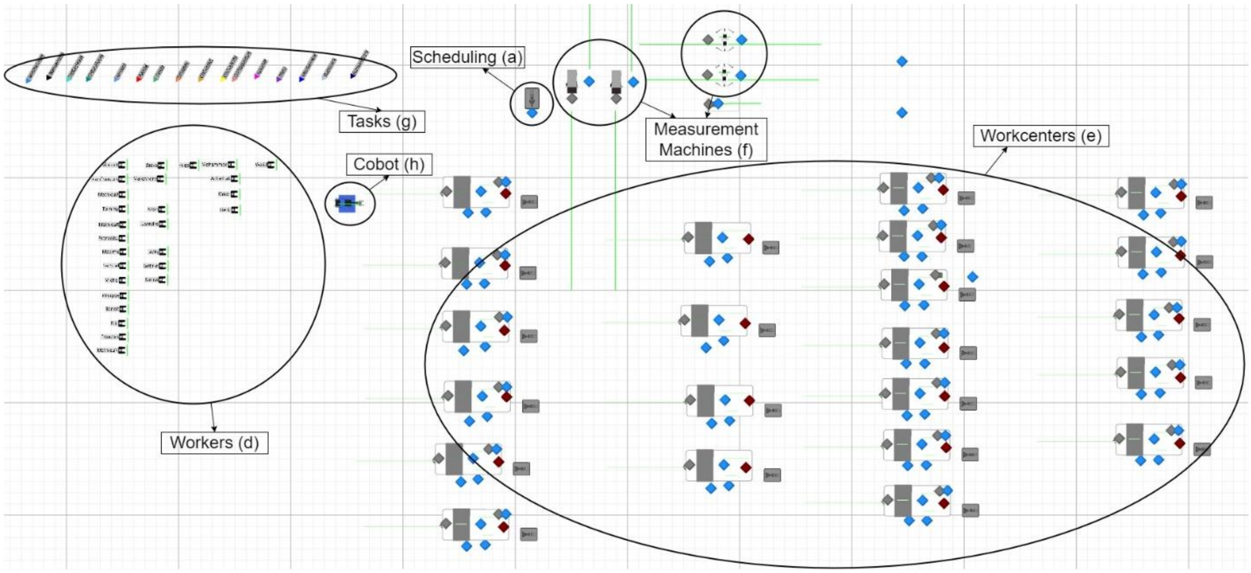


Figure 21: Illustration of the complete factory simulation model as represented in Simio

The factory simulation model includes multiple workcenters that behave in a similar way. Workcenters (e) were created as sub-models that is then duplicated. An illustration of a single workcenter is presented in Figure 22. The workcenter model has two *servers*; one (SrvMachine) representing the CNC machine (c), and another (SrvPoste) representing the worktable (b) where manual measures may be taken. For the rest of this section, the SrvPoste will be referred to as the worktable.

Task *entities* enter the workcenter model and are directly routed to the input *node* of the CNC machine. Depending on their data, they are next routed either directly to the *sink*, to the worktable *server*, or to an external *server* in the complete simulation model that corresponds to a measuring machine.

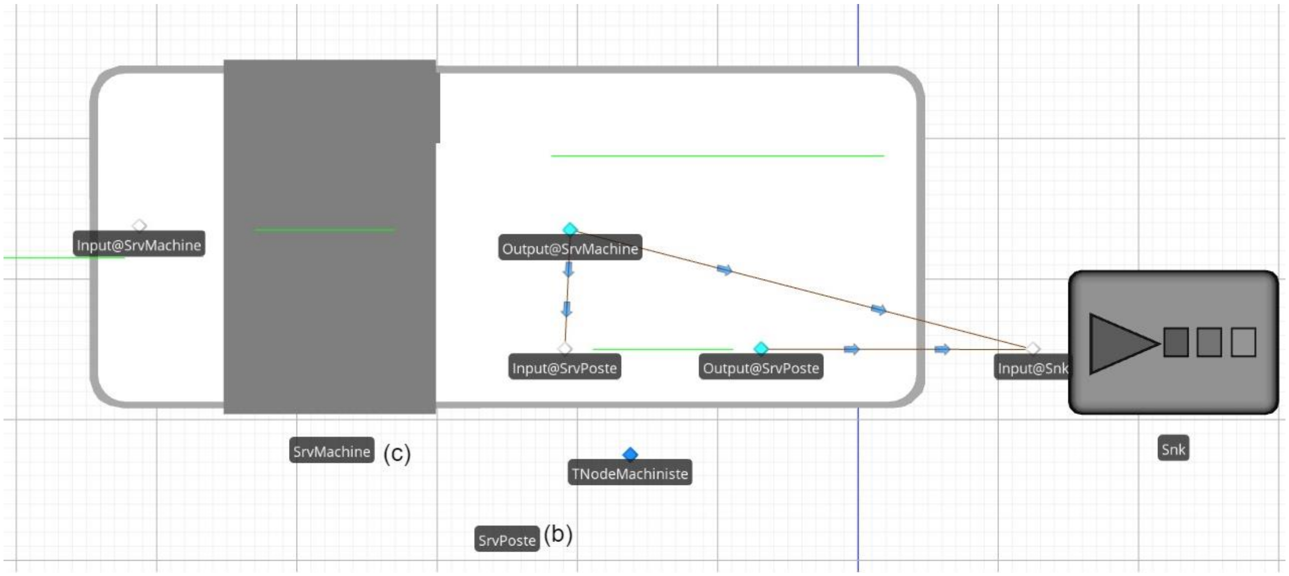


Figure 22: Illustration of the workcenter simulation model as represented in Simio.

Since the simulation model was intended to be used by higher management, it was designed to be as representative of the factory as possible. An appealing 3D model view was developed. In Figure 23, we can see the view of workcenters that possess a worktable and a CNC machine on top, as well as the different measuring machines and even the cobot. Since tasks are an abstract concept, it was decided to leave them as triangles icons so as not to mix up the users of the model into thinking that they represent metal parts.

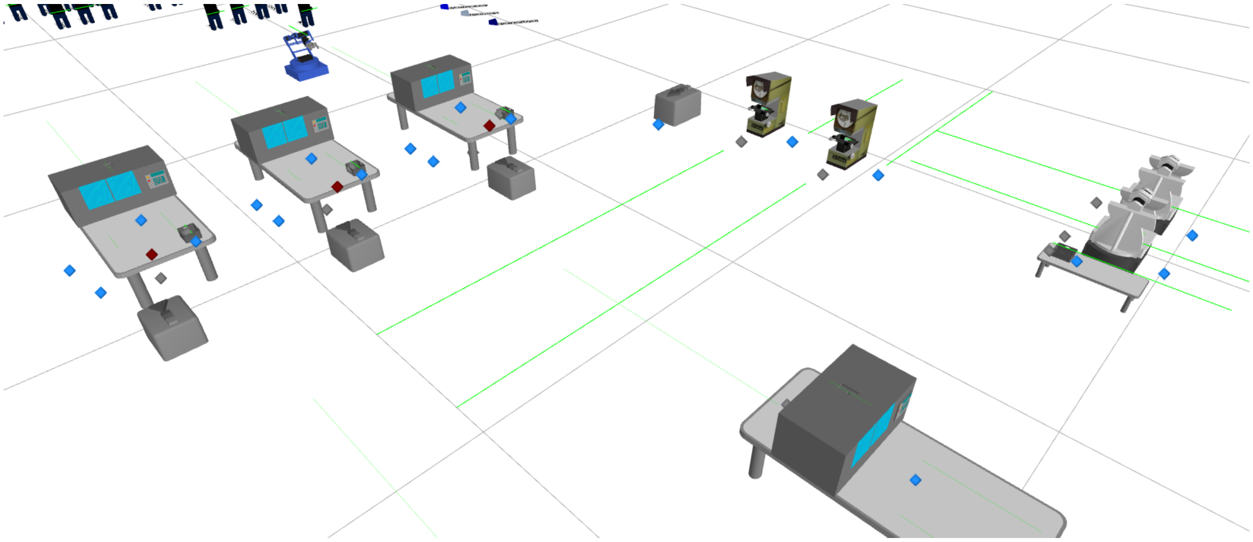


Figure 23: Illustration of the complete 3D factory simulation model as represented in Simio

3.2.1 Modeling assumptions

Some assumptions needed to be made in order to simulate the production floor. Different parameters chosen in Simio are also listed here.

- The walking speed of all operators was set to 1.4 meters per second.
- Operators select the next available task located at the smallest distance from their current position once they become available or a task is released.
- When off shift, operators in the simulation “wait” at a given *node*, common for all operators.
- When idle, operators wait where they were located when their last task ended.
- The cobot moves at a speed of 0.5 meter per second, corresponding to its real speed.
- The cobot may only transport one *entity* at a time.
- The cobot selects the task that is the first in queue. In reality, the cobot uses a complex algorithm to decide which part should be measured next, but that was not implemented in the simulation model.
- If an operator is measuring a part on the optical comparator when its shift finishes or he begins his break, it must first complete the task started.
- If an operator is processing a task on one of the workcenters’ *servers* (the worktable or the CNC machine) when its shift finishes or he begins his break, the task can be allocated to another operator (switch *resources* if possible).
- The worktable *server* has an infinite capacity, meaning that many operators may execute tasks at the same time on this *server*.

3.2.2 Input data

Task entities generated by the *source* are created following an input data table. This table contains all the required properties for each *task entity*. All the properties of the input table are as follows:

- *Row*: An incremental ID in order to reference a task when needed in the model.
- *Job*: The job number associated to a task.
- *Workcenter*: The workcenter on which the task must be processed.

- *ArrivalTime*: All tasks arrive at the beginning of the simulation and are placed in order following their priority, so that a given CNC machine never runs out of tasks to process.
- *TypeEntity*: Represents the type of tasks. Multiple types of tasks are implemented in the simulation model but only some are created for the current experiments.
- *Qty*: The *source* of the Simio model produces the *entities* following the table, with a quantity determined in the Qty column. The quantity is either 1 or 0 and represents the initial quantity that the *source* must produce. For every deterministic task, the quantity is of 1. For tasks that may occur depending on a stochastic process, the quantity is 0.
- *IDEntity*: Tasks all relate to a given part number. The IDEntity corresponds to the part number. Many tasks have the same IDEntity when they relate to the same part.
- *Priority*: Some task types have higher priorities than others. For example, a tool change has a higher priority than a part measuring.
- *TimeCNC*: Processing time of the task on the CNC machines. Can be 0 if the task does not require the CNC machine.
- *TimeWorktable*: Processing time of the task on the worktable. Can be 0 if the task does not require the worktable.
- *TimeCMM*: Processing time of the task on the CMM. Can be 0 if the task does not require the CMM.
- *TimeACO*: Processing time of the task on the automatic optical comparator. Can be 0 if the task does not require the automatic optical comparator.
- *TimeMCO*: Processing time of the task on the manual optical comparator. Can be 0 if the task does not require the manual optical comparator.
- *NeedOperator*: If the task is fully automated, this field has the value 0 since it does not require an operator. If the task requires an operator, this field has the value 1.
- *IsMeasure*: If the task is a measuring type task, this field has the value 1, and it has the value 0 otherwise. This field exists since many subtypes of measuring tasks exist (e.g. a measuring task in the backtrack context) and we need to distinguish all the measuring tasks.
- *NeedProcessingWorktable*: If the task has a processing time on the worktable that is not null, this field has a value of 1, it has a value of 0 otherwise. This field helps in routing the *entities* to the correct *server* in the simulation.

- *ProbGenToolChange*: This field contains the probability of an *entity* generating a tool change task once it is completed. It is used in the context of the non-conformity process when a corrective action is required.
- *ProbGenOffset*: This field contains the probability of an *entity* generating an offset task once it is completed. It is used in the context of the non-conformity process when a corrective action is required.
- *ProbGenBacktrack*: This field contains the probability of an *entity* generating a backtrack measuring task once it is completed. It is used in the context of the non-conformity process when a corrective action is required.
- *InputMachine*: This field has a reference to the input *node* of the corresponding workcenter's machine *server*. It is useful in routing the *entities*.
- *InputSink*: This field has a reference to the input *node* of the corresponding workcenter's *sink*. It is useful in routing the *entities* correctly.
- *PickupCMM*: This field has a reference to the input *node* of the correct pickup location. The pickup location is modeled with a *server*. All workcenters possess their own pickup locations with the exception of the Combitecs who all have the same common pickup location.
- *SrvCMM*: When there are two CMM available, parts may only be measured on one of them depending on the available probes (measuring tools) currently installed on the CMMs. This field has a reference to the CMM where the part must be routed.
- *InputCMM*: This field has a reference to the input *node* of the corresponding CMM *server*. It is useful in routing the *entities* correctly.
- *CMMInputBuffer*: This field has a reference to the input buffer of the corresponding CMM *server*.
- *CMMOutputBuffer*: This field has a reference to the output buffer of the corresponding CMM *server*.
- *ListOperatorsTransporter*: This field lists the operators that may transport the *entity*. The list of operators in this field will change depending on the experimental scenario.
- *ListOperatorsObject*: This field lists the operators that may process the *entity*. The list of operators in this field will change depending on the experimental scenario.

- *WCMachine*: This field has a reference to the CNC machine *server* of the corresponding workcenter.

Data validity was ensured by screening the collected data with experts of the company.

3.2.3 Modeling system behaviours with *add-on processes*

Different Simio *add-on processes* were developed to model particular model behaviours, as described below.

3.2.3.1 Routing tasks

One complex aspect of the simulation model is to route the task *entities* correctly. While some simple tasks such as parts machining only need to be processed on the CNC machine *server* before ending up in the *sink*, other tasks need to go directly to the worktable while some need to travel in the factory to be processed on different measurement machines (those tasks represent a part moving in the factory).

A simple method is used inside the workcenter to route the *entity* depending on whether it needs to be processed on the worktable or not. The *connector* towards the worktable *server* uses the binary value of the *NeedProcessingWorktable* field. *Entities* move towards the worktable if the binary value is equal to one.

A Simio *add-on process* named *DecidingDestination* was implemented in order to route the *entities* correctly once they exit the workcenter (Figure 24). This process is triggered each time an *entity* enters the output *node* of the workcenter. Many different Simio objects may enter this *node*: an *entity* that exits the process, as well as a *vehicle* (i.e., an operator or a cobot) coming to pick up the *entity*. The process verifies whether what just triggered the process is a *vehicle*, in which case it is ignored. Otherwise, if the object is an *entity*, then it enters into the decisional part of the process. Depending on which *server* the *entity* needs to visit, which corresponds to processing times on these *servers* that are not null, the *entity* will be routed to the next *server* it has not yet visited. *Entities* hold a variable that keeps track if a *server* has been visited or not (see previous section Assigning values to tasks). If an *entity* is not a measuring task or once all *servers* that needed to be visited have been visited, it can then be routed to the *sink* and disappear from the simulation model.

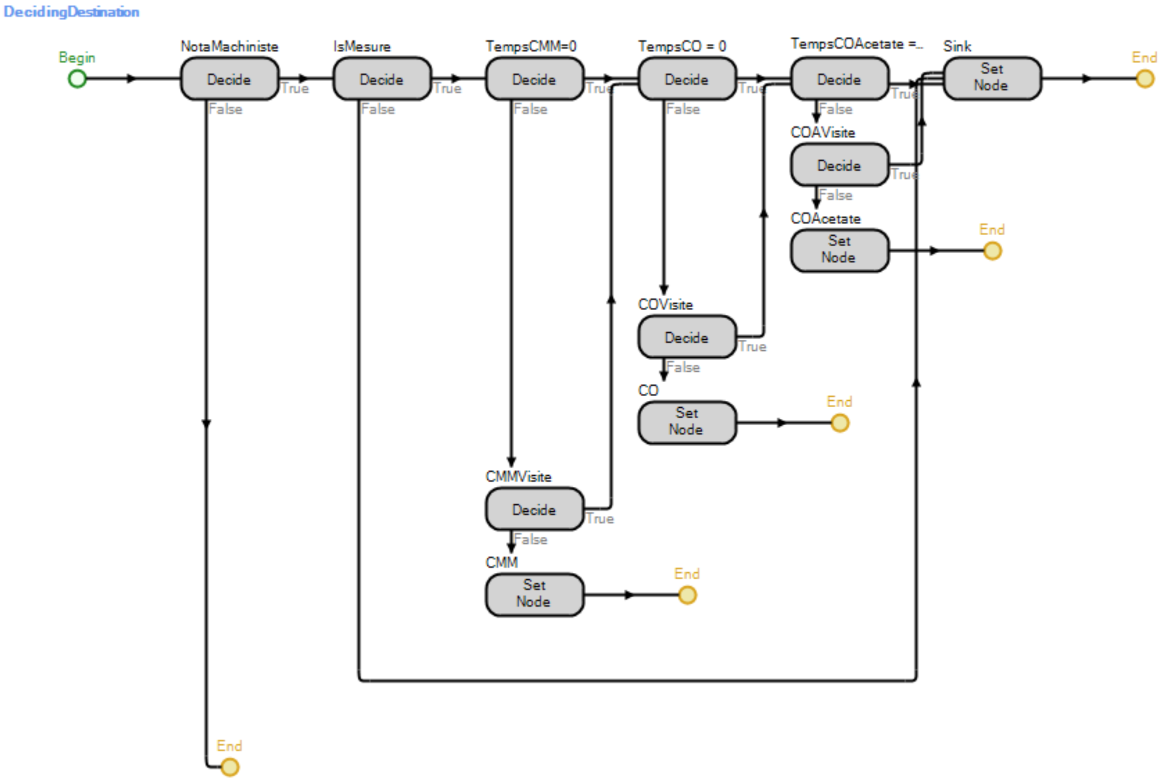


Figure 24: DecidingDestination for the workcenter add-on process. This process is triggered when the output node of the workcenter is entered and routes the entities to the correct server.

Finally, the pickup stations use a similar process in order to correctly route the *entities* to and from the CMM machine. If the CMM machine has not yet been visited, it is routed there, otherwise, the *entity* is routed to the *sink*, riding the cobot. Figure 25 shows the implementation of this process in Simio.

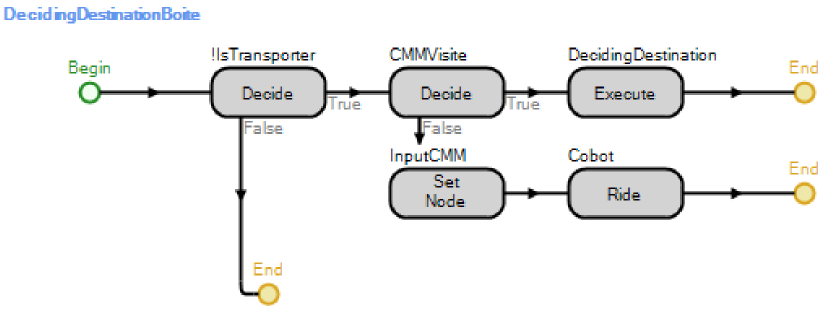


Figure 25: DecidingDestination for the pickup station add-on process. This process in triggered when the output node of the pickup station is entered and routes the entities to the correct server.

3.2.3.2 Generating additional tasks

The probabilities (*ProbGenToolChange*, *ProbGenOffset*, *ProbGenBacktrack* in ***3.2.2 Input Data***) of a part being outside the specification range comes directly from APN historical data. In Simio, the non-conformity process is triggered when an *entity* reaches the input *node* of the *sink* in its workcenter sub-model. This process randomly decides if a machined part meets criteria or not.

The process to create tasks is crucial to appropriately represent the stochastic aspect of unexpected tasks happening at stochastic moments. The different types of tasks that may be created are the forward tracking parts measuring, the backtracking parts measuring, as well as two types of corrective actions: an offset and a tool change. In Figure 26, we see what this *add-on process* looks like in Simio.

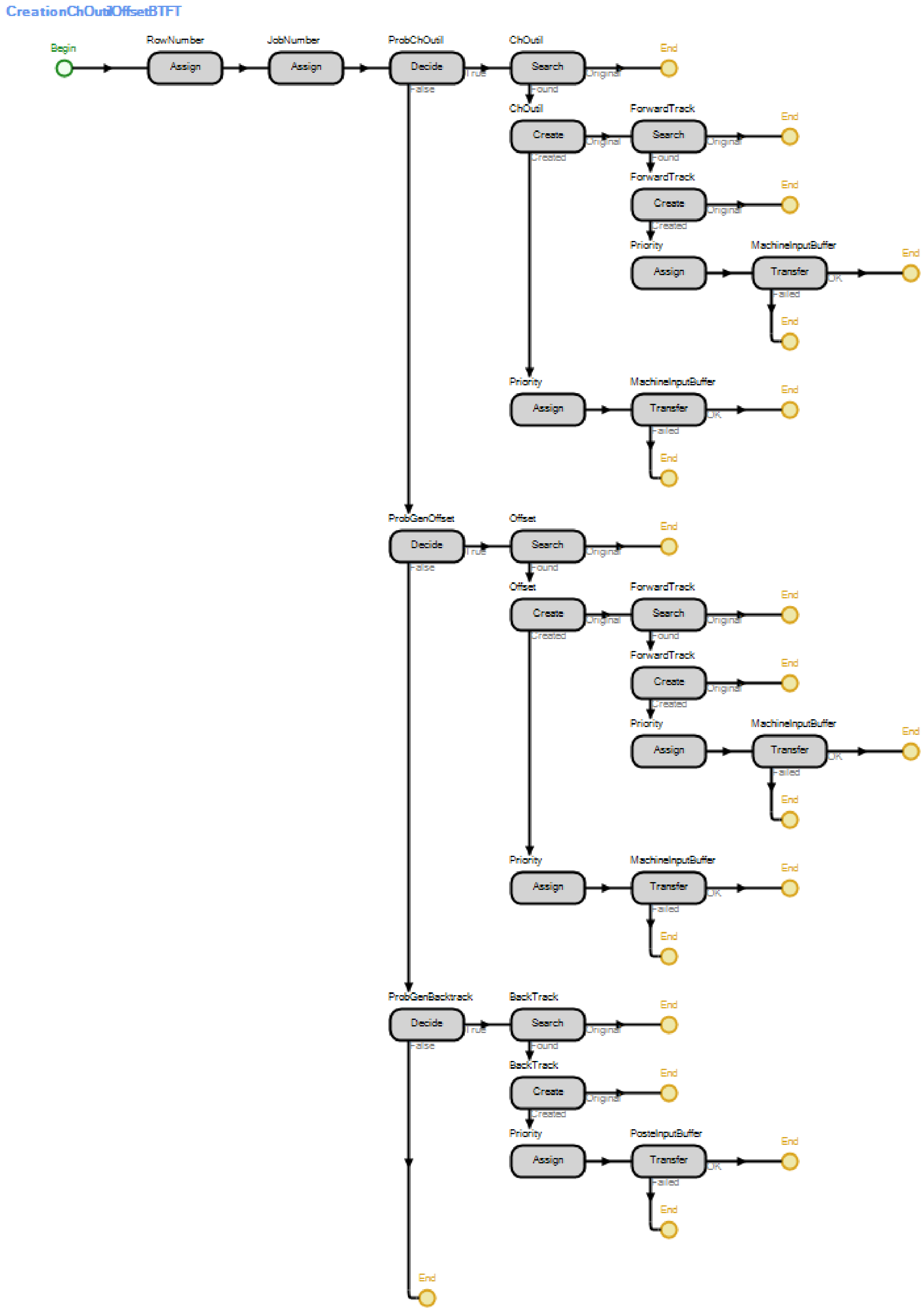
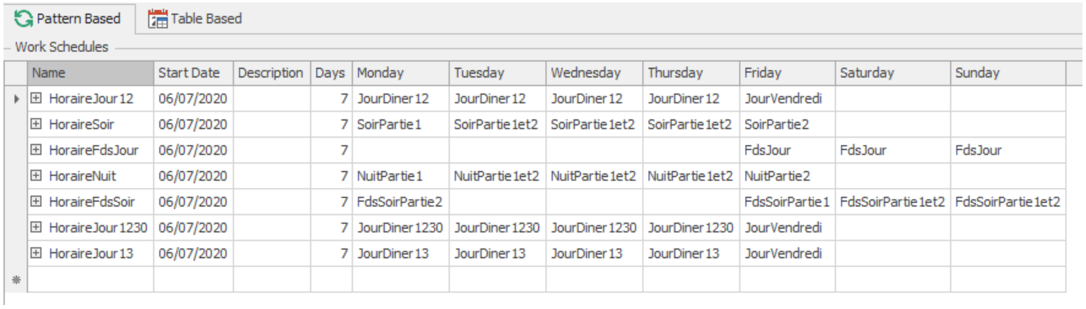


Figure 26: Task creation add-on process. This process creates entities that represent the random tasks that need to be executed during the production and is triggered every time a task entity enters the wokrcenter's sink.

The task *entities* linked to the non-conformity process are created inside the workcenter sub model. Based on the value of the parameters (*ProbGenToolChange*, *ProbGenOffset*, *ProbGenBacktrack*) of the *entity* triggering the process, this process will generate new *entities*. Once an *entity* is created, we must look at the data table to find the line that corresponds to the *entity* that needs to enter the model so as to associate the created *entity* to its correct parameter values. Every *entity* in the simulation model must refer to a line in the input data table since it is this table that holds many important parameters (e.g., the processing time of the *entities*). Those *entities* exist in the input table but were not created at the beginning of the simulation since they have 0 as their quantity. Once the correct row has been found, the *entity* is created based on this row. We assign the highest priority to this new task *entity*, so it is inserted at the beginning of the workcenter queue. Finally, it is transferred to the input buffer queue of the appropriate *server*, that is, the CNC machine or the worktable.

3.2.4 Resource schedules

Operators work under different schedules. They also have pre-determined lunch breaks and throughout their shift. Each operator modeled in Simio was given its regular schedule as a parameter. The data must be provided in a standard table in Simio (Figure 27).



Name	Start Date	Description	Days	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
HoraireJour12	06/07/2020		7	JourDiner12	JourDiner12	JourDiner12	JourDiner12	JourVendredi		
HoraireSoir	06/07/2020		7	SoirPartie1	SoirPartie1et2	SoirPartie1et2	SoirPartie1et2	SoirPartie2		
HoraireFdsJour	06/07/2020		7					FdsJour	FdsJour	FdsJour
HoraireNuit	06/07/2020		7	NuitPartie1	NuitPartie1et2	NuitPartie1et2	NuitPartie1et2	NuitPartie2		
HoraireFdsSoir	06/07/2020		7	FdsSoirPartie2				FdsSoirPartie1	FdsSoirPartie1et2	FdsSoirPartie1et2
HoraireJour1230	06/07/2020		7	JourDiner1230	JourDiner1230	JourDiner1230	JourDiner1230	JourVendredi		
HoraireJour13	06/07/2020		7	JourDiner13	JourDiner13	JourDiner13	JourDiner13	JourVendredi		

Figure 27: Work schedules of the operators in Simio

As for the workcenters, they may run 24/7, provided that an operator in present to process its tasks. Otherwise, workcenters will naturally stop processing parts when they are waiting for an operator.

With the parameters of the simulation determined by hypotheses and input data, as well as the different behaviours modeled with *add-on processes*, the simulation model was ready to run experiments.

3.2.5 Verification

The simulation model was verified using a combination of techniques, as proposed by (Sargent, 2004). Sargent (2004) defines model verification as "ensuring that the computer program of the computerized model and its implementation are correct". Firstly, the technique of *face validity* was used, which consisted of asking experts in the company whether the modeled system behave correctly. A test of *internal validity* was also run, which consisted of experiments of several replications that helped to verify how the stochastic variability affected the results obtained based on the confidence intervals of different performance indicators. With the animation of the simulation model, we were also able to verify the system behaviour by observing the different moving parts under different circumstances. A *sensitivity analysis* was furthermore performed. With different values of probability, we looked at the stochastic variability in the model to ensure that it concorded with our expectations of how it should behave. At many stages in the development of the simulation model, the trace, which details each step executed in the simulation model by each *entity*, was examined step by step to ensure the accuracy of the model's logic. Finally, different events were compared between the simulation model and the real system to verify if they were comparable under a technique called *event validity*. For example, we looked at the number of machined parts and the number of setups executed in a week.

3.3 Experiments

In this section, we will present two experiments. The first experiment aimed at validating the simulation model as well as giving first results on the comparison of the different scenarios. (Schlesinger, *et al*. 1979) defined model validation as "substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model". The second experiment compares dynamic and static allocation on the basis of a week of production to analyze different performance indicators such as the total number of machined parts.

Three different scenarios were compared to see how they affect the productivity of the production system (total time needed to carry out a production schedule) and the total distance traveled by operators. Each scenario is defined by an allocation policy and a set of operators skill levels. The scenarios are summarized in Table 10.

Table 10: Scenarios presented

Name of the scenario	Allocation policy	Skills
StaticAllocation	One operator ↔ X machines	Current
DynamicAllocationCurrentSkills	X operators ↔ Y machines	Current
DynamicAllocationAllSkills	X operators ↔ Y machines	All

The base case scenario is called *StaticAllocation*. One operator is allocated to one CNC. Once the shift is over, another operator takes over. This scenario represents the way the tasks are currently allocated. Under the *DynamicAllocationCurrentSkills* scenario, we dynamically allocate it to the closest free and compatible operator that has the skills for the task. As an upper bound, we also simulated the utopic *DynamicAllocationAllSkills* scenario, where each operator would be trained for all machines and tasks. In Figure 28, the two different allocation policies are represented.

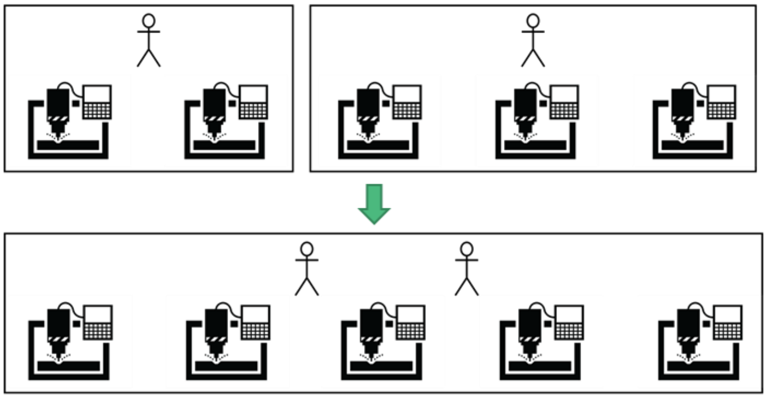


Figure 28: Allocation policies. One Operator <-> X machines (top) and X Operators <-> Y Machines (bottom).

3.3.1 First experiment: Model validation

The first experiment helped validate the simulation model when looking at the results for the *StaticAllocation* scenario, which represents the real system in its current state. It consisted of providing a finite number of parts as part of different jobs to machines. The same quantity of jobs was scheduled on each workcenter. Jobs may vary in number of parts. Then, the experiment ran for as long as it took to complete all tasks related to the machining parts tasks. In this experiment, there was no time limit imposed since our principal interest was to determine the time required to produce a given number of jobs on the CNC machines in the production system. A single dataset was collected as explained in previous section. Once the simulation model had processed all of the *entities*, we considered the production over and we analyzed the results.

3.3.1.1 Results

The following are the average results from 100 replications (95 % confidence intervals) of a given production schedule. Figure 29 shows the total production time in hours it took for the simulation model to process all *entities* (that is, the makespan).

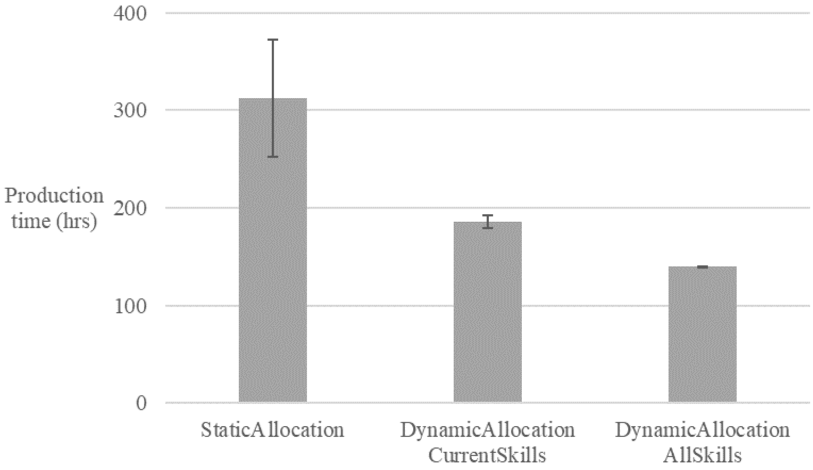


Figure 29: Production time in hours required to produce a finite number of parts represented by parts machining tasks.

Results obtained for the *StaticAllocation* scenario were consistent with what was expected of the real system based on different performance indicators such as the production time it takes to produce the number of jobs given as input. Results also show that on average, dynamic allocation (*DynamicAllocationCurrentSkills*) reduced total production time by 40.4 %. The *DynamicAllocationAllSkills* scenario led to some additional improvements, but it involved important training costs. The reduction of the confidence interval comes from the fact that under the *DynamicAllocationCurrentSkills* scenario we are much less affected by the stochastic nature of the processing times. Since jobs are scheduled on CNC machines in advance and cannot be processed by another machine, makespan may be greatly impacted by a delay on a single machine, while other CNC machines may be done with all their tasks.

In Figure 30, we present the results for the total distance traveled by operators in order to complete all tasks.

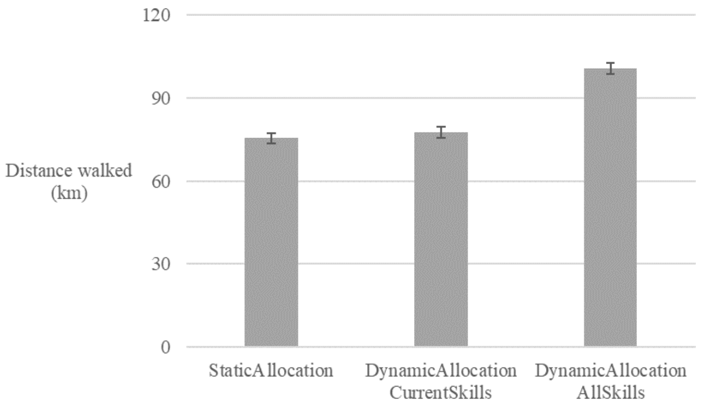


Figure 30: Total distance walked by the operators depending on the scenario.

We notice that results are very similar for *StaticAllocation* and *DynamicAllocationCurrentSkills*. This may be explained by the fact that operators are currently skilled for a subset of machines that are physically close to one another. *DynamicAllocationAllSkills* greatly impacts walking distance since multi-skilled operators may be called upon to travel between the different departments to deal with tasks instead of mostly staying in the same one.

This validation phase led to interesting results, although many hypotheses were required to produce all necessary input data. After seeing the promising results from this phase, our industrial partner was much more inclined to provide necessary data to simulate a week of production using the simulation model.

3.3.2 Simulation of a full week of production

In this experiment, an entire week of production was simulated. We want to compare the proposed scenarios on the basis of a full week of production, all while using the same performance indicators as those with which the company is comfortable in order to motivate the implementation of the task system on the production floor. The production team relies on weekly indicators indicating the number of parts produced during the week as well as the utilization time of machines, that is, the time spent machining parts on CNC machines. Additionally, the number of parts produced during a week may give better indications on the overall productivity of the factory.

The simulation model was kept identical with a few exceptions following the validation made in the previous experiment. Firstly, the partner provided the layout of the factory so we were able to place the different objects in the Simio simulation model in a manner that would better represent the actual production floor. The four different sectors were represented, that is: the Combitec sector, the lathes sector, the measurement machines sector, and the milling machines sector. Figure 31 shows the different production sectors in the simulation model.

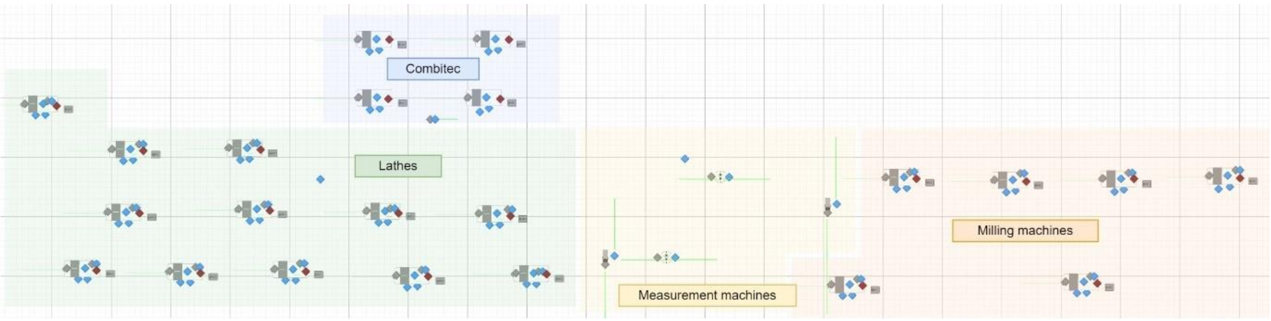


Figure 31: Modifications to the simulation model – Illustration of the realistic floor layout with the different production sectors in Simio.

3.3.2.1 Results

The following are the average results from 100 replications (95 % confidence intervals) of a given production week. Figure 32 presents the number of parts machined in a 168-hour week of production.

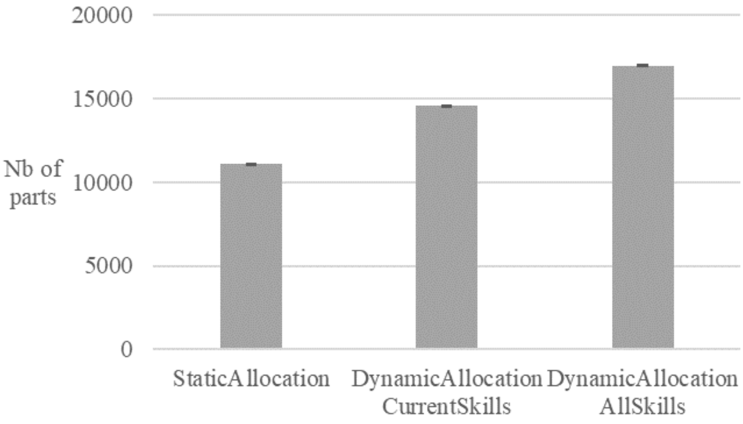


Figure 32: Number of machined parts during a 168-hour work week under the different scenarios

Dynamic allocation under the *DynamicAllocationCurrentSkills* scenario allowed part production to increase by more than 30% in a week of production. Supposing completely versatile operators increased the number of parts produced by an additional 17%. Interval confidences were very small compared to the validation phase. This can be explained by the fact that a single machine could cause an important bottleneck and delay the end of the production for all parts. In this experiment, we only consider a week of production, which is more replicable in terms of production output.

Table 11 gives further information on how the workload is divided between CNC machines. This indicator is extremely relevant since the number of parts produced depends on the cycle time of the parts on the scheduled job. It is important to note that even if an operator attends a machine 100% of the time, it would not be able to produce parts for 168 hours since some tasks require the machine to be inactive. The aim is then to minimize the time where a machine is waiting for an operator to resume producing parts. Most machines are also not in production for 168 hours per week, some being turned off during the night or the weekend, for example.

Table 11: Average utilization time of CNC machines in a 168-hour work week.

	StaticAllocation	DynamicAllocation CurrentSkills	DynamicAllocation AllSkills
Avg	105.23 ± 0.46	113.79 ± 0.63	114.83 ± 0.60
Max	155.35 ± 1.28	157.14 ± 1.53	155.58 ± 2.12
Min	45.60 ± 0.03	67.28 ± 0.06	75.36 ± 0.02
Std.-dev.	25.34 ± 0.37	21.27 ± 0.40	22.49 ± 0.51

The average time spent by CNC machines machining metal parts is shown in Table 11. CNC machines spent on average 8% more-time machining parts in the *DynamicAllocationCurrentSkills* scenario compared to the *StaticAllocation* scenario. Average utilization time was almost the same in both scenarios representing the new paradigm. While the maximum utilization time was similar in all scenarios, it is extremely interesting to compare the minimum utilization rate. In the *StaticAllocation*, some machines may be neglected when their allocated operator is too busy dealing with tasks coming from its other machines. The *DynamicAllocationAllSkills* scenario proposes the higher minimum, supposing that all machines are producing parts for at least 75 hours in the week. The standard deviation between machines was similar in all three scenarios.

3.4 Conclusion

A discrete-event simulation was developed in order to simulate task allocation in a high precision metal parts machining workshop. The main goal was to measure how productivity varies when more operators are allowed to process a given task. Different skill levels were also taken into consideration. Results showed that important productivity gains can be obtained when allowing a dynamic allocation of the tasks to the operators instead of having a single operator dealing with all tasks associated with a given CNC machine. We simulated a week of production in the factory, providing performance indicators followed by the company's management team. These results were motivating to pursue this dynamic task allocation project. We have also shown that higher skill levels led to further improvement. With dynamic allocation, training employees could have a very interesting impact in terms of productivity.

Chapter 4: Dynamic allocation of human resources: Case study in the metal 4.0 manufacturing industry

The article named « Dynamic allocation of human resources: Case study in the metal 4.0 manufacturing industry » is inserted in this thesis section. It was submitted to the *Industrial Journal of Production Research*. The version submitted is identical to the version presented in this thesis.

Résumé

Les concepts de l'industrie 4.0 nous emmènent à repenser l'allocation des ressources humaines en usine, même dans des environnements plus traditionnels comme l'usinage des métaux. Bien que l'usinage de pièces sur des machines à commande numérique soit automatisé, certaines tâches manuelles doivent encore être exécutées par des opérateurs. L'approche actuelle consiste généralement à affecter de manière statique les opérateurs à une ou plusieurs machines. Cette stratégie provoque des goulots d'étranglement qui pourraient être évitables. Nous proposons donc un modèle d'optimisation pour assigner dynamiquement les tâches aux opérateurs dans le but de minimiser les délais de production. Trois scénarios différents sont comparés; l'un représentant la méthode d'allocation statique présentement largement utilisée dans l'industrie et deux autres qui permettent une plus grande flexibilité dans l'allocation des opérateurs. Le problème d'allocation dynamique des tâches est résolu à l'aide d'un modèle d'optimisation développé avec la programmation par contraintes. Le modèle a été appliqué à une étude de cas d'un atelier d'usinage de métaux de haute précision. Les résultats expérimentaux montrent que le passage d'une allocation statique opérateur-machine à une allocation dynamique réduit de 76% les retards moyens de production causés par les opérateurs. En augmentant la polyvalence des opérateurs dans un contexte d'allocation dynamique, on diminue encore d'avantage les retards de production.

Abstract

Industry 4.0 concepts makes it possible to rethink human resources allocation, even for more traditional environments like metal machining. While parts machining on Computer Numerical Control (CNC) machines is automated, some manual tasks must still be executed by operators. The current approach is typically that operators are statically allocated to one or many machines. This causes avoidable bottlenecks. We propose an optimization model to dynamically assign the tasks to the operators with the objective of minimizing production delays. Three different scenarios are compared; one representing the current widely used static allocation method and two others that allow for more flexibility in the operators' allocation. The dynamic task assignment problem is solved using a constraint programming model. The model was applied to a case study from a high-precision metal manufacturing job shop. Experimental results show that switching from a static operator-to-machine allocation to a dynamic one reduces by 76% the average production delays caused by human operators. Supposing more versatile operators under the dynamic allocation leads to further improvements.

Introduction

The relationship between humans and machines has been constantly evolving since the first industrial revolution. The manufacturing sector has seen many major changes, and the advent of new technologies is modifying the way humans are involved in the production process. When the first metal milling machine was invented in the early 1800s, it required a full-time operator to operate it. With the advent of computers, automation made its way during the third industrial revolution. The metal parts machining industry has shifted from mechanical machining to automated machining, using Computer Numerical Control (CNC) machines (Ivanov *et al.*, 2019). Indeed, parts could now be produced in an automated manner, removing the need for a human to mechanically operate the machines to produce metal parts. CNC machines can lead to many financial benefits when used to their full potential. To do so, the production flow organization must be coordinated, since human operators are now required to execute supporting tasks (Hamrol *et al.*, 2018). It therefore remains crucial to schedule and allocate these resources properly. Industry 4.0 is characterized by integration, interactivity and interconnexion of all production processes of an industrial company, made possible by emerging new technologies (Idrisov *et al.*, 2018).

A typical machining workshop is composed of multiple CNC machines. Even with parts machining being automated, supporting tasks such as parts measuring, or raw material addition still require human intervention. These tasks occur on a periodic basis during the job production. Jobs consist of a pre-determined quantity of identical parts processed on a CNC machine. Some tasks, such as dealing with non-conformities or changing a broken tool, may also occur in a stochastic manner. Conventionally, every operator is allocated to one or more CNC machines. This operator assists with production by handling all of the machine's various tasks. Task-to-operator allocation is static and follows the machine-to-operator allocation, e.g., a tool changing task on a machine will be automatically allocated to the only operator in charge of this given machine. This method is simple and requires few decisions that can be taken by production floor managers at the beginning of every work shift. However, this static allocation is sub-optimal since certain machines might happen to have tasks to perform at the same time. In the event where machines allocated to the same operator have simultaneous tasks, a bottleneck forms around the operator. Failure to provide instant human support may lead to a machine shutdown, delaying production, and inducing delays in orders deliveries. Simultaneously, other operators might be available for extended periods of time while their allocated machine(s) run on automatic mode.

Operator allocation policies were shown to have a significant reduction effect on due date performance when considering job tardiness in a dual-resource constrained job-shop system (DRC) in Kher and Fry (2001). Dynamic allocation approaches have also proven to be beneficial in terms of productivity in Greis et al. (2019) and Beauchemin et al. (2020).

However, such a dynamic task-operator allocation requires making decisions constantly (i.e., several decisions per minute), based on real-time information. With the collection of actual data, tasks can now be allocated to operators in a dynamic and optimal manner. Moreover, it also requires an intelligent agent for the real-time decision-making. While in Greis et al. (2019) and Beauchemin et al. (2020), simple heuristics were used, this research aims to propose a new way for allocating tasks to operators in a job-shop context, taking into account the skills of the different operators.

To achieve this goal, industrial data was first collected. The company is a high-precision metal parts factory located in Québec City, Canada. It uses built-in systems interconnecting and collecting data on different aspects of the job-shop production, while collaborative robots move the parts from one machine to another.

We then developed an optimization model based on the flexible job-shop problem adapted to the task allocation of human resources. Different scenarios were then tested, and the results analysed so as to measure the effect of a dynamic allocation on the system. Results showed how the dynamic allocation of operators could improve the overall tardiness of production in the factory.

Assigning operators to tasks and scheduling their different tasks is an NP-hard problem (Bouajaja and Dridi, 2017), but with current qualified labour shortages, we believe a change of paradigm is necessary since a production manager can no longer be expected to make all the decisions regarding task allocation in a real-time context. Our article thus brings a scientific contribution by proposing a new optimization method to dynamically allocate real-time tasks to operators in a 4.0 metal machining industry. While one can reasonably expect an upgrade in performance when using dynamic allocation, this article quantifies this performance upgrade while proposing a new dynamic allocation system that can be implemented in the industry. Indeed, by using industrial data, the results obtained are practical and can be extended to the manufacturing sector under similar conditions.

The article is divided as follows: Section 1 presents the preliminary concepts for the article. In section 2, the methodology implemented is detailed as well and the model developed. Section 3 describes the experimentation results and their analysis while section 4 concludes the paper.

Preliminary concepts

A job consists of a number of identical parts that must be produced on a machine. During a job, many tasks devoted to human operators need to be completed at different frequencies (e.g., measuring every 10th part). Scheduling for a metal machining factory is a two-step process. The first step (job scheduling) is to assign/schedule the jobs to CNC machines (Figure 33: Scheduling horizons for job scheduling and task scheduling.). Orders/jobs are typically promised to customers months in advance and scheduling the jobs on the machines is done far in advance given a planning horizon that lasts many weeks. Scheduling the jobs on the CNC machines falls in the flexible job-shop scheduling category and is NP-hard (Özkul *et al.*, 2021).

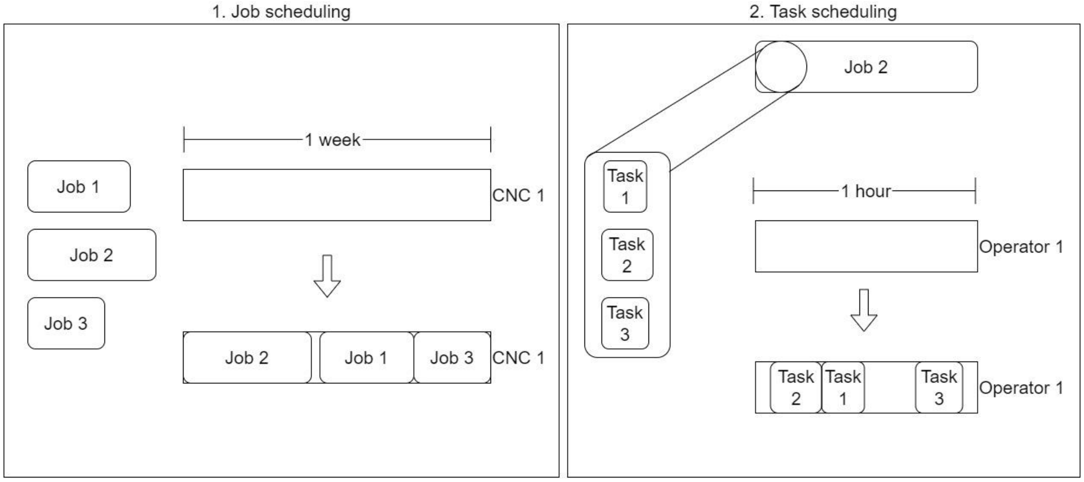


Figure 33: Scheduling horizons for job scheduling and task scheduling.

The second step (task scheduling) is to schedule and allocate the tasks to the operators in real time, which is our main concern in this study. The different types of tasks are parts measuring (M), raw material addition (A), tool changes (C), machine setups (S), and management of non-conformities (N). Machining the parts on the CNC machines were not explicitly included in the model since they are *de facto* already scheduled (see Figure 1).

Coordinates Measuring Machines (CMM) are used in the industry since they are precise and highly automated (Zheng *et al.*, 2018). Robots can even transport the part from the CNC machine to the CMM and vice-versa, to completely automate the parts measuring task. However, not all characteristics of a part can be measured automatically with the CMM, and operators still need to take manual measurements. Parts measuring on CMM machines must then be allocated and scheduled, at the same time as the other supporting tasks executed by operators.

Allocating jobs, shifts and/or tasks to human resources is a domain area that has been greatly studied in the last few years, although in different contexts than the one in this article.

A problem that shares similarities with ours is the staff scheduling (Caprara, Monaci and Toth, 2003) or employee timetabling problem (Meisels and Schaerf, 2003). In this type of problem, the goal is to assign each employee to a work shift. This problem can also mean to assign workers to tasks/jobs (Sicong, Weng and Shigeru, 2009). Bouajaja and Dridi (2017) proposed a review in which human resource allocation problems were investigated. They found that the different methods used to deal with the human resources allocation problem are exact methods (such as linear programming), heuristic algorithms, meta-heuristics, as well as any hybridization of those methods. An area in which this problem is very important is in hospital systems. Indeed, many articles (Bourdais, Galinier and Pesant, 2003; Eiselt and Marianov, 2008; Lanzarone and Matta, 2014; Ho *et al.*, 2018) were published tackling the Nurse Rostering Problem, which basically aims at creating the schedule for the nurses in a hospital or health-care establishment. Ho *et al*. (2018) proposed a platform for dynamic nurse scheduling based on integer linear programming. While integer linear programming has been often used to solve the Nurse Rostering Problems in the last few years, many articles proposed constraint programming (CP) models (Bourdais, Galinier and Pesant, 2003). CP consists of representing a problem with its constraints and then finding a solution that satisfies these constraints or optimizes an objective (Apt, 2003). It has been used (Alade and Amusat, 2019) as well as compared with integer linear programming (Trilling, Guinet and Magny, 2006) to solve the nurse rostering problem. Lanzarone and Matta (2014) proposed different heuristics to solve this problem with the objective of minimizing the overtime of the nurses, while trying to make their solution robust to unpredictable changes in the demand. In the same vein, Eiselt and Marianov (2008) aimed at assigning tasks to minimize the overtime of their skilled employees and make the schedule as fair as possible by mapping their skills in a skill-space.

One important aspect of assigning jobs, shifts and/or tasks to employees is to make sure the employee is appropriately skilled to perform the operation. The multi-skilling research domain focusses on assignment problems for which the employees have different sets of skills that must be considered. A 2021 review considered 160 publications on the subject (Afshar-Nadjafi, 2021). In this review, articles on production planning (Costa, Cappadonna and Fichera, 2014) and shift scheduling (Bhulai, Koole and Pot, 2008) were discussed in relation to the multi-skilling domain. In her thesis, Eriksson (2020) aimed at scheduling the workforce in a contact center by using three different Mixed-Integer linear Programming (MIP) models. Edi and Duquenne (2009) also used a multi-skill optimization model to assign tasks to workers.

Human resources allocation has also been considered of great interest in the Resource Constrained Project Scheduling problem (RCPSP) since the 1950s (Kelley, James E. and Walker, Morgan R., 1959). This project approach is concerned with longer, more complex tasks that may require many workers. Zammori and Bertolini (2015) proposed an interesting framework to allocate multi-skilled resources. Their model applies to long project tasks that can be chosen and then shared between multiple resources.

While all these papers aim at assigning resources to jobs, tasks or shifts, they all suppose that activities are previously scheduled. Scheduling problems like the job-shop scheduling problem (JSP) aims at ordering the jobs (or the tasks) in a way to optimize a certain objective, such as the makespan (Manne, 1960). The JSP is NP-hard with three jobs to schedule on three machines (Sotskov and Shakhlevich, 1995). Constraint programming (CP) has gained in popularity in the last few years because of its flexibility and overall good performance to solve this problem

(Oliveira, Smith and Jt, 2000). Many recent articles concluded that the CP approach outperforms the MIP when it comes to the resolution of the JSP (Kress and Müller, 2019; Meng *et al.*, 2020; Ham, Park and Kim, 2021). CP can also be used in hybridization with other techniques, such as local-search (Watson and Beck, 2008). While most formulations of the JSP consider the last completion times, Bülbül and Kaminsky (2013) included tasks' individual completion times in their objective function. Agnetis *et al*. (2014) tackled the job-shop scheduling problems with two resources; machines and operators. In their paper, each task must be allocated to both an operator and a machine. They solved the problem with two heuristics. When jobs or tasks need to be scheduled in addition to being allocated to resources, the JSP is extended and becomes the flexible job-shop scheduling problem (FJSP). The problem can be divided in two sub-problems: scheduling and assignment. It is also considered NP-hard (Garey, Johnson and Sethi, 1976). This is the problem that best characterizes ours. This problem has been solved using a hybrid of particle swarm optimization and simulated annealing (Xia and Wu, 2005).

When operators are included in the job-shop scheduling problem, they are always considered as a secondary resource in the dual-resource job-shop scheduling problem and its many variants and are often considered the restraining resource (Xu, Xu and Xie, 2011). Cunha *et al*. (2019) tackled the dual-resource problem considering that jobs may require an operator's assistance at specific times during the process, in the context of quality control laboratories. Their method cannot be applied in a real-time scheduling context since their MIP approach is not solved to optimality in less than one hour. In the aeronautical industry, the cyclic flexible job-shop scheduling problem with operators was solved with a MIP (Borreguero-Sanchidrián *et al.*, 2018). (Sierra, Mencía and Varela, 2015) proposed an optimal schedule generation scheme in the case of the dual-resource job-shop scheduling problem. Artigues *et al*. (2009) even integrated the employee timetabling problem and the job-shop scheduling problem with CP hybridized with a linear programming relaxation.

While these articles address resource allocation and scheduling, to the best of our knowledge, none address the scheduling and allocation of periodic/stochastic tasks to human resources in a real-time context under Industry 4.0. Some articles scheduled human resources at the same time as machines based on an integrated method (*dual-resource job-shop scheduling*). However, this integrated approach is not conceivable in a context where production scheduling aims at scheduling long jobs that last many hours, even days, while shorter tasks of a few minutes with greater uncertainty levels need to be allocated to human operators. Integrating these two types of scheduling is not appropriate in this context.

Both Beauchemin, *et al.* (2020) and Greis *et al.* (2019) simulated operators dynamic allocation in an environment highly based on IoT in which machines are aware and operators are tracked on the factory floor. In their proposition, operators would be allocated by an intelligent cognitive engine that can learn the dynamic patterns of part production in the factory. However, the allocation engine has not been developed and their simulation uses heuristics for assignation (e.g. assigning the closest available operator to the task). In the next sections, an optimization model is proposed in order to carry on with allocation in an optimal way.

Methodology

The methodology followed during the research is illustrated in Figure 34.

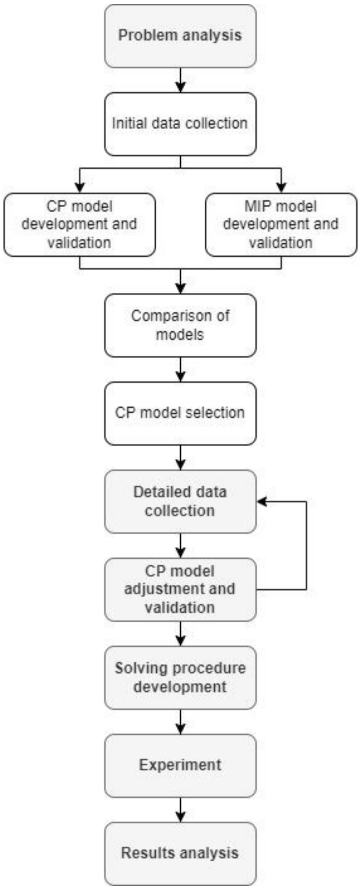

Problem analysis
Initial data collection
CP model development and validation
MIP model development and validation
Comparison of models
CP model selection
Detailed data collection
CP model adjustment and validation
Solving procedure development
Experiment
Results analysis

Figure 34: Methodology followed during the research.

In order to investigate the profitability of using a dynamic allocation policy for resources, an initial set of data from a high-precision metal parts factory were first collected and explored. As a preliminary study, the allocation problem was next encoded in both a Mixed-Integer Programming (MIP) model and a Constraint Programming (CP) model. The models were solved and the obtained results were compared. This comparison was performed based on an experiment encompassing 30 instances of different sizes, aiming at scheduling between 14 and 57 tasks. The time-limit was set to 10 minutes. It was observed that the CP model managed to find the optimal solution for 25 instances while the MIP model only found the optimal solution for 10 instances. In addition, the MIP model did not manage to find a first feasible solution for 6 instances under the time-limit selected. In only one instance out of the 30 did the MIP model outperform the CP model by managing to find a better (but still non-optimal) solution during the time limit (this was one of the 5 instances for which the CP did not manage to find the optimal solution). While this preliminary comparison showed that the CP model outperformed the MIP in terms of solution performance, it also showed that the CP model was faster to solve. Indeed, for the ten instances where an optimal solution was found, the CP model was on average 82% faster at obtaining the optimal solution, which is critical when considering solving instances in a real-time production system. These results are aligned with the experiments presented in (Ham, Park and Kim, 2021) which demonstrated that "*CP is quick to generate efficient (or optimal) solutions*". Their problem shares many similarities to ours, taking place in a flexible job shop environment with heterogeneous machines in which each operation must be processed on a qualified machine (or resource). In addition, (Kress and Müller, 2019) also compared MIP and CP in a flexible job-shop scheduling context with machines and human operators and concluded that "*the CP solver clearly outperforms the MIP solver for the considered modeling approaches. The CP solver tends to provide high quality solutions within reasonable time*". The conclusion of these articles is consistent with our findings and convinced us to pursue the project using the constraint programming model.

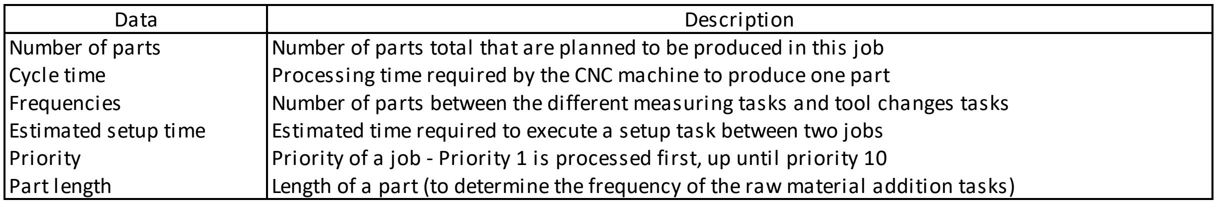
Following this comparison, more data were rendered available and collected under an extended data collection. A solving procedure that consisted of a two-stage approach as well as search strategies was also developed to conduct the different experimentations. The constraint programming model was then solved using the detailed data set so as to compare the current scenario which is a static allocation of tasks to operators with two other ones, a first considering a dynamic allocation in which the operators are fully polyvalent and a second in which the operators possess only certain skills. An additional experiment was performed in order to consider the impact of increasing the machine/operator ratio.

While a problem analysis has been provided in the introduction and the preliminary concepts, we will now detail the greyed-out steps in Figure 2, starting with the detailed data collection.

Data

For the metal machining job shop under study, jobs take at least one entire day to complete, but often more than one day. Numerous information is available regarding the jobs at hand, such as the cycle time of the machined parts, the frequencies at which the different tasks must be accomplished, the estimated setup time a job should take, the number of parts that need to be machined on the job, and the priority of this job. The priorities represent the order in which the jobs will be executed on a CNC machine. Each CNC machine has its next ten jobs planned at any given time, which covers minimally the next week of production. This number of jobs is more than enough to cover a rolling horizon of one hour. Table 12 presents some of the job-related data useful for the planning algorithm.

Table 12: Available data relating to jobs.

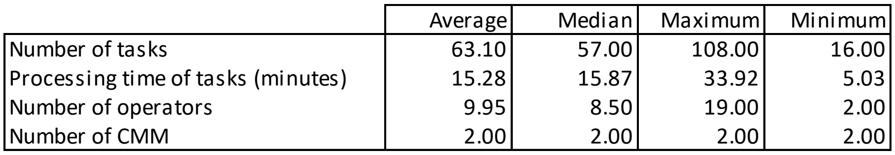


Data	Description
Number of parts	Number of parts total that are planned to be produced in this job
Cycle time	Processing time required by the CNC machine to produce one part
Frequencies	Number of parts between the different measuring tasks and tool changes tasks
Estimated setup time	Estimated time required to execute a setup task between two jobs
Priority	Priority of a job - Priority 1 is processed first, up until priority 10
Part length	Length of a part (to determine the frequency of the raw material addition tasks)

The data available regarding the factory state concerned current non-conformities that need to be dealt with, remaining processing time and number of the current part on a CNC machine, completed tasks, operators currently working, as well as current machine-operator allocations.

We extracted this data every hour (tasks, states of the resources, etc.) over the course of a week which led to a total of 168 datasets. From this number, 60 datasets were taken during hours in which the factory was offline, so they were not considered. Table 13 shows various statistics concerning these datasets. The average number of tasks to schedule in a dataset is 63.10, and the average task processing time is 15.28 minutes. These tasks need to be allocated to 9.95 operators on average and 2 CMMs. It is assumed that the transfer time required to move from one CNC machine to another is negligible (which is the case). The CNC machines are denoted as workcenters for clarity purposes.

Table 13: Summary of the 108 datasets.



	Average	Median	Maximum	Minimum
Number of tasks	63.10	57.00	108.00	16.00
Processing time of tasks (minutes)	15.28	15.87	33.92	5.03
Number of operators	9.95	8.50	19.00	2.00
Number of CMM	2.00	2.00	2.00	2.00

Constraint programming optimization model

The CP model developed to solve the allocation problem goes as follows:

Sets

T (tasks): Set of tasks, released in the upcoming hour, that need to be scheduled.

TR (type resource): Set of possible types of resources {Operator, CMM}.

R (resources): Set of resources available in the upcoming hour.

TA (type_action): Set of different types of tasks {M (measuring parts), C (tool changes), A (raw material addition), N (non-conformities management), S (setups)}.

W (workcenters): Set of workcenters {CNC1, CNC2, …}.

TS (timespan): Set of minutes in the planning horizon.

Parameters

RT_t: Release time of task *t*. First minute at which a given task $t \in T$ can start.

DT_t: Due time of task *t*. Last minute at which a given task can be processed.

PT_t: Processing time. Number of minutes required to complete task *t*.

PA_t: Part number linked to task *t*.

TRE_t: Type of resource required to process task *t*.

TAC_t: Type of action of task *t*.

W_t: Workcenter on which task *t* occurs.

RES_r: Type of resource *r*.

$SK_{t,r}$: 1 if a resource *r* is skilled to execute a task *t*, 0 if not.

AV_r: First minute at which a resource *r* may start processing a task.

Variables

$s_t \in TS$: Start time. Minute at which task *t* starts.

$a_t \in R$: Allocation. Resource allocated to task *t*.

$p_{t,r} \in TS$: Processing time in minutes resource *r* spends on task *t*.

$ta_t \in TS$: Tardiness in minutes of a task *t*.

Objective function

Minimize

$$\sum_{t \in T} ta_t \tag{1}$$

Constraints

$$s_t \geq RT_t \qquad \forall\, t \in T \tag{2}$$

$$s_{t2} \geq RT_{t_2} + \max(0, \min(60 - RT_{t_1}, s_{t_1} + PT_{t_1} - DT_{t_1})) \qquad \forall\, t_1, t_2 \in\ T \text{ where } W_{t_1} = W_{t_2}\ \wedge PA_{t_1} < PA_{t_2}\ \wedge TAC_{t_1} \neq \mathrm{M} \tag{3}$$

$$p_{t,r} = \begin{cases} PT_t & \text{if } a_t = r \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ r \in R, t \in T \tag{4}$$

$$\text{disjunctive}([s_t | t \in T], [p_{t,r} | t \in T]) \qquad \forall\ r \in R \tag{5}$$

$$s_{t_1} > s_{t_2} + PT_{t_2} - 1\ \vee s_{t_1} + PT_{t_1} - 1\ < s_{t_2} \qquad \forall\, t_1, t_2 \in T \text{ where } t_1\ \neq t_2 \wedge PA_{t_1} = PA_{t_2} \wedge\ TAC_{t_1} = \mathrm{M} \wedge TAC_{t_2} = \mathrm{M} \wedge W_{t_1} = W_{t_2} \tag{6}$$

$$TRE_t = RES_{a_t} \qquad \forall\ t \in T \tag{7}$$

$$SK_{t,r} = 0 \Longrightarrow\ a_t \neq r \qquad \forall\ t \in \mathrm{T}, r \in R \tag{8}$$

$$\text{if } a_t = r \text{ then } s_t \geq AV_r \text{ endif} \qquad \forall\, t \in T, r \in R \tag{9}$$

$$ta_{t_2} = \max(0, A_{t_2}) - \min(\max\left(0, A_{t_2}\right), B_{t_1,t_2}) \qquad \forall\ t_2 \in T \tag{10}$$

$$A_{t_2} = \min(60 - RT_{t_2}, s_{t_2} + PT_{t_2} - DT_{t_2})$$

$$B_{t_1,t_2} = \sum_{\substack{t_1 \in T \\ W_{t_1}=W_{t_2} \\ PA_{t_1}<PA_{t_2} \\ TAC_{t_1}\neq \mathrm{M}}} ta_{t_1}$$

Constraint (2) ensures that tasks do not begin before their release time. Constraint (3) adds a delay to the release time of a given task if the previous task on the CNC machine was started late. Since parts machining tasks are not included in the model, this constraint ensures that we account for their delayed start following a delay in a critical task that leads to a machine shutdown. Constraint (4) populates the resources $p_{t,r}$ variable with the processing time of its allocated tasks. Constraint (5) uses the *disjunctive* constraint (Carlier, 1982) to ensure that tasks allocated to the same resource do not overlap. To ensure that only the tasks allocated to the same resource are considered, we use the processing time variable $p_{t,r}$ populated in Constraint (4) which is null when a task is not associated to the resource considered. Constraint (6) ensures that parts measuring tasks related to the same part do not overlap. It is intended for parts that need to have characteristics measured both with the CMM and manually by an operator. Constraint (7) ensures that the tasks are being allocated to the appropriate type of resource (CMM or operator). Constraint (8) imposes that an operator cannot be allocated to a task for which s/he is not skilled. Constraint (9) considers the availability of the resources at the beginning of the timespan, so as not to allocate a resource to a task scheduled to start before the resource is available. Constraint (10) serves to ignore the tardiness in a task linked to a previous delayed task. Indeed, tasks that have a delay in their release time because of Constraint (3) should not be penalized for their tardiness.

Solving procedure

Two-stage approach

Since the problem investigated here deals with tasks in two different categories (i.e., critical tasks and non-critical tasks), the model was solved using a two-stage approach. The non-critical tasks were first removed from the original dataset and the proposed model solved in order to minimize the tardiness on critical tasks. In a second stage, the non-critical tasks were added back and the model solved considering the complete set of tasks (i.e., critical and non-critical). The tardiness obtained in first phase for the critical tasks had to be respected in the second phase, since we do not want to allow additional tardiness to be planned on critical tasks in favor of non-critical tasks. The total tardiness was then minimized with the additional constraint (11) of reproducing the tardiness obtained in phase one for the critical tasks. Constraint (11) makes use of an additional parameter:

TP: Tardiness in minutes obtained in phase 1.

$$\sum_{\substack{t \in T \\ TAC_t \neq \mathrm{M}}} ta_t \leq TP \tag{11}$$

Using this two-stage approach, it became possible to make sure that tardiness on all critical tasks was minimized before minimizing tardiness on non-critical tasks, which is consistent with what is needed in the industrial context considered.

Search strategies

One of the advantages of constraint solvers is that they easily allow fine-tuning the search strategy (variables and values selection priority) (Apt, 2003). The first variables to be assigned are the critical tasks' starting time variables, followed by the parts measuring tasks' starting time variables. The branching heuristic chooses the task that can start the earliest and assign its starting time to its earliest. Once the starting times are assigned, the branching heuristic selects variable a_t, in lexicographical order, and assign the smallest value in its domain.

For the second phase of the optimization model, the search strategy was modified slightly in order to branch first on the tasks that are present in phase 1, that is, the critical tasks. Then, the model branches on the new tasks added in phase 2.

Experiments

Three different scenarios were tested and compared. The first scenario, *StaticAllocation*, represents the status quo. While the factory does not currently use such a model to allocate its resources, this first scenario was created in order to emulate the current manual method used in the factory. We did so by constraining the operators to only process tasks related to one (or more) given CNC machines, as is currently the case in the factory. The machine(s)-operator assignations are those obtained from the industrial partner. In this scenario, the only decisions being made by the optimization model are to prioritize between tasks on a given resource if they arrive at the same time.

In the second scenario, *DynamicAllocationCurrentSkills*, operators are allocated to tasks given their different sets of skills. The skills matrix was created according to the partner's data. In this scenario, operators may be dynamically allocated to any tasks for which they are appropriately skilled.

Finally, an optimistic scenario, *DynamicAllocationAllSkills*, was studied to show the impact on production if all operators have the skills to process every task. In this scenario, each task may be allocated to any operator. This will be used to put an upper bound to measure the additional gain the company could reach if investing in training.

The model was implemented in MiniZinc 2.5.5 (Nethercote *et al.*, 2007) and solved with the Chuffed 0.10.4 solver. All experiments were performed on a Microsoft Windows 10 Pro machine with 16.0 GB of RAM memory and an Intel Core i7-4790K (4.00GHz) processor.

Results

For each scenario, the computation time, tardiness, and occupation rates were examined. The tardiness of the critical tasks was also computed separately. Indeed, some tasks such as raw material addition and tool changes have a direct 1:1 impact on production delays, while parts measuring does not have a direct impact but may have consequences such as an increased number of non-conformities. How each resource type affected the tardiness of the solution was also studied. The indicator for resource utilization was the average amount of time worked for each type of resource. The fairness of the solution was then evaluated by comparing the occupation rate distributions.

Table 14 presents the computation time for each scenario. The total column represents the sum of the resolution time for both phases in a given instance. With an average sum of under 3 seconds and a median of around 1 second in every scenario, the model is performant enough to be used in real life instances and be implemented in real-time. Framinan, Fernandez-Viagas and Perez-Gonzalez (2019) proposed a 45-second time limit in order to get nearly real-time reactions, while Harmonosky and Robohn (1991) mentioned that depending on processing times, a resolution in under five minutes might still be acceptable in a real-time system. Since the maximum resolution time of the model for all instances is a little over two minutes, this model can easily be used in real-time and is therefore considered as acceptable. Phase 2 instances are always longer to solve than phase 1 since the tasks contained in phase 1 are a subset of the tasks contained in phase 2.

Table 14: Computation time (in seconds) for the 108 datasets. Results are presented for phase 1, phase 2, and the total of both phases.

	Computation time (seconds)								
Scenario	StaticAllocation			DynamicAllocationCurrentSkills			DynamicAllocationAllSkills		
	Phase 1 (Critical tasks)	Phase 2 (All tasks)	Total	Phase 1 (Critical tasks)	Phase 2 (All tasks)	Total	Phase 1 (Critical tasks)	Phase 2 (All tasks)	Total
Average	0.32	2.03	2.36	0.38	1.95	2.33	0.39	1.48	1.87
Median	0.29	0.64	0.93	0.33	0.86	1.19	0.34	0.68	1.01
Maximum	0.58	131.68	131.98	0.76	110.01	110.35	0.77	71.56	71.90
Minimum	0.21	0.24	0.45	0.21	0.25	0.46	0.22	0.25	0.47

Table 15 presents the average tardiness obtained per CNC machine, for each type of tasks and each scenario. For example, the non-conformity management task had an average tardiness of 0.15 minutes in the *StaticAllocation* scenario.

Table 15: Average tardiness in minutes per task type for the 108 datasets for each scenario. Each dataset represents an hour of production.

		Tardiness (minutes)					
		StaticAllocation		DynamicAllocationCurrentSkills		DynamicAllocationAllSkills	
		Average	Std. Dev.	Average	Std. Dev.	Average	Std. Dev.
Task type	Critical tasks	21.13	24.55	2.94	8.32	0.23	1.74
	Non-conformity management	0.15	0.49	0.03	0.29	0.00	0.00
	Raw material addition	5.86	10.25	1.60	5.70	0.23	1.74
	Set up	10.64	18.09	1.00	5.19	0.00	0.00
	Tool change	4.48	8.93	0.31	1.36	0.00	0.00
	Parts measuring	101.28	78.41	26.30	59.92	0.85	2.34
	Total	122.41	83.82	29.24	62.40	1.08	2.85

For the scenario *DynamicAllocationAllSkills*, the total average tardiness (bottom line in Table 15) is extremely low with about one minute of tardiness per instance. This represents a diminution of 96.3% from the *StaticAllocation* scenario This scenario clearly yields an optimal solution in which human operators are not responsible for delays in the production.

The scenario *DynamicAllocationCurrentSkills* manages to generate 76.1% less tardiness than the scenario *StaticAllocation*. Distributed over the 25 CNC machines, we obtain an average total tardiness of almost 5 minutes per machine over the 108 instances yielded by the scenario *StaticAllocation*, compared to 1.2 minutes per machine in the *DynamicAllocationCurrentSkills* scenario (we recall each instance represents an hour of production).

Also, Table 15 highlights that the improvement obtained with the scenario *DynamicAllocationCurrentSkills* is distributed over every task type. While raw material addition tasks are the ones that get the smallest improvement with a diminution of 73% in tardiness, this scenario manages to reduce tardiness in tool changes tasks by 93%. Every task type gets an improvement when going from the *StaticAllocation* to the *DynamicAllocationCurrentSkills* scenario. Finally, critical tasks benefit from almost twice as much improvement as non-critical tasks (parts measuring) in scenario *DynamicAllocationCurrentSkills*.

Having workers dynamically allocated to tasks coming from different CNC machines during their work shift leads to less overall tardiness for most instances in a typical week of production. While current skill levels at the industrial partner site allowed to decrease human caused tardiness by almost four times, completely versatile operators would lead to a (theoretical) complete removal of human induced delays in production.

Given the important standard-deviation reported in Table 15, we analysed thoroughly the comparison between the solutions obtained in each instance. The *StaticAllocation* scenario is outperformed by the *DynamicAllocationCurrentSkills* scenario for 89 instances. The *DynamicAllocationCurrentSkills* is only outperformed in 45 instances by the *DynamicAllocationAllSkills* scenario. Furthermore, 103 instances have less tardiness in the *DynamicAllocationAllSkills* than in the *StaticAllocation*. This means that in five instances, the three scenarios obtained solutions yielding the same amount of tardiness. The transition from the current scenario to the dynamic allocation with the current skills of operators allows an improvement in twice as many instances as the training of all operators does. No instance obtained a better solution in a scenario Y than in a scenario X, at best they obtained equivalent solutions.

Table 16 reports tardiness aggregated per resource type.

Table 16: Average tardiness by resource type. Each dataset represents an hour of production.

		Tardiness (minutes)					
		StaticAllocation		DynamicAllocationCurrentSkills		DynamicAllocationAllSkills	
		Average	Std. Dev.	Average	Std. Dev.	Average	Std. Dev.
Resource type	Operators	122.16	83.82	28.95	62.48	0.77	2.80
	CMM	0.25	0.74	0.29	0.75	0.31	0.73
	Total	122.41	83.82	29.24	62.40	1.08	2.85

In both the *StaticAllocation* and *DynamicAllocationCurrentSkills* scenarios, human operators are responsible for 99% of tardiness on average. In the *DynamicAllocationAllSkills* scenario, CMMs take on 30% of the responsibility.

While operators represent a clear bottleneck in all scenarios, Figure 35 shows that their average occupation rate is quite low. In the *StaticAllocation* scenario, operators process tasks for less than 40% of their work shift, while CMMs are used 85% of the time. The operators average occupation rate rises over 50% in the *DynamicAllocationCurrentSkills* and *DynamicAllocationAllSkills* scenarios. The CMM average occupation rate diminishes slightly.

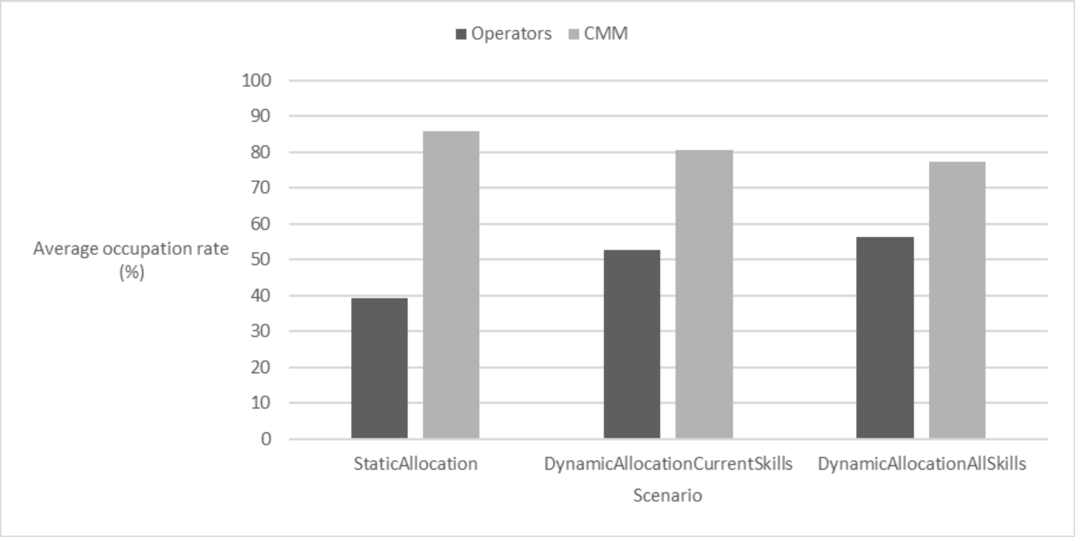


Figure 35: Average occupation rates (%) for the different types of resource under the proposed scenarios.

Because of the nature of the tasks at hand with precise and spread-out release times, operators have important idle time. This observation is in line with the ones made in the production factory. While idle time diminishes in both improving scenarios, resources are still not used to their full potential, with too many operators to handle the machines. This suggests the machine/operator ratio could increase in order to diminish operators' idle time. However, since CMM idle time is significantly lower than the operators', increasing the number of CNC machines in the factory could result in CMMs becoming the limiting resource and causing tardiness. Figure 3 also illustrates that CMM occupation rate diminishes in the dynamic allocation scenarios since they do not need to process as many tasks in the production plan to compensate for the lack of operator resources and mitigate the tardiness.

Figure 36 shows the distribution of the occupation rates throughout the work week in each scenario. For example, in the *StaticAllocation*, 22% of the hours worked had an occupation rate of [0%,10%]. We can see that more than 20% of worked hours are only productive for less than 10% of the time, while another 20% is productive for more than 90% of the time. We notice that when operators can process a higher number of tasks (that is, going from scenario *StaticAllocation* to scenarios *DynamicAllocationCurrentSkills* and then *DynamicAllocationAllSkills*), productive work hours with an occupation rate ranging between 40% and 90% are more frequent. Extremely productive hours of over 90% occupation rates are higher than 20% in all three scenarios.

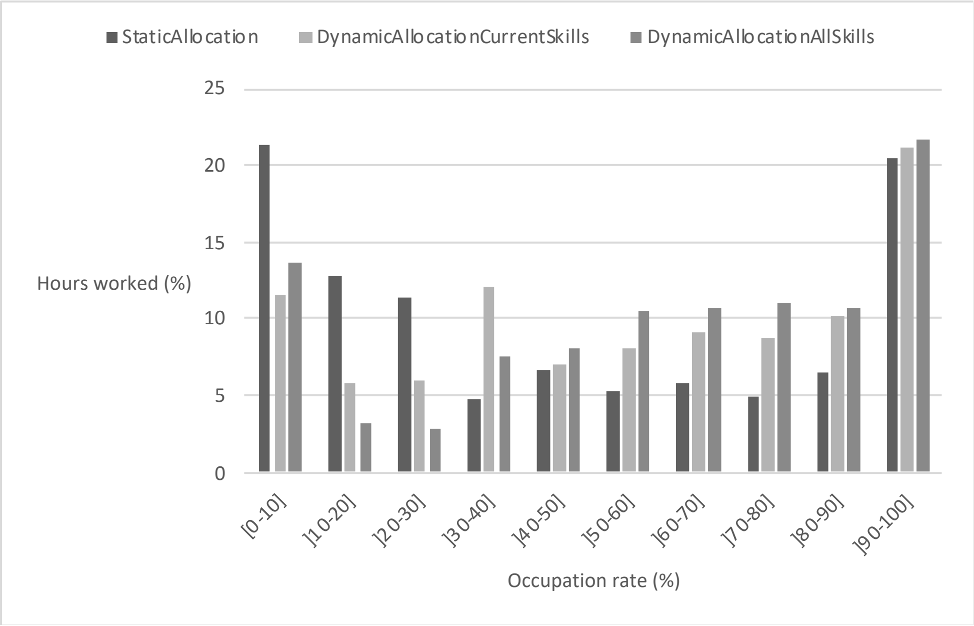


Figure 36: Distribution of the occupation rates (%) for the hours worked (%) under the different scenarios.

An important factor of a resource allocation plan is its fairness between operators. For every work hour, tasks should be distributed evenly between operators so as not to have overworked operators while others are underworked. The distribution of the utilization time during an hour for every operator gives more information on the distribution of the workload amongst employees. The phenomenon in which work hours are very unproductive (less than 10% occupation rate) happens for different reasons in each scenario. In the *StaticAllocation* scenario, operators allocated to CNC machine(s) processing jobs with small task processing times and long parts cycle times have a low occupation rate. While production managers try to even out the schedule when allocating operators to CNC machines, this situation may still arise.

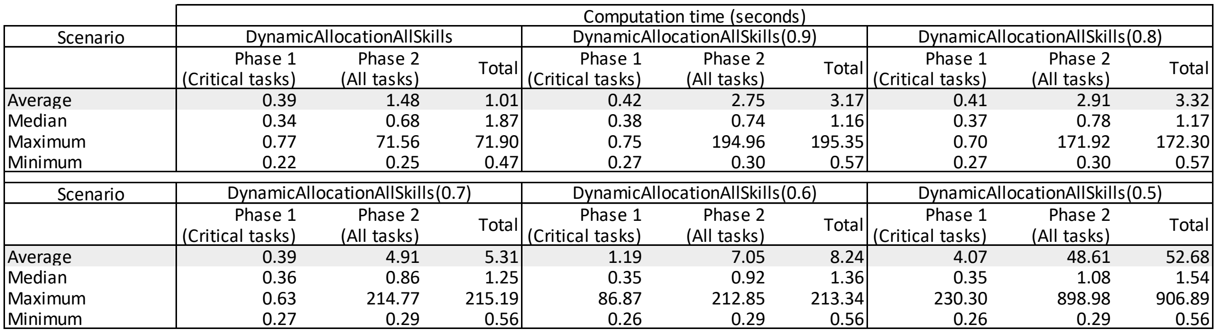
While non-urgent tasks, such as preventive maintenance or cleaning, can be allocated to these underused resources, this poor allocation still leads to more important tasks being late. In the *DynamicAllocationCurrentSkills* scenario, operators that are very versatile may be over allocated to tasks while new operators with less skills are underused. In the *DynamicAllocationAllSkills* scenario, which assumes that every operator is completely versatile, the task is allocated to the first operator at the top of the list, leading to an underutilization of operators at the bottom of the list. In the *StaticAllocation* scenario, production managers that allocate CNC machines to operators try to allocate CNC machines fairly between them, but this task is hard given the limited amount of information they have to make this decision.

Effect of the staffing level

To observe the relationship between the tardiness generated and the staffing level, an additional experiment was conducted. With the *DynamicAllocationAllSkills* as the base scenario, the number of available operators was reduced by 10%, 20%, 30% and 50% to observe how a reduction of human resources would affect tardiness. The *DynamicAllocationAllSkills* scenario was used as a basis since all operators have the same skill set, so there would be no difference between removing one operator or another. The number of CNC machines was not modified from the base scenario (*DynamicAllocationAllSkills*).[1]

Table 17 presents the results obtained in Experiment 2. Results for the original *DynamicAllocationAllSkills* scenario are presented again as a baseline.

Table 17: Computation time (seconds) for the 108 datasets for Experiment 2. The results are provided for the original DynamicAllocationAllSkills scenario, as well as scenarios for which workforce is reduced by 10 % to 50 %.



	Computation time (seconds)								
Scenario	DynamicAllocationAllSkills			DynamicAllocationAllSkills(0.9)			DynamicAllocationAllSkills(0.8)		
	Phase 1 (Critical tasks)	Phase 2 (All tasks)	Total	Phase 1 (Critical tasks)	Phase 2 (All tasks)	Total	Phase 1 (Critical tasks)	Phase 2 (All tasks)	Total
Average	0.39	1.48	1.01	0.42	2.75	3.17	0.41	2.91	3.32
Median	0.34	0.68	1.87	0.38	0.74	1.16	0.37	0.78	1.17
Maximum	0.77	71.56	71.90	0.75	194.96	195.35	0.70	171.92	172.30
Minimum	0.22	0.25	0.47	0.27	0.30	0.57	0.27	0.30	0.57
Scenario	DynamicAllocationAllSkills(0.7)			DynamicAllocationAllSkills(0.6)			DynamicAllocationAllSkills(0.5)		
	Phase 1 (Critical tasks)	Phase 2 (All tasks)	Total	Phase 1 (Critical tasks)	Phase 2 (All tasks)	Total	Phase 1 (Critical tasks)	Phase 2 (All tasks)	Total
Average	0.39	4.91	5.31	1.19	7.05	8.24	4.07	48.61	52.68
Median	0.36	0.86	1.25	0.35	0.92	1.36	0.35	1.08	1.54
Maximum	0.63	214.77	215.19	86.87	212.85	213.34	230.30	898.98	906.89
Minimum	0.27	0.29	0.56	0.26	0.29	0.56	0.26	0.29	0.56

[1] With the reduction of available resources, instances became increasingly difficult to solve and we set the time limit for the optimization model to fifteen minutes for each phase.

As the number of operators decreases, the average time to solve phase 2 of the optimization model increases. The scenario with *DynamicAllocationAllSkills(0.5)* operators represents a challenge to solve for the current optimization model, with its average of around one minute to solve both phases and a maximum reaching the time limit in phase 2. Apart from this harder scenario, all other scenarios have resolution times that may be considered real-time.

Figure 37 shows the relationship between tardiness and workforce for critical and non-critical tasks. Tardiness increases exponentially with the reduction of operators, reaching an average tardiness of over 25 minutes when half the operators are removed. Critical tasks are less affected by the changes of scenario than the parts measuring tasks. Indeed, parts measuring tasks are 25 times more delayed in the scenario *DynamicAllocationAllSkills(0.5)* than in the base scenario while critical tasks are only 16 times more delayed.

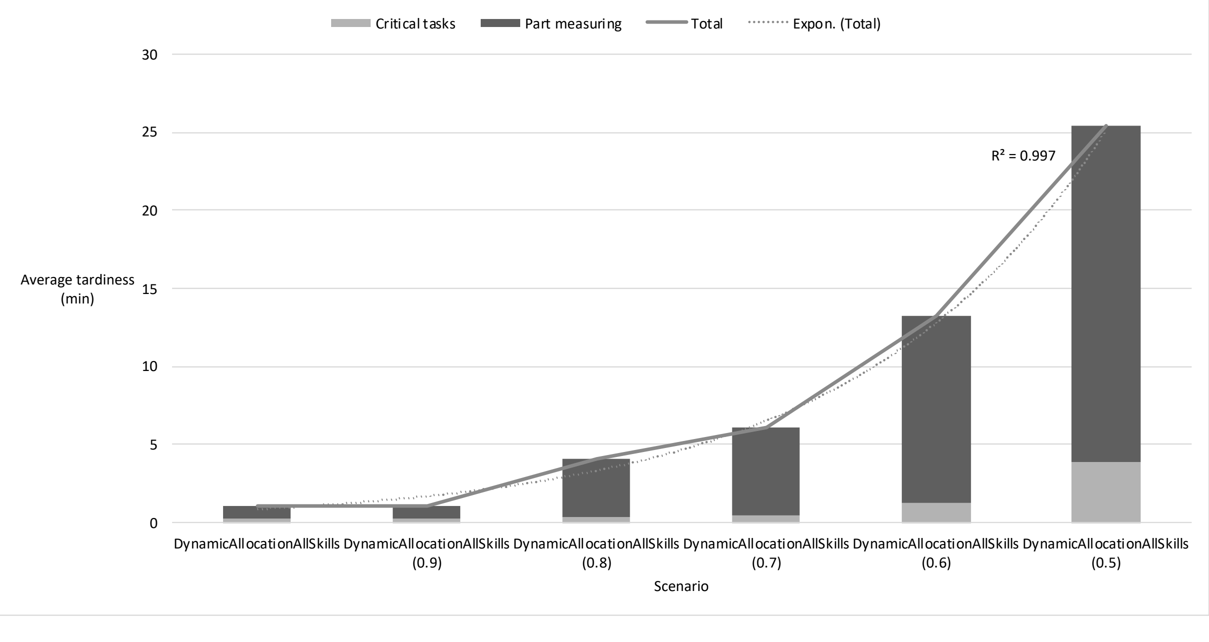


Figure 37: Average tardiness (min) according to workforce for critical and non-critical tasks.

Removing 10% of the operators every hour would not impact the tardiness. This means that with their current 9.9 operators on average (Table 6), if operators were completely versatile as supposed in the *DynamicAllocationAllSkills* scenario, the industrial partner could operate on average 18.3 machines instead of their current 15.9, without creating order delays. Average tardiness over the 108 instances increases exponentially as the number of operators is reduced.

Figure 38 shows how the occupation rate of the operators rises with the gradual reduction of operators available. It goes from 56% to 82%, steadily rising in each scenario where less operators are present to manage the tasks. It is interesting to notice that CMM average occupation rate rises as well, although less drastically than for the operators.

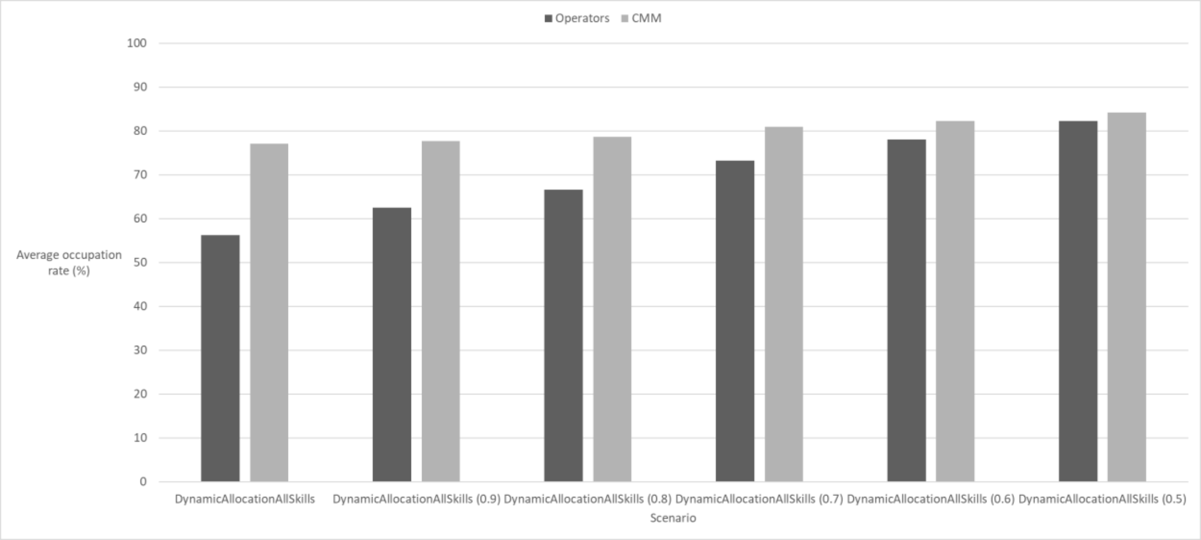


Figure 38: Average occupation rate according to workforce.

Figure 39 illustrates that the distributions are more and more decentralized to the right (increase occupation rate, less non-productive work hours) as workforce is reduced. Indeed, removing operators almost completely eliminates non-productive [0,10] work hours while those with a]90,100] occupation rate rise steadily, representing a larger proportion of total occurrences. Overworked human resources might lead to the undesirable effects of worker fatigue (Li, Xu and Fu, 2020) and would not be sustainable.

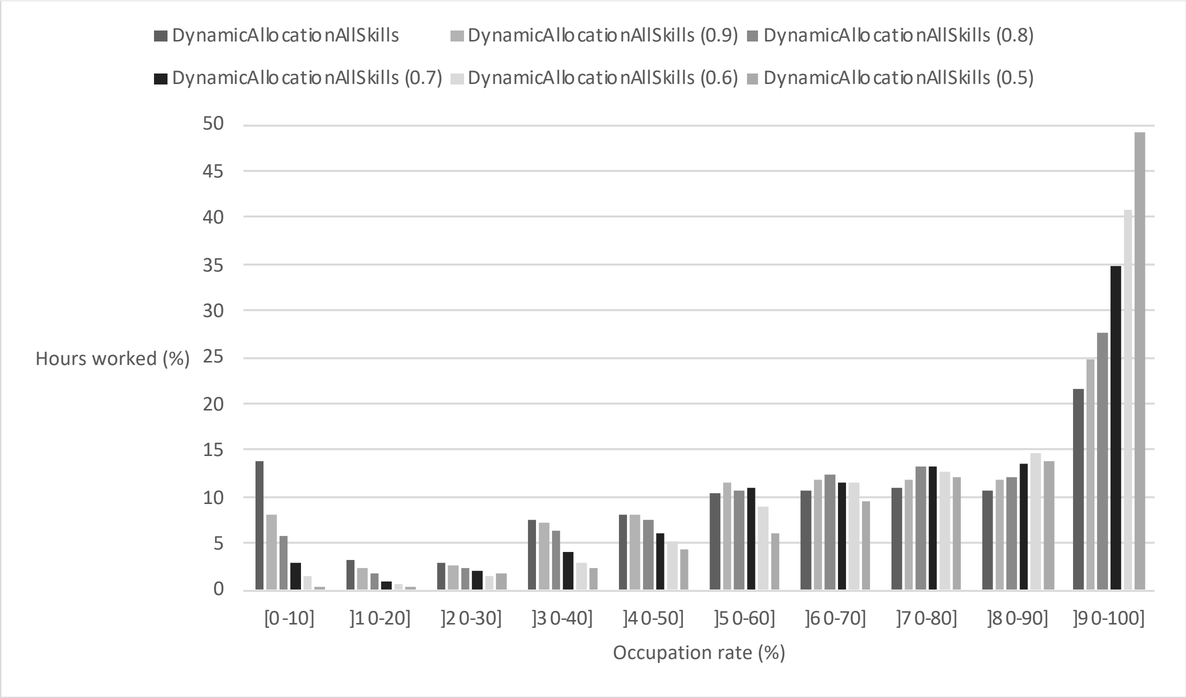


Figure 39: Distribution of the operators occupation rate according to manpower.

Figure 40 is presented in order to compare the *DynamicAllocationCurrentSkills* scenario from the first experiment with the *DynamicAllocationAllSkills(0.5)* scenario.

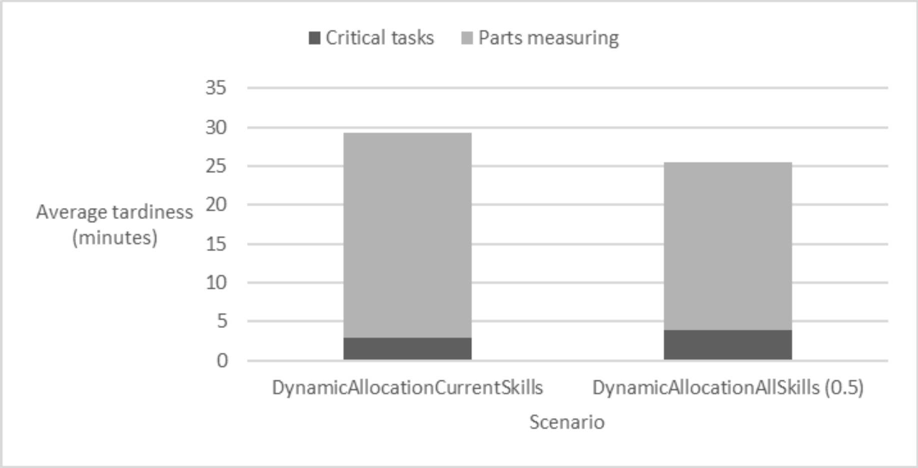


Figure 40: Comparison between the DynamicAllocationCurrentSkills and the DynamicAllocationAllSkills(0.5) scenarios.

The *DynamicAllocationCurrentSkills* scenario with the current level of skills the operators have is almost equivalent to having half the number of the operators in the *DynamicAllocationAllSkills(0.5)* scenario, when they are considered fully skilled. This leads us to conclude that in a dynamic allocation system, the more the operators receive appropriate training, the less the number of operators are needed in the factory to operate the same amount of CNC machines. The firm may then decrease the staffing level given that the operators are optimally allocated and appropriately skilled.

Managerial insights

As the results showed, implementing a dynamic allocation of operators in the metal manufacturing job-shop factory considered could be profitable. It becomes therefore interesting to think about the different steps to follow to make possible such an implementation. First, data on the upcoming jobs (on a longer time span of about a week) would need to be extracted in order to create a list of the upcoming tasks that would need to be allocated. In addition, data concerning the processing times for task as well as employee schedules would have to be obtained. This data collection can be a hard task for a company since it is not typically readily available for collection. Then, real-time data would need to be collected, which would serve to make more precise the list of upcoming tasks for the next hour to come. Real-time data would encompass data coming directly from the CNC machines, making precise the part number in process at the time of the extraction, as well as data concerning stochastic events such as non-conformities. A user interface dashboard could be used to show the operators the upcoming tasks in real-time. The list of tasks would then serve as an input in the optimization model proposed in this article. A plan of operator-task allocation would then be proposed for the next hour and displayed on the dashboard. We propose regenerating a new plan every five minutes, or at any frequency determined optimal, in order to take into account stochastic events and make sure that the list of tasks to allocate is always up-to-date.

Conclusion

We proposed and showed the advantages of dynamic human resources allocation in the metal manufacturing sector in an Industry 4.0 context. This is achieved through an optimization model that could serve as a task manager in the factory in lieu of the current production manager. The optimization model succeeded in solving instances quickly enough to be used in a real-time system directly in the factory. Results showed that the proposed dynamic allocation, under its simpler form (using the skills that the operators already have), could lead to an average reduction of 76% in tardiness on the CNC machines. When considering only critical tasks, that is, tasks that directly delay parts production, their tardiness could decrease by 86%. The scenario supposing perfectly multi-skilled operators practically eliminated human caused tardiness in every instance.

An additional experiment helped characterize the relationship between the staffing level, that is, the number of employees per CNC machine in the factory and the tardiness. We showed that it would be possible to operate on average 2.4 additional machines with the same quantity of operators in a scenario where operators are dynamically allocated and completely versatile, without a significant impact on the overall tardiness.

The results obtained with these experiments indicate that the current policies on task allocation and training used by the industrial partner should be revised. Given the recurrent global labour shortage (Gruzauskas, 2016), we believe our model could help them in optimizing the use of their available human resources and increase the productivity of their automated equipment. Such a model could also be useful for systems where tasks are released in a periodic and stochastic manner. This type of task assignment has gained popularity in crowd-sourcing type applications, such as Uber (Song *et al.*, 2020).

It is important to mention that predicting the occurrence of tasks in real-time required a lot of data transformation, since the data were not available in this exact format. With various information extracted from the company, our team was able to extract a prediction of when the next tasks would occur for each instance. Implementation of a dashboard with the upcoming tasks to execute is currently in progress in the factory, with promising results.

For future research, it would be interesting to see which skills lead to the biggest improvements in terms of productivity of the factory to help guide workforce training strategies. We also suggest working on the fairness of the generated task allocation plan, as well as limiting the occupation rate of operators so as to not overwork the most versatile operators.

References

Afshar-Nadjafi, B. (2021) ‘Multi-skilling in scheduling problems: A review on models, methods and applications’, *Computers & Industrial Engineering*, 151, p. 107004. doi:10.1016/j.cie.2020.107004.

Agnetis, A., Murgia, G. and Sbrilli, S. (2014) ‘A job shop scheduling problem with human operators in handicraft production’, *International Journal of Production Research*, 52(13), pp. 3820–3831. doi:10.1080/00207543.2013.831220.

Alade, O.M. and Amusat, A.O. (2019) ‘Solving Nurse Scheduling Problem Using Constraint Programming Technique’, *arXiv:1902.01193 [cs]* [Preprint]. Available at: http://arxiv.org/abs/1902.01193 (Accessed: 4 August 2021).

Al-Behadili, M., Ouelhadj, D. and Jones, D. (2019) ‘Multi-objective Biased Randomised Iterated Greedy for Robust Permutation Flow Shop Scheduling Problem under Disturbances’, *Journal of the Operational Research Society* [Preprint]. doi:10.1080/01605682.2019.1630330.

Allgeier, H. *et al.* (2020) ‘Simulation-Based Evaluation of Lot Release Policies in a Power Semiconductor Facility - a Case Study’, in *2020 Winter Simulation Conference (WSC), 14-18 Dec. 2020*. Piscataway, NJ, USA: IEEE (Proceedings of the 2020 Winter Simulation Conference (WSC)), pp. 1503–14. doi:10.1109/WSC48552.2020.9384094.

Apt, K. (2003) *Principles of Constraint Programming*. Cambridge University Press.

Artigues, C. *et al.* (2009) ‘Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound’, *Computers and Operations Research*, 36(8), pp. 2330–2340. doi:10.1016/j.cor.2008.08.013.

Banks, J. *et al.* (2005) *Discrete-Event System Simulation*. Prentice-Hall. Available at: http://www.worldcat.org/oclc/55847249 (Accessed: 26 January 2022).

Bär, K., Herbert-Hansen, Z.N.L. and Khalid, W. (2018) ‘Considering Industry 4.0 aspects in the supply chain for an SME’, *Production Engineering*, 12(6), pp. 747–758. doi:10.1007/s11740-018-0851-y.

Barlas, P. and Heavey, C. (2016) ‘Automation of input data to discrete event simulation for manufacturing: A review’, *International Journal of Modeling, Simulation, and Scientific Computing*, 07, p. 1630001. doi:10.1142/S1793962316300016.

Beauchemin, M. *et al.* (2020) ‘Evaluating workers allocation policies through the simulation of a high-precision machining workshop’, in *Winter Simulation Conference*.

Bhulai, S., Koole, G. and Pot, A. (2008) ‘Simple Methods for Shift Scheduling in Multiskill Call Centers’, *Manufacturing & Service Operations Management*, 10(3), pp. 411–420. doi:10.1287/msom.1070.0172.

Borreguero-Sanchidrián, T. *et al.* (2018) ‘Flexible Job Shop Scheduling With Operators in Aeronautical Manufacturing: A Case Study’, *IEEE Access*, 6, pp. 224–233. doi:10.1109/ACCESS.2017.2761994.

Borshchev, A. (2014) ‘Multi-method modelling: AnyLogic’, in, pp. 248–279. doi:10.1002/9781118762745.ch12.

Boschert, S., Heinrich, C. and Rosen, R. (2018) *Next Generation Digital Twin*.

Boschert, S. and Rosen, R. (2016) ‘Digital Twin—The Simulation Aspect’, in Hehenberger, P. and Bradley, D. (eds) *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers*. Cham: Springer International Publishing, pp. 59–74. doi:10.1007/978-3-319-32156-1_5.

Bouajaja, S. and Dridi, N. (2017) ‘A survey on human resource allocation problem and its applications’, *Operational Research*, 17(2), pp. 339–369. doi:10.1007/s12351-016-0247-8.

Bourdais, S., Galinier, P. and Pesant, G. (2003) ‘hibiscus: A Constraint Programming Application to Staff Scheduling in Health Care’, in Rossi, F. (ed.) *Principles and Practice of Constraint Programming – CP 2003*. Berlin, Heidelberg: Springer (Lecture Notes in Computer Science), pp. 153–167. doi:10.1007/978-3-540-45193-8_11.

Brik, B. *et al.* (2019) *Accuracy and Localization-Aware Rescheduling for Flexible Flow Shops in Industry 4.0*. doi:10.1109/CoDIT.2019.8820445.

Bülbül, K. and Kaminsky, P. (2013) ‘A linear programming-based method for job shop scheduling’, *Journal of Scheduling*, 16(2), pp. 161–183. doi:10.1007/s10951-012-0270-4.

Butt, J. (2020) ‘A Strategic Roadmap for the Manufacturing Industry to Implement Industry 4.0’, *Designs*, 4(2), p. 11. doi:10.3390/designs4020011.

Caprara, A., Monaci, M. and Toth, P. (2003) ‘Models and algorithms for a staff scheduling problem’, *Mathematical Programming*, 98(1), pp. 445–476. doi:10.1007/s10107-003-0413-7.

Carlier, J. (1982) ‘The one-machine sequencing problem’, *European Journal of Operational Research*, 11(1), pp. 42–47. doi:10.1016/S0377-2217(82)80007-6.

Costa, A., Cappadonna, F.A. and Fichera, S. (2014) ‘Joint optimization of a flow-shop group scheduling with sequence dependent set-up times and skilled workforce assignment’, *International Journal of Production Research*, 52(9), pp. 2696–2728. doi:10.1080/00207543.2014.883469.

Costigliola, A. *et al.* (2017) ‘Simulation Model of a Quality Control Laboratory in Pharmaceutical Industry’, *IFAC-PapersOnLine*, 50(1), pp. 9014–9019. doi:10.1016/j.ifacol.2017.08.1582.

Cunha, M.M. *et al.* (2019) ‘Dual Resource Constrained Scheduling for Quality Control Laboratories’, *IFAC-PapersOnLine*, 52(13), pp. 1421–1426. doi:10.1016/j.ifacol.2019.11.398.

Dantzig, G.B. and Wolfe, P. (1960) ‘Decomposition Principle for Linear Programs’, *Operations Research*, 8(1), pp. 101–111. doi:10.1287/opre.8.1.101.

Deming, W., Neumann, J. and Morgenstern, O. (1944) ‘Theory of Games and Economic Behavior’. doi:10.2307/2280142.

Demir, Y. and Kürşat İşleyen, S. (2013) ‘Evaluation of mathematical models for flexible job-shop scheduling problems’, *Applied Mathematical Modelling*, 37(3), pp. 977–988. doi:10.1016/j.apm.2012.03.020.

Detty, R.B. and Yingling, J.C. (2000) ‘Quantifying benefits of conversion to lean manufacturing with discrete event simulation: A case study’, *International Journal of Production Research*, 38(2), pp. 429–445. doi:10.1080/002075400189509.

Devezas, T., Leitão, J. and Sarygulov, A. (eds) (2017) *Industry 4.0: Entrepreneurship and Structural Change in the New Digital Landscape*. Cham: Springer International Publishing (Studies on Entrepreneurship, Structural Change and Industrial Dynamics). doi:10.1007/978-3-319-49604-7.

Edi, K.H. and Duquenne, P. (2009) ‘Intégration de la polyvalence et de la modulation d’horaire dans une approche d’affectation flexible de la ressource humaine’, *revue ivoirienne des sciences et technologies*, pp. 1–20.

Eiselt, H.A. and Marianov, V. (2008) ‘Employee positioning and workload allocation’, *Computers & Operations Research*, 35, pp. 513–524. doi:10.1016/j.cor.2006.03.014.

Endsley, M. (1995) ‘Toward a Theory of Situation Awareness in Dynamic Systems’, *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37, pp. 32–64. doi:10.1518/001872095779049543.

Eriksson, S. (2020) ‘Optimal Multi-Skilled Workforce Scheduling for Contact Centers Using Mixed Integer Linear Programming’. KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF ENGINEERING SCIENCES.

Erming Zhou, Jin Zhu, and Ling Deng (2017) ‘Flexible job-shop scheduling based on genetic algorithm and simulation validation’, in *13th Global Congress on Manufacturing and Management (GCMM 2016), 28-30 Nov. 2016. MATEC Web of Conferences*, France: EDP Sciences (MATEC Web Conf. (France)), p. 02047 (8 pp.). doi:10.1051/matecconf/201710002047.

Evangelou, G. *et al.* (2021) ‘An approach for task and action planning in Human–Robot Collaborative cells using AI’, *Procedia CIRP*, 97, pp. 476–481. doi:10.1016/j.procir.2020.08.006.

Fattahi, P., Jolai, F. and Arkat, J. (2009) ‘Flexible job shop scheduling with overlapping in operations’, *Applied Mathematical Modelling*, 33(7), pp. 3076–3087. doi:10.1016/j.apm.2008.10.029.

Fattahi, P., Saidi Mehrabad, M. and Jolai, F. (2007) ‘Mathematical modeling and heuristic approaches to flexible job shop scheduling problems’, *Journal of Intelligent Manufacturing*, 18(3), pp. 331–342. doi:10.1007/s10845-007-0026-8.

Ferjani, A. *et al.* (2017) ‘A simulation-optimization based heuristic for the online assignment of multi-skilled workers subjected to fatigue in manufacturing systems’, *Computers & Industrial Engineering*, 112, pp. 663–674. doi:10.1016/j.cie.2017.02.008.

Fourer, R. (no date) ‘AMPL: A Mathematical Programming Language’, p. 65.

Framinan, J., Fernandez-Viagas, V. and Perez-Gonzalez, P. (2019) ‘Using real-time information to reschedule jobs in a flowshop with variable processing times’, *Computers & Industrial Engineering*, 129. doi:10.1016/j.cie.2019.01.036.

Garey, M.R., Johnson, D.S. and Sethi, R. (1976) ‘The Complexity of Flowshop and Jobshop Scheduling’, *Mathematics of Operations Research*, 1(2), pp. 117–129.

Garrido, J. and Sáez, J. (2019) ‘Integration of automatic generated simulation models, machine control projects and management tools to support whole life cycle of industrial digital twins.’, *IFAC-PapersOnLine*, 52(13), pp. 1814–1819. doi:10.1016/j.ifacol.2019.11.465.

Ghaleb, M., Taghipour, S. and Zolfagharinia, H. (2020) ‘Real-Time Optimization of Maintenance and Production Scheduling for an Industry 4.0-Based Manufacturing System’, in *2020 Annual Reliability and Maintainability Symposium (RAMS). 2020 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–8. doi:10.1109/RAMS48030.2020.9153721.

Ghaleb, M., Zolfagharinia, H. and Taghipour, S. (2020) ‘Real-time production scheduling in the Industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns’, *Computers & Operations Research*, 123, p. 105031. doi:10.1016/j.cor.2020.105031.

Ghobakhloo, M. (2018) ‘The future of manufacturing industry: a strategic roadmap toward Industry 4.0’, *Journal of Manufacturing Technology Management*, 29(6), pp. 910–936. doi:10.1108/JMTM-02-2018-0057.

Gleixner, A. (2018) ‘Computational Mixed-Integer Programming’.

Goldsman, D., Nance, R. and Wilson, J. (2010) *A brief history of simulation*, *Proceedings - Winter Simulation Conference*, p. 313. doi:10.1109/WSC.2009.5429341.

Greis, N.P. *et al.* (2019) ‘Manufacturing-Uber: Intelligent Operator Assignment in a Connected Factory’, *IFAC-PapersOnLine*, 52(13), pp. 2734–2739. doi:10.1016/j.ifacol.2019.11.621.

Grieco, A. *et al.* (2017) ‘An Industry 4.0 Case Study in Fashion Manufacturing’, *Procedia Manufacturing*, 11, pp. 871–877. doi:10.1016/j.promfg.2017.07.190.

Gruzauskas, V. (2016) ‘Labour and machine efficient utilization importance to the enterprise profit’, *Journal of Management*, 28, pp. 127–125.

Gunal, M.M. (ed.) (2019) *Simulation for Industry 4.0: Past, Present, and Future*. Cham: Springer International Publishing (Springer Series in Advanced Manufacturing). doi:10.1007/978-3-030-04137-3.

Ham, A. and Cakici, E. (2016) ‘Flexible job shop scheduling problem with parallel batch processing machines: MIP and CP approaches’, *Computers & Industrial Engineering*, 102. doi:10.1016/j.cie.2016.11.001.

Ham, A., Park, M.-J. and Kim, K.M. (2021) ‘Energy-Aware Flexible Job Shop Scheduling Using Mixed Integer Programming and Constraint Programming’, *Mathematical Problems in Engineering*, 2021, p. e8035806. doi:10.1155/2021/8035806.

Hammann, J.E. and Markovitch, N.A. (1995) ‘Introduction to Arena [simulation software]’, in *Winter Simulation Conference Proceedings, 1995. Winter Simulation Conference Proceedings, 1995.*, pp. 519–523. doi:10.1109/WSC.1995.478785.

Hamrol, A. *et al.* (2018) ‘Analysis of the Conditions for Effective Use of Numerically Controlled Machine Tools’, in Hamrol, A. et al. (eds) *Advances in Manufacturing*. Cham: Springer International Publishing (Lecture Notes in Mechanical Engineering), pp. 3–12. doi:10.1007/978-3-319-68619-6_1.

Haralick, R.M. and Elliott, G.L. (1980) ‘Increasing tree search efficiency for constraint satisfaction problems’, *Artificial Intelligence*, 14(3), pp. 263–313. doi:10.1016/0004-3702(80)90051-X.

Harmonosky, C.M. and Robohn, S.F. (1991) ‘Real-time scheduling in computer integrated manufacturing: a review of recent research’, *International Journal of Computer Integrated Manufacturing*, 4(6), pp. 331–340. doi:10.1080/09511929108944511.

Henning, K. (2013) *Recommendations for implementing the strategic initiative INDUSTRIE 4.0*. Available at: https://www.semanticscholar.org/paper/Recommendations-for-implementing-the-strategic-4.0-Henning/80d4b3eccd6951cf6dba61f1b33889a0edbf0407 (Accessed: 1 February 2022).

Ho, T.-W. *et al.* (2018) ‘A Platform for Dynamic Optimal Nurse Scheduling Based on Integer Linear Programming along with Multiple Criteria Constraints’, in *Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference on ZZZ - AICCC ’18. the 2018 Artificial Intelligence and Cloud Computing Conference*, Tokyo, Japan: ACM Press, pp. 145–150. doi:10.1145/3299819.3299825.

van Hoeve, W.-J. and Katriel, I. (2006) ‘Global constraints’, in.

Hollocks, B. (1992) ‘A well-kept secret?’, *OR Insight*, 5(4), pp. 12–17. doi:10.1057/ori.1992.29.

Hughes, K. and Jiang, X. (2010) ‘Using discrete event simulation to model excavator operator performance’, *Human Factors and Ergonomics in Manufacturing & Service Industries*, 20(5), pp. 408–423. doi:10.1002/hfm.20191.

Huynh, B.H., Akhtar, H. and Li, W. (2020) ‘Discrete Event Simulation for Manufacturing Performance Management and Optimization: A Case Study for Model Factory’, in *2020 9th International Conference on Industrial Technology and Management (ICITM)*. *2020 9th International Conference on Industrial Technology and Management (ICITM)*, pp. 16–20. doi:10.1109/ICITM48982.2020.9080394.

Idrisov, G.I. *et al.* (2018) ‘New technological revolution: Challenges and opportunities for Russia’, *Voprosy Ekonomiki*, pp. 5–25. doi:10.32609/0042-8736-2018-4-5-25.

Ingemansson, A., Bolmsjö, G.S. and Harlin, U. (2002) ‘A Survey of the Use of the Discrete-Event Simulation in Manufacturing Industry’, p. 5.

Ivanov, V. *et al.* (2019) ‘Technology for complex parts machining in multiproduct manufacturing’, *Management and Production Engineering Review*, Vol. 10, No. 2, pp. 25–36. doi:10.24425/mper.2019.129566.

Jung, W.-K. *et al.* (2020) ‘Real-time data-driven discrete-event simulation for garment production lines’, *Production Planning & Control*, 0(0), pp. 1–12. doi:10.1080/09537287.2020.1830194.

Kagermann, H. (2013) *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry ; Final Report of the Industrie 4.0 Working Group*. Forschungsunion.

Keller, J. (2002) ‘Human performance modeling for discrete-event simulation: workload’, in *Proceedings of the Winter Simulation Conference*. *Proceedings of the Winter Simulation Conference*, pp. 157–162 vol.1. doi:10.1109/WSC.2002.1172879.

Kelley, James E. and Walker, Morgan R. (1959) ‘Critical-path planning and scheduling’, in. *IRE-AIEE-ACM computer conference*. Available at: https://dl-acm-org.acces.bibl.ulaval.ca/doi/abs/10.1145/1460299.1460318 (Accessed: 22 November 2021).

Khayal, O. (2018) *A review for Dynamic Scheduling in Manufacturing*. doi:10.13140/RG.2.2.15345.33129.

Kher, H.V. and Fry, T.D. (2001) ‘Labour flexibility and assignment policies in a job shop having incommensurable objectives’, *International Journal of Production Research*, 39(11), pp. 2295–2311. doi:10.1080/00207540110036704.

Kress, D. and Müller, D. (2019) ‘Mathematical Models for a Flexible Job Shop Scheduling Problem with Machine Operator Constraints ∗∗This work has been supported by the European Union and the state North Rhine-Westphalia through the European Fund for Regional Development (EFRD). It has been conducted as part of the project “EKPLO: Echtzeitnahes kollaboratives Planen und Optimieren” (EFRE-0800463).’, *IFAC-PapersOnLine*, 52(13), pp. 94–99. doi:10.1016/j.ifacol.2019.11.144.

Lan, C.-H. and Lan, T.-S. (2005) ‘A combinatorial manufacturing resource planning model for long-term CNC machining industry’, *The International Journal of Advanced Manufacturing Technology*, 26(9–10), pp. 1157–1162. doi:10.1007/s00170-004-2090-y.

Lanzarone, E. and Matta, A. (2014) ‘Robust nurse-to-patient assignment in home care services to minimize overtimes under continuity of care’, *Operations Research for Health Care*, 3(2), pp. 48–58. doi:10.1016/j.orhc.2014.01.003.

Lasi, H. *et al.* (2014) ‘Industry 4.0’, *Business & Information Systems Engineering*, 6(4), pp. 239–242. doi:10.1007/s12599-014-0334-4.

Law, A.M. and Kelton, W.D. (1991) *Simulation modeling and analysis*. 2nd ed. New york: McGraw-Hill (McGraw-Hill series in industrial engineering and management science).

Le, V.T. *et al.* (2013) ‘Dynamic control of skilled and unskilled labour task assignments’, in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 955–960. doi:10.1109/AIM.2013.6584217.

Leusin, M.E. *et al.* (2018) ‘Solving the Job-Shop Scheduling Problem in the Industry 4.0 Era’, *Technologies*, 6(4), p. 107. doi:10.3390/technologies6040107.

Li, K., Xu, S. and Fu, H. (2020) ‘Work-break scheduling with real-time fatigue effect and recovery’, *International Journal of Production Research*, 58(3), pp. 689–702. doi:10.1080/00207543.2019.1598600.

Lidberg, S., Pehrsson, L. and Ng, A.H.C. (2018) ‘Using aggregated discrete event simulation models and multi-objective optimization to improve real-world factories’, in *2018 Winter Simulation Conference, WSC 2018, December 9, 2018 - December 12, 2018*. Gothenburg, Sweden: Institute of Electrical and Electronics Engineers Inc. (Proceedings - Winter Simulation Conference), pp. 2015–2024. doi:10.1109/WSC.2018.8632337.

Lin, Y. and Schrage, L. (2009) ‘The global solver in the LINDO API’, *Optimization Methods and Software*, 24(4–5), pp. 657–668. doi:10.1080/10556780902753221.

Liu, F. *et al.* (2017) ‘On the Robust and Stable Flowshop Scheduling Under Stochastic and Dynamic Disruptions’, *IEEE Transactions on Engineering Management*, 64(4), pp. 539–553. doi:10.1109/TEM.2017.2712611.

Manne, A.S. (1960) ‘On the Job-Shop Scheduling Problem’, *Operations Research*, 8(2), pp. 219–223.

Matt, D.T., Modrák, V. and Zsifkovits, H. (eds) (2020) *Industry 4.0 for SMEs: Challenges, Opportunities and Requirements*. Springer Nature. doi:10.1007/978-3-030-25425-4.

Meisels, A. and Schaerf, A. (2003) ‘Modelling and Solving Employee Timetabling Problems’, *Annals of Mathematics and Artificial Intelligence*, 39(1), pp. 41–59. doi:10.1023/A:1024460714760.

Meng, L. *et al.* (2020) ‘Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem’, *Computers & Industrial Engineering*, 142, p. 106347. doi:10.1016/j.cie.2020.106347.

Munasingha, K. and Adikariwattage, V. (2020) ‘Discrete Event Simulation Method to Model Passenger Processing at an International Airport’, in *2020 Moratuwa Engineering Research Conference (MERCon)*. *2020 Moratuwa Engineering Research Conference (MERCon)*, pp. 401–406. doi:10.1109/MERCon50084.2020.9185370.

Nehme, C., Crandall, J. and Cummings, M. (2008) *Using Discrete-Event Simulation to Model Situational Awareness of Unmanned-Vehicle Operators*. Available at: https://www.semanticscholar.org/paper/Using-Discrete-Event-Simulation-to-Model-Awareness-Nehme-Crandall/cdb956db32c9e46158cf4305eef7c6534477c46e (Accessed: 31 January 2022).

Nethercote, N. *et al.* (2007) ‘MiniZinc: Towards a standard CP modelling language’, in *Proceedings of the 13th International Conference on the Principles and Practice of Constraint Programming*. *International Conference on Principles and Practice of Constraint Programming 2007*, Springer-Verlag London Ltd., pp. 529–543. Available at: https://research.monash.edu/en/publications/minizinc-towards-a-standard-cp-modelling-language (Accessed: 21 April 2021).

Oliveira, E., Smith, B.M. and Jt, L. (2000) *A Job-Shop Scheduling Model for the Single-Track Railway Scheduling Problem*. Problem, Research Report 2000.21, University of Leeds. OpenTrack Railway Technology, Railway Simulation. http://www.opentrack.ch.

Onggo, B., Hill, J. and Brooks, R. (2013) ‘A pilot survey on data identification and collection in simulation projects’, in. *Modelling and Simulation 2013 - European Simulation and Modelling Conference, ESM 2013*.

Özkul, A.O. *et al.* (2021) ‘An Implementation of Flexible Job Shop Scheduling Problem in a Metal Processing Company’, in Durakbasa, N.M. and Gençyılmaz, M.G. (eds) *Digital Conversion on the Way to Industry 4.0*. Cham: Springer International Publishing (Lecture Notes in Mechanical Engineering), pp. 817–830. doi:10.1007/978-3-030-62784-3_68.

Pabla, B.S. and Adithan, M. (1994) *CNC Machines*. New Age International.

Palombarini, J.A. and Martínez, E.C. (2019) ‘Closed-loop Rescheduling using Deep Reinforcement Learning’, *IFAC-PapersOnLine*, 52(1), pp. 231–236. doi:10.1016/j.ifacol.2019.06.067.

de Paula Ferreira, W., Armellini, F. and De Santa-Eulalia, L.A. (2020) ‘Simulation in industry 4.0: A state-of-the-art review’, *Computers & Industrial Engineering*, 149, p. 106868. doi:10.1016/j.cie.2020.106868.

Pegden, C.D. (2008) ‘Introduction to Simio’, in *2008 Winter Simulation Conference*. *2008 Winter Simulation Conference*, pp. 229–235. doi:10.1109/WSC.2008.4736072.

Pegden, C.D. (2009) ‘An Introduction to Simio® for Beginners’, p. 7.

Pfitzer, F. *et al.* (2018) ‘Event-Driven Production Rescheduling in Job Shop Environments’, in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. *2018*

IEEE 14th International Conference on Automation Science and Engineering (CASE), pp. 939–944. doi:10.1109/COASE.2018.8560523.

Popkova, E.G., Ragulina, Y.V. and Bogoviz, A.V. (eds) (2019) *Industry 4.0: Industrial Revolution of the 21st Century*. Cham: Springer International Publishing (Studies in Systems, Decision and Control). doi:10.1007/978-3-319-94310-7.

Rahmani, D. and Ramezanian, R. (2016) 'A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study', *Computers & Industrial Engineering*, 98, pp. 360–372. doi:10.1016/j.cie.2016.06.018.

Robinson, S. (2005) 'Discrete-event simulation: from the pioneers to the present, what next?', *Journal of the Operational Research Society*, 56(6), pp. 619–629. doi:10.1057/palgrave.jors.2601864.

Rohde, D. (2019) *Dynamic simulation of future integrated energy systems*.

Rossi, E.F., van Beek, P. and Walsh, T. (2006) 'Handbook of Constraint Programming', p. 969.

Salido, M.A. *et al.* (2017) 'Rescheduling in job-shop problems for sustainable manufacturing systems', *Journal of Cleaner Production*, 162, pp. S121–S132. doi:10.1016/j.jclepro.2016.11.002.

Sargent, R.G. (2004) 'Validation and verification of simulation models', in *Proceedings of the 36th conference on Winter simulation*. Washington, D.C.: Winter Simulation Conference (WSC '04), pp. 17–28.

Schlesinger (1979) 'Terminology for model credibility', *SIMULATION*, 32(3), pp. 103–104. doi:10.1177/003754977903200304.

Schmidt, J.W. and Taylor, R.E. (1970) *Simulation and analysis of industrial systems*. Homewood, Ill: R. D. Irwin (Irwin series in quantitative analysis for business).

Semini, M., Fauske, H. and Strandhagen, J.O. (2006) 'Applications of Discrete-Event Simulation to Support Manufacturing Logistics Decision-Making: A Survey', in *2006 Winter Simulation Conference*. *2006 Winter Simulation Conference*, pp. 1946–1953. doi:10.1109/WSC.2006.322979.

Sicong, T., Weng, W. and Shigeru, F. (2009) 'Scheduling of Worker Allocation in the Manual Labor Environment with Genetic Algorithm', *Lecture Notes in Engineering and Computer Science*, 2174.

Sierra, M.R., Mencía, C. and Varela, R. (2015) 'New schedule generation schemes for the job-shop problem with operators', *Journal of Intelligent Manufacturing*, 26(3), pp. 511–525. doi:10.1007/s10845-013-0810-6.

Song, T. *et al.* (2020) 'Multi-skill aware task assignment in real-time spatial crowdsourcing', *GeoInformatica*, 24(1), pp. 153–173. doi:10.1007/s10707-019-00351-4.

Sotskov, Yu.N. and Shakhlevich, N.V. (1995) ‘NP-hardness of shop-scheduling problems with three jobs’, *Discrete Applied Mathematics*, 59(3), pp. 237–266. doi:10.1016/0166-218X(95)80004-N.

Sreekara Reddy, M.B.S. *et al.* (2018) ‘An effective hybrid multi objective evolutionary algorithm for solving real time event in flexible job shop scheduling problem’, *Measurement*, 114, pp. 78–90. doi:10.1016/j.measurement.2017.09.022.

Trilling, L., Guinet, A. and Magny, D.L. (2006) ‘Nurse scheduling using integer linear programming and constraint programming’, *IFAC Proceedings Volumes*, 39(3), pp. 671–676. doi:10.3182/20060517-3-FR-2903.00340.

Tsarouchi, P., Makris, S. and Chryssolouris, G. (2016) ‘Human–robot interaction review and challenges on task planning and programming’, *International Journal of Computer Integrated Manufacturing*, 29(8), pp. 916–931. doi:10.1080/0951192X.2015.1130251.

Turker, A.K. *et al.* (2019) ‘A Decision Support System for Dynamic Job-Shop Scheduling Using Real-Time Data with Simulation’, *Mathematics*, 7(3), p. 278. doi:10.3390/math7030278.

Turner, C.J. *et al.* (2016) ‘Discrete Event Simulation and Virtual Reality Use in Industry: New Opportunities and Future Trends’, *IEEE Transactions on Human-Machine Systems*, 46(6), pp. 882–894. doi:10.1109/THMS.2016.2596099.

Uhlmann, I.R. and Frazzon, E.M. (2018) ‘Production rescheduling review: Opportunities for industrial integration and practical applications’, *Journal of Manufacturing Systems*, 49, pp. 186–193. doi:10.1016/j.jmsy.2018.10.004.

Vieira, G., Herrmann, J. and Lin, E. (2003) ‘Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods’, *J. Scheduling*, 6, pp. 39–62. doi:10.1023/A:1022235519958.

Vielma, J.P. (2015) ‘Mixed Integer Linear Programming Formulation Techniques’, *SIAM Review*, 57(1), pp. 3–57. doi:10.1137/130915303.

Walsh, T. (2001) ‘Stochastic Constraint Programming’, p. 7.

Waschneck, B. *et al.* (2016) ‘Production Scheduling in Complex Job Shops from an Industry 4.0 Perspective: A Review and Challenges in the Semiconductor Industry’, in *SAMI@iKNOW*.

Watson, J.-P. and Beck, J.C. (2008) ‘A Hybrid Constraint Programming / Local Search Approach to the Job-Shop Scheduling Problem’, in Perron, L. and Trick, M.A. (eds) *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Berlin, Heidelberg: Springer (Lecture Notes in Computer Science), pp. 263–277. doi:10.1007/978-3-540-68155-7_21.

Wichmann, R.L., Eisenbart, B. and Gericke, K. (2019) ‘The Direction of Industry: A Literature Review on Industry 4.0’, *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1), pp. 2129–2138. doi:10.1017/dsi.2019.219.

Xia, W. and Wu, Z. (2005) ‘An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems’, *Computers & Industrial Engineering*, 48(2), pp. 409–425. doi:10.1016/j.cie.2005.01.018.

Xie, J. *et al.* (2019) ‘Review on flexible job shop scheduling’, *IET Collaborative Intelligent Manufacturing*, 1(3), pp. 67–77. doi:10.1049/iet-cim.2018.0009.

Xu, J., Xu, X. and Xie, S.Q. (2011) ‘Recent developments in Dual Resource Constrained (DRC) system research’, *European Journal of Operational Research*, 215(2), pp. 309–318. doi:10.1016/j.ejor.2011.03.004.

Yao, N. and Zhang, F. (2020) *Optimal Real-time Scheduling of Human Attention for a Human and Multi-robot Collaboration System*, p. 35. doi:10.23919/ACC45564.2020.9147782.

Yow, A. *et al.* (2005) ‘Predicting nuclear power-plant operator performance using discrete event simulation’, *Cognition, Technology & Work*, 7(1), pp. 29–35. doi:10.1007/s10111-004-0167-x.

Zammori, F. and Bertolini, M. (2015) *A Conceptual Framework for Project Scheduling with Multi-Skilled Resources*. doi:10.2991/aiie-15.2015.103.

Zhang, J. *et al.* (2019) ‘Review of job shop scheduling research and its new perspectives under Industry 4.0’, *Journal of Intelligent Manufacturing*, 30(4), pp. 1809–1830. doi:10.1007/s10845-017-1350-2.

Zhang, S. and Wang, S. (2018) ‘Flexible Assembly Job-Shop Scheduling With Sequence-Dependent Setup Times and Part Sharing in a Dynamic Environment: Constraint Programming Model, Mixed-Integer Programming Model, and Dispatching Rules’, *IEEE Transactions on Engineering Management*, 65(3), pp. 487–504. doi:10.1109/TEM.2017.2785774.

Zheng, P. *et al.* (2018) ‘Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives’, *Frontiers of Mechanical Engineering*, 13(2), pp. 137–150. doi:10.1007/s11465-018-0499-5.

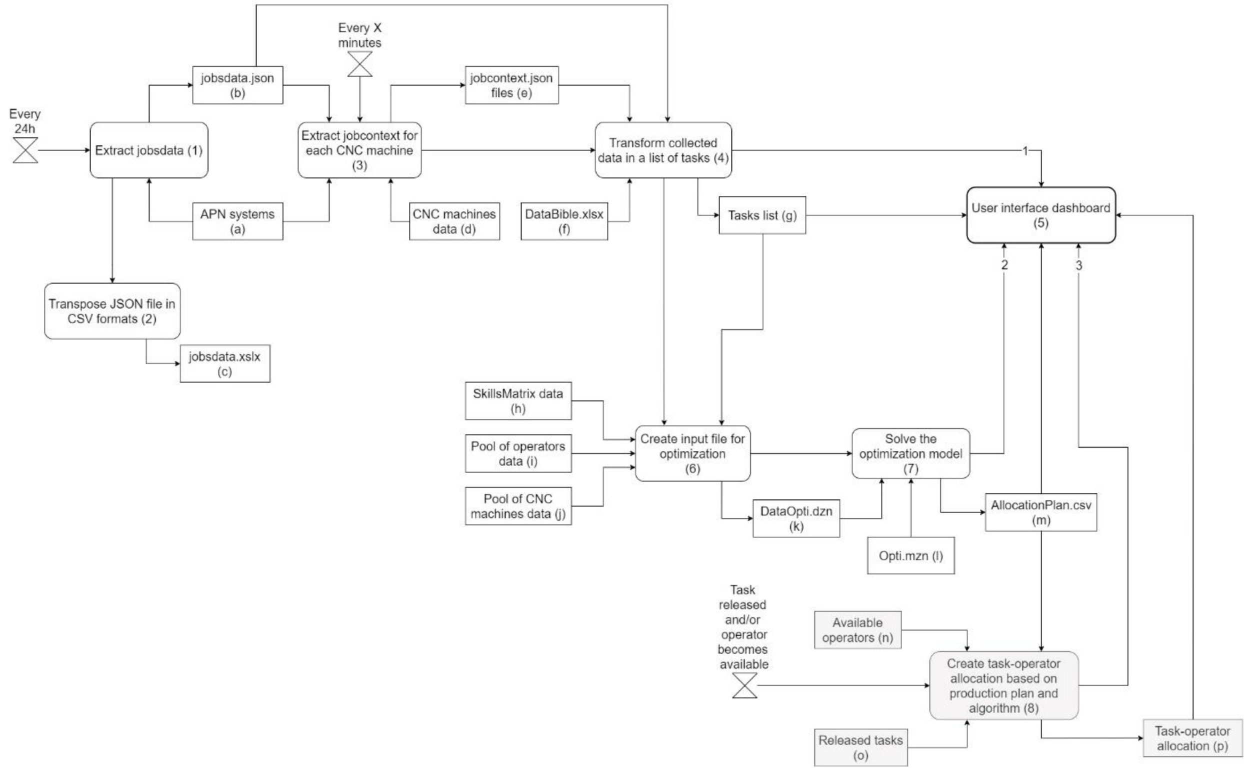
Zhu, H. *et al.* (2019) ‘An Adaptive Real-Time Scheduling Method for Flexible Job Shop Scheduling Problem With Combined Processing Constraint’, *IEEE Access*, 7, pp. 125113–125121. doi:10.1109/ACCESS.2019.2938548.

Zupan, H. and Herakovic, N. (2015) ‘Production line balancing with discrete event simulation: A case study’, *IFAC-PapersOnLine*, 48(3), pp. 2305–2311. doi:10.1016/j.ifacol.2015.06.431.

Chapter 5: Industrial project: realization and implementation

The bridge between scientific research and industrial applications may sometimes be difficult to cross. While providing new scientific results, our project also aims at proposing a functional implementation of the new system, that is, dynamic allocation of operators in a metal manufacturing job-shop.

In this chapter, we will detail the realization and implementation of our industrial project executed in a small or medium enterprise (SME) under Industry 4.0. In Figure 41, we present the workflow diagram describing the planned implementation of the system proposed in the form of an UML activity diagram. Every step required to implement this system will be detailed in this chapter, as well as the different data sources used.



Every 24h
Extract jobsdata (1)
jobsdata.json (b)
Every X minutes
Extract jobcontext for each CNC machine (3)
jobcontext.json files (e)
Transform collected data in a list of tasks (4)
1
User interface dashboard (5)
APN systems (a)
CNC machines data (d)
DataBible.xlsx (f)
Tasks list (g)
2
3
Transpose JSON file in CSV formats (2)
jobsdata.xslx (c)
SkillsMatrix data (h)
Pool of operators data (i)
Pool of CNC machines data (j)
Create input file for optimization (6)
Solve the optimization model (7)
DataOpti.dzn (k)
Opti.mzn (l)
AllocationPlan.csv (m)
Task released and/or operator becomes available
Available operators (n)
Create task-operator allocation based on production plan and algorithm (8)
Released tasks (o)
Task-operator allocation (p)

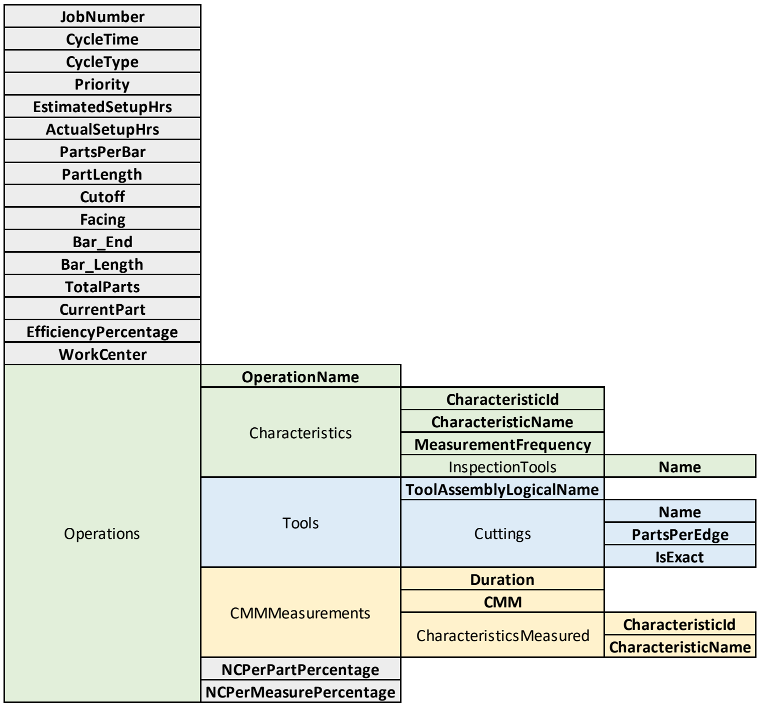
Figure 41: Workflow diagram describing the planned implementation of the system

5.1 Extract *jobsdata* (1) and transpose it in CSV format (2)

The first steps (1, 2) are to extract an embedded JSON file named *jobsdata* (b) from our industrial partner's databases (a) and transpose it in CSV formats (c). The systems contain data regarding the planification and scheduling of the 10 next jobs on each CNC machine ordered by priority number (1 to 10). For each prioritized job, data is extracted (1) coming from the computer-integrated manufacturing (CIM) database and the enterprise resource planning (ERP) database. These two databases contain exhaustive data regarding each job in the schedule.

The structure of the jobsdata JSON file (b) presented in Figure 42 then needs to be transposed (2). The different colors in Figure 2 present how the JSON file is transposed in four different CSV files compiled in a single *jobsdata.xslx* (c) under different tabs. The grey fields are placed in a *MasterJobData* tab, the green fields are placed in a *Characteristics* tab, the blue fields are placed in a *Tools* tab, and the yellow fields are placed in a *CMMMeasurements* tab. This step is useful in terms of validation of the input data, since the different CSV files produced (c) can easily be validated by a human, as compared to the extracted JSON *jobsdata* file (b).

These first two steps are executed every 24 hours. Since a single jobsdata JSON file contains data regarding the next 10 priorities in the factory, the timespan covers more than enough tasks to be extracted only once a day.

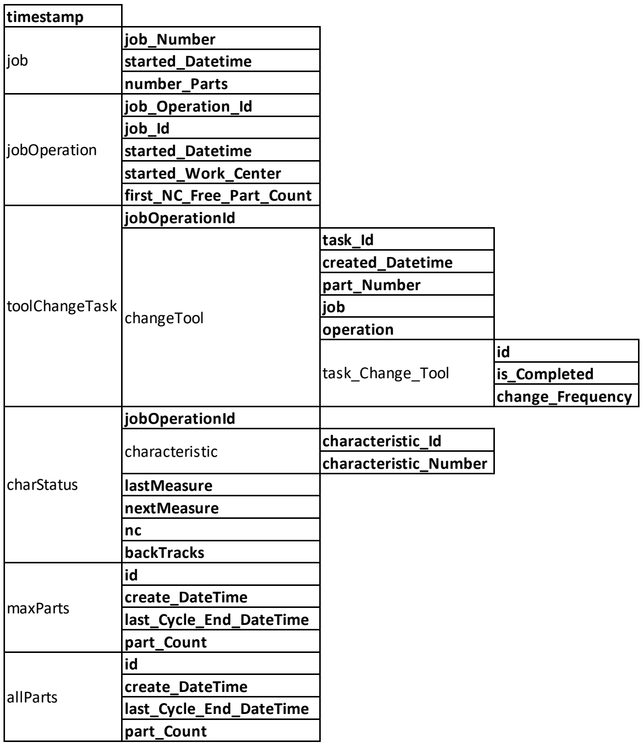


<table>
<tr><td colspan="4">JobNumber</td></tr>
<tr><td colspan="4">CycleTime</td></tr>
<tr><td colspan="4">CycleType</td></tr>
<tr><td colspan="4">Priority</td></tr>
<tr><td colspan="4">EstimatedSetupHrs</td></tr>
<tr><td colspan="4">ActualSetupHrs</td></tr>
<tr><td colspan="4">PartsPerBar</td></tr>
<tr><td colspan="4">PartLength</td></tr>
<tr><td colspan="4">Cutoff</td></tr>
<tr><td colspan="4">Facing</td></tr>
<tr><td colspan="4">Bar_End</td></tr>
<tr><td colspan="4">Bar_Length</td></tr>
<tr><td colspan="4">TotalParts</td></tr>
<tr><td colspan="4">CurrentPart</td></tr>
<tr><td colspan="4">EfficiencyPercentage</td></tr>
<tr><td colspan="4">WorkCenter</td></tr>
<tr><td rowspan="15">Operations</td><td colspan="3">OperationName</td></tr>
<tr><td rowspan="4">Characteristics</td><td colspan="2">CharacteristicId</td></tr>
<tr><td colspan="2">CharacteristicName</td></tr>
<tr><td colspan="2">MeasurementFrequency</td></tr>
<tr><td>InspectionTools</td><td>Name</td></tr>
<tr><td rowspan="4">Tools</td><td colspan="2">ToolAssemblyLogicalName</td></tr>
<tr><td rowspan="3">Cuttings</td><td>Name</td></tr>
<tr><td>PartsPerEdge</td></tr>
<tr><td>IsExact</td></tr>
<tr><td rowspan="4">CMMMeasurements</td><td colspan="2">Duration</td></tr>
<tr><td colspan="2">CMM</td></tr>
<tr><td rowspan="2">CharacteristicsMeasured</td><td>CharacteristicId</td></tr>
<tr><td>CharacteristicName</td></tr>
<tr><td colspan="3">NCPerPartPercentage</td></tr>
<tr><td colspan="3">NCPerMeasurePercentage</td></tr>
</table>

Figure 42: Data structure of the extracted jobsdata JSON file (b). Each color represents a different CSV file placed in a different tab in the jobsdata Excel file (c).

5.2 Extract *jobcontext* for each CNC machine (3)

More data is required in addition to the contents of the *jobsdata* file (b) since it concerns medium/long term events. The horizon covered by the data in this file is a few weeks. We also need extremely precise data with a much smaller horizon, regarding the real-time events in the factory. To this end, data coming directly from the *CNC machines* (d) is extracted and is aggregated to data from the *APN systems* (a). A new file named *jobcontext* (e) for each CNC machine is built. The extracted file contains more fields than the data structure presented in Figure 43, only the pertinent ones are shown.



timestamp			
job	job_Number		
	started_Datetime		
	number_Parts		
jobOperation	job_Operation_Id		
	job_Id		
	started_Datetime		
	started_Work_Center		
	first_NC_Free_Part_Count		
toolChangeTask	jobOperationId		
	changeTool	task_Id	
		created_Datetime	
		part_Number	
		job	
		operation	
		task_Change_Tool	id
			is_Completed
			change_Frequency
charStatus	jobOperationId		
	characteristic	characteristic_Id	
		characteristic_Number	
	lastMeasure		
	nextMeasure		
	nc		
	backTracks		
maxParts	id		
	create_DateTime		
	last_Cycle_End_DateTime		
	part_Count		
allParts	id		
	create_DateTime		
	last_Cycle_End_DateTime		
	part_Count		

Figure 43 : Data structure of the extracted jobcontext files (e) in which data coming from APN systems (a) and CNC machines (d) is aggregated.

The timestamp field is used to log the time for each extraction. This extraction corresponds to what is going on in the factory at the time on the *timestamp*. The *job* and *jobOperation* fields are used in order to find the correct tasks in the transformed *jobsdata* CSV. Normally, the *number_Parts* field should correspond to the *TotalParts* field of the *jobsdata*. Every time a value is different in both extractions, the *jobcontext* was considered more accurate.

The field *toolChangeTask* contains useful updated information regarding the tool change tasks, initially determined from the *jobsdata* file.

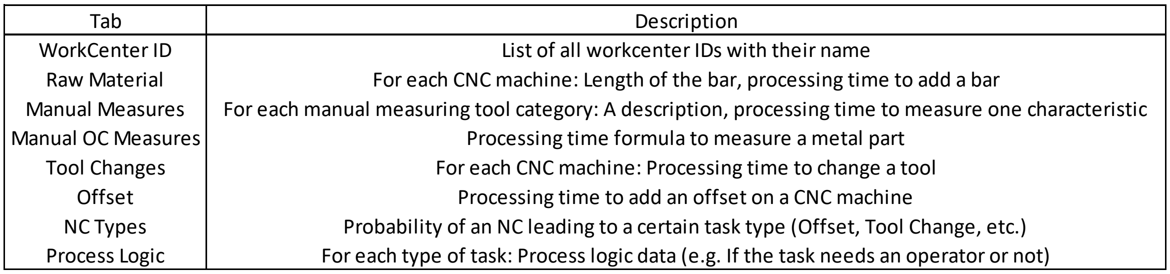
In the same way, the *charStatus* field is useful to update the parts measuring tasks. It also provided crucial information regarding the non-conformity tasks and the backtracked parts measuring tasks, since this data is exclusively real-time data and cannot be planned out in advance with the jobsdata file.

Finally, the last two fields, *maxParts* and *allParts* contain similar information. The *allParts* field contains a list of all the machined parts on this given job and acts as historical data in order to estimate the cycle time extracted from the *jobsdata* more accurately. With a more accurate cycle time, the next tasks can be planned out more precisely. Similarly, *maxParts* only contains the data for the last machined part. It gives important information as to the progress of the ongoing job (i.e., which part number was last machined and when it was completed).

5.3 Transform collected data in a list of tasks (4)

Once all the data is collected, we are ready to apply a transformation in order to obtain a list of tasks ready to allocate to operators. In this step, we then use the *jobsdata* file (b), the last extracted *jobcontext* files for each CNC machine (e), as well as an additional file named *DataBible.xslx* (f). This file was previously created to contain all the important information that is not a part of the extracted files. The data compiled in the *DataBible* file (f) was located in eight tabs, listed, and described in Table 18. This data comes from observations in the factory, interviews with experts in the company as well as previous project conducted by the industrial partner. This step leads to the creation of a tasks list (g) that contains ordered data on upcoming tasks.

Table 18: Tabs content in the Data Bible (f) file.

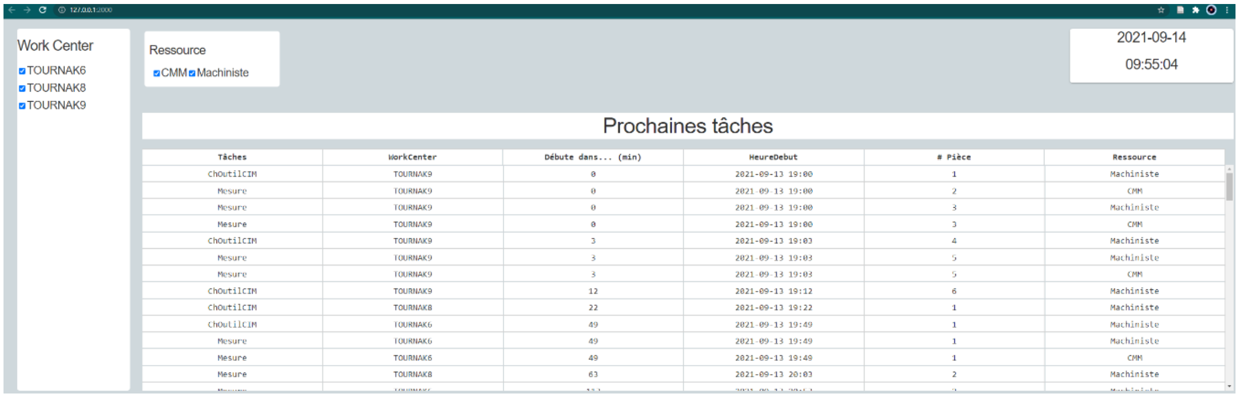


Tab	Description
WorkCenter ID	List of all workcenter IDs with their name
Raw Material	For each CNC machine: Length of the bar, processing time to add a bar
Manual Measures	For each manual measuring tool category: A description, processing time to measure one characteristic
Manual OC Measures	Processing time formula to measure a metal part
Tool Changes	For each CNC machine: Processing time to change a tool
Offset	Processing time to add an offset on a CNC machine
NC Types	Probability of an NC leading to a certain task type (Offset, Tool Change, etc.)
Process Logic	For each type of task: Process logic data (e.g. If the task needs an operator or not)

5.4 User interface dashboard (5)

We then create a user interface dashboard named the *Tasks Dashboard* following arrow 1 in Figure 41. The purpose of this dashboard is to display the upcoming tasks to the operators on a screen in the factory. In order to keep this dashboard up to date, a refresh frequency is determined and every *X* minutes we extract the *jobcontext* files again for every CNC machine.

We now have a fully functioning dashboard that is currently being implemented on the production floor. In Figure 44, we present a screenshot of the *Tasks Dashboard* as it is following step 4. This dashboard's purpose is to show the upcoming tasks and their planned release time in minutes. Each line in the dashboard represents a task. The columns (from left to right) display the type of task (e.g. a tool change), the workcenter on which the task occurs, the number of remaining minutes until the task is ready to be executed, the timestamp at which the task will be ready to be executed, the number of the part related to the task, and the resource type required to tackle the task. The user can choose which resource type must be displayed (some tasks require the CMM, others a human operator) and which workcenters are included in the lot.



Work Center

☑TOURNAK6
☑TOURNAK8
☑TOURNAK9

Ressource

☑CMM ☑Machiniste

2021-09-14
09:55:04

Prochaines tâches

Tâches	WorkCenter	Débute dans... (min)	HeureDebut	# Pièce	Ressource
ChOutilCIM	TOURNAK9	0	2021-09-13 19:00	1	Machiniste
Mesure	TOURNAK9	0	2021-09-13 19:00	2	CMM
Mesure	TOURNAK9	0	2021-09-13 19:00	3	Machiniste
Mesure	TOURNAK9	0	2021-09-13 19:00	3	CMM
ChOutilCIM	TOURNAK9	3	2021-09-13 19:03	4	Machiniste
Mesure	TOURNAK9	3	2021-09-13 19:03	5	Machiniste
Mesure	TOURNAK9	3	2021-09-13 19:03	5	CMM
ChOutilCIM	TOURNAK9	12	2021-09-13 19:12	6	Machiniste
ChOutilCIM	TOURNAK8	22	2021-09-13 19:22	1	Machiniste
ChOutilCIM	TOURNAK6	49	2021-09-13 19:49	1	Machiniste
Mesure	TOURNAK6	49	2021-09-13 19:49	1	Machiniste
Mesure	TOURNAK6	49	2021-09-13 19:49	1	CMM
Mesure	TOURNAK8	63	2021-09-13 20:03	2	Machiniste

Figure 44: Tasks Dashboard illustration showing the upcoming tasks to the user.

While the *Tasks Dashboard* is useful for the user to see which tasks are coming in the next minutes, it does not provide an allocation plan. The user must decide by him or herself which tasks he or she will do first when multiple tasks occur at the same time. We can continue to add functions to this dashboard to aim at displaying to the user an allocation plan based on the optimization model presented in Chapter 4 of this thesis.

5.5 Create input file for optimization (6)

The next step creating an input file to use in the optimization model. In order to produce this file, additional information is required. First, a file named *SkillsMatrix* (h) which provides in detail the different skills that each operator possess is required. We require the skills that each operator has on each CNC machine, for each type of tasks, and if the operator is certified. A certified operator means that they can tackle tasks on jobs for a particular client that requires special training. For this project, we manually produced this file based on human resource's documentations and meetings with resources at our industrial partner's.

Additionally, we require information on the pool of operators (i) that must be considered in the production plan as well as the pool of CNC machines (j). This can be provided in many different manners to the system. For example, when realizing experiments with the optimization model, a complete operator schedule of the upcoming week was used as well as the schedule for the CNC machines. When the system will be used in real-time, operators could manually "punch in" the system if they want to be considered in the pool of operators. Similarly, production managers could choose the pool of CNC machines which tasks need to be allocated.

This step produces an input file for the optimization model. The planning horizon is currently set to one hour, so all tasks included in the input file are planned to be released in the next hour. In this project, the file is called *DataOpti.dzn* (k) since this is the format required for our optimization model that is built in MiniZinc programming language.

5.6 Solve the optimization model (7)

Once we have a data file in the right format, we can solve the optimization model. We use the model called *Opti.mzn* (k) which has been tested in experiments presented in Chapter 4 and produces a plan in *AllocationPlan.csv* (m). All details on the optimization model can be found in Chapter 4.

5.7 Display the solution on the user interface dashboard (5)

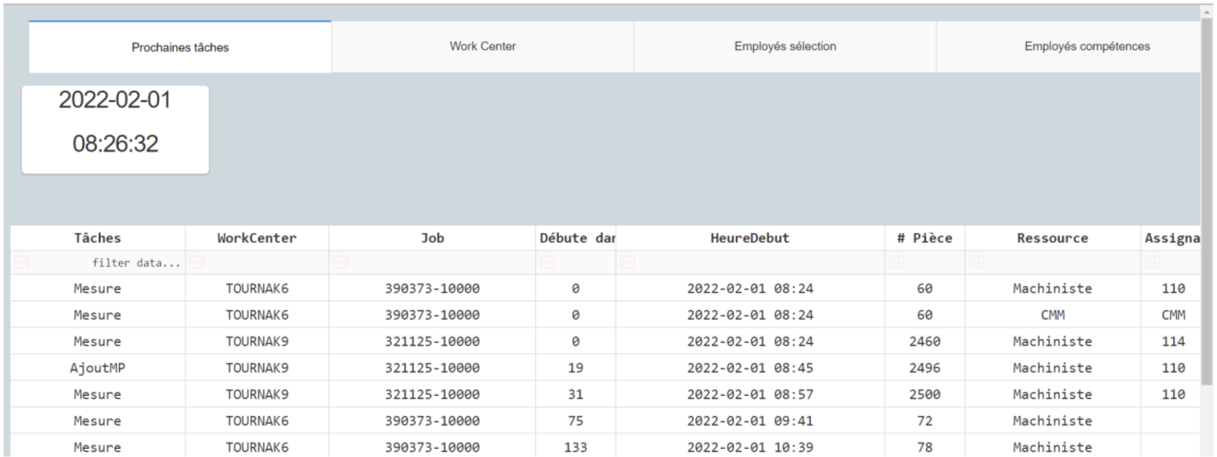
The tasks allocation plan created can then be displayed on the *Tasks Dashboard* (5), following arrow 2 in Figure 41. Using the solution of the optimization model in the *AllocationPlan.csv* file, we produce an improved dashboard which shows the allocation plan proposed. In Figure 45, we see an illustration of the new user interface dashboard (5).

Each line in the dashboard represents a task. On the main tab, the columns (left to right) show the type of task, the workcenter on which the task occurs, the job from which the task occurs, the remaining minutes before the task is ready to be executed, the timestamp of the release of the task, the part number related to the task, the resource required to tackle the task and finally the employee number of the operator to whom the task is allocated in the plan.

This dashboard also has three additional tabs. The second tab provide the same function than in the initial Tasks Dashboard, which is for the user to select which workcenter must be included in the plan. In the third tab, the user may now also select which operators (by checking their employee number) must be included in the allocation plan. Finally, the last tab can be used by the user to quickly modify the *SkillsMatrix* (g) if it is outdated.

Additionally, tasks which release date is further than the chosen planning horizon (one hour) are not allocated in the current plan (as shown in the last two lines in Figure 5). Once they are included in the planning horizon, an operator's employee number will appear in the last column on the right which represents the allocation.

While this plan may not be feasible (see future developments section), this dashboard is a new way to start implementing the project on the factory floor and further validate the data as well as the plan proposed by the optimization module.



2022-02-01

08:26:32

Tâches	WorkCenter	Job	Débute dar	HeureDebut	# Pièce	Ressource	Assigna
filter data...							
Mesure	TOURNAK6	390373-10000	0	2022-02-01 08:24	60	Machiniste	110
Mesure	TOURNAK6	390373-10000	0	2022-02-01 08:24	60	CMM	CMM
Mesure	TOURNAK9	321125-10000	0	2022-02-01 08:24	2460	Machiniste	114
AjoutMP	TOURNAK9	321125-10000	19	2022-02-01 08:45	2496	Machiniste	110
Mesure	TOURNAK9	321125-10000	31	2022-02-01 08:57	2500	Machiniste	110
Mesure	TOURNAK6	390373-10000	75	2022-02-01 09:41	72	Machiniste	
Mesure	TOURNAK6	390373-10000	133	2022-02-01 10:39	78	Machiniste	

Figure 45: Tasks Allocation Plan dashboard 's illustration showing the allocation plan for the next hour of production.

5.8 Create task-operator allocation based on production plan and algorithm (8)

Following the creation of a task allocation plan, we need to verify if the plan is feasible each time a task is released, or an operator becomes available in order to create final allocations and deliver the *Tasks System*. Because of the nature of the problem solved by the optimization model, the generated plan may quickly become unfeasible. When a task's start time is reached, if the allocated operator from the plan is unavailable due to some unforeseen event, the task should be allocated to another free operator immediately.

We first need to determine which operators are currently available (n). While we know the pool of operators from the previous phase, some of these operators may be unavailable for different reasons. For example, an operator may already be processing a task, but he can also be currently on a small break to go to the restroom or talking with a supervisor. Different methods may be used to determine which operators are available, from a manual input to a tracker based on the internet of things.

In a similar way, we need to determine at the same time which tasks are currently released but have not been allocated to and started by an operator (o). While the previous steps in the system, such as the extraction of the *jobcontext* files (e) provide useful real-time data that gets more and more precise as the planned release time of a task approaches, we need to be absolutely certain that a certain task is released before allocating it to an operator. There needs to be a certain module that confirms which tasks are now released and ready to be started by an operator.

Using the information on which operator is available as well as which tasks is released, we are finally ready to create pairs of released task with available operator. In order to produce these pairs, when a task is confirmed to be released, first we will look at the production plan produced in the previous phase (*AllocationPlan.csv*). Real-time allocation model will then be used to either seal the pairs proposed by the plan, if feasible. If not, the model will propose an alternative solution to allocate the released task (n) to an available operator (n). This operator-task allocation (p) pair will also be used to confirm that this task is now started and does not require to be allocated anymore (back in step 9).

Finally, once a pair has been created, the available operator will be notified of its new current task with an updated dashboard (5). Since operators are mobile in the factory, we need a way to tell the operator that a new task has been allocated to them and is ready to be processed. Many methods can be used to accomplish this, from individual devices carried by operators to a big screen displayed in the factory.

5.9 Future developments

This project's implementation is currently ongoing in the factory. The *Tasks Dashboard* produced following step (4) is being deployed with a super-user in the factory and a feedback template has been provided to help validate and improve the accuracy of the tasks predicted. The *Tasks Allocation Plan* following step (7) is currently being developed and validated with the research and development team from APN. The last step (8) of the project has not yet officially begun.

Additionally, two future developments are proposed by our team. First, we propose analyzing the rescheduling strategy. While currently, the system loops at step 3 with a chosen frequency (every *X* minutes), Ghaleb, Zolfagharinia and Taghipour (2020) suggest that triggering the rescheduling upon certain pre-selected events might prove to be optimal. We suggest examining this proposition, and if re-optimizing the schedule upon event is chosen, an event observation module would need to be integrated (12), as shown in the greyed out step in phase 1. Second, as seen in box (13), we highly suggest that a system be implemented in order to keep the skills of the operators updated since the entire system is based upon them. A tab in the *Tasks Allocation Plan* dashboard has been put in place by our team to do last minute changes to the skills matrix, but a systematic management of the workforce skills would be optimal.

Conclusion

The integration and interconnexion of systems make it possible to rethink different paradigms under Industry 4.0. In the manufacturing industry, real-time data is becoming more available, which yields an opportunity to change the current methods of allocating resources. The paradigm shift studied in this master thesis is the change from a static to a dynamic allocation of human resources to machines in the metal manufacturing industry. The goal was to determine and quantify the benefits associated to human resources allocation in an Industry 4.0 context, in collaboration with an industrial partner willing to implement this new paradigm. The main methods used to reach this objective were discrete-event simulation and optimization.

As part of this study, three scenarios were defined and tested throughout the project. The first scenario represented the current paradigm, that is, the static allocation of resources. Both other scenarios consisted of dynamic allocation, one considering the actual skills of the workforce while the other supposed perfectly versatile operators in the factory.

In Chapter 3, a discrete-event simulation model was built based on the case study of our industrial partner's factory, using Simio. Once validated, the simulation model allowed us to perform two different experiments in order to compare the different scenarios on the basis of different criteria such as the time required to produce a certain number of parts in the factory, the total distance walked by the operators during production, the total number of parts produced during a week of production and the productive utilization rate of the machines during the production (i.e., the time spent machining parts). The experiments showed that the most important productivity gain was moving from the static allocation scenario to the dynamic allocation scenario with the current skills. While the assumption that the operators are completely versatile suggests further improvements, this would also require important training costs.

While the simulation experiments considered a simple allocation method of "closest free and available resource", we believe implementing a dynamic task allocation system in a factory would require a more sophisticated method, making use of the available real-time information from the systems. A real-time optimization model was built and presented as part of the article in Chapter 4. In a preliminary exploration, both a mixed-integer programming (MIP) model and a constraint programming (CP) model were built. Experiments comparing the three scenarios with industrial data on these models showed that the CP model outperformed the MIP model in all instances, so the project was pursued using the CP model. Using real-time data, an exhaustive experiment was conducted using the data extracted every hour for an entire week of production. These results agreed with the ones from the simulation experiments. Both dynamic allocation scenarios proposed production plans for the next hour of production that yielded extremely low amounts of tardiness when compared to plans produced with the static allocation. The performance of the CP optimization model lets us believe that it could be used in a real-time setting, to allocate the operators to the tasks on the production floor since it can be solved quite rapidly (few seconds at most). Finally, a second experiment with the optimization model suggested that it would be possible to reduce the number of operators in the factory and still manage to yield low levels of tardiness under a dynamic scenario with completely versatile operators.

Given the interesting results obtained, the industrial partner is currently moving forward with the implementation of this project that they have named the *tasks system*. In Chapter 5, we proposed an implementation system for the project in an industrial context. The chapter provided industrial insights into the realization and the implementation of an Industry 4.0 project for a manufacturing company.

While this new paradigm has been shown to have many interesting productivity benefits in a metal manufacturing job-shop, we must state a few precautions that must be taken. First, experiments with the plans yielded with the optimization model tend to allocate more tasks to highly-skilled operators while keeping the operators with few skills idle for extended periods of time. Obviously, this strategy is not recommended since operators need to develop skills and keeping them unoccupied does not achieve this goal. Additionally, overworking highly-skilled operators may lead to frustration linked to the unfairness of the workload distribution as well as possible burnout. The project will continue with the intent of maximizing the productivity gains it can bring to an organization while minimizing its possible inconveniences. Various future developments can be explored in this sense, starting by using allocation rules which favor a fair allocation plan between operators. Additionally, operator-training decisions could be guided by to the possible productivity gains identified by the simulation model or the optimization model. The optimization model could also be connected to the simulation model in order to analyze additional aspects of the dynamic allocation strategy. Finally, other aspects of the dynamic allocation paradigm proposed could be studied, for example, the effect on part quality. Since an operator is not responsible for the execution of a job from beginning to end, the constant change of context could make the machining process harder to control.

In this project, we analyzed and proved the relevance of embracing a new paradigm in the manufacturing sector under Industry 4.0. This new paradigm consists of dynamically allocating human resources to the different tasks requiring human intervention during the production. Dynamic allocation was proven to help improve different performance indicator such as the total number of parts produced in a week of production in a metal manufacturing job-shop. Furthermore, using real-time data, we proposed an allocation algorithm in the form of a constraint programming optimization model that yielded a good performance and managed to allocate the tasks in the next production hour in real time. Finally, we proposed an implementation framework for this type of project in a 4.0 context.

References

Afshar-Nadjafi, B. (2021) ‘Multi-skilling in scheduling problems: A review on models, methods and applications’, *Computers & Industrial Engineering*, 151, p. 107004. doi:10.1016/j.cie.2020.107004.

Agnetis, A., Murgia, G. and Sbrilli, S. (2014) ‘A job shop scheduling problem with human operators in handicraft production’, *International Journal of Production Research*, 52(13), pp. 3820–3831. doi:10.1080/00207543.2013.831220.

Alade, O.M. and Amusat, A.O. (2019) ‘Solving Nurse Scheduling Problem Using Constraint Programming Technique’, *arXiv:1902.01193 [cs]* [Preprint]. Available at: http://arxiv.org/abs/1902.01193 (Accessed: 4 August 2021).

Al-Behadili, M., Ouelhadj, D. and Jones, D. (2019) ‘Multi-objective Biased Randomised Iterated Greedy for Robust Permutation Flow Shop Scheduling Problem under Disturbances’, *Journal of the Operational Research Society* [Preprint]. doi:10.1080/01605682.2019.1630330.

Allgeier, H. *et al.* (2020) ‘Simulation-Based Evaluation of Lot Release Policies in a Power Semiconductor Facility - a Case Study’, in *2020 Winter Simulation Conference (WSC), 14-18 Dec. 2020*. Piscataway, NJ, USA: IEEE (Proceedings of the 2020 Winter Simulation Conference (WSC)), pp. 1503–14. doi:10.1109/WSC48552.2020.9384094.

Apt, K. (2003) *Principles of Constraint Programming*. Cambridge University Press.

Artigues, C. *et al.* (2009) ‘Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound’, *Computers and Operations Research*, 36(8), pp. 2330–2340. doi:10.1016/j.cor.2008.08.013.

Banks, J. *et al.* (2005) *Discrete-Event System Simulation*. Prentice-Hall. Available at: http://www.worldcat.org/oclc/55847249 (Accessed: 26 January 2022).

Bär, K., Herbert-Hansen, Z.N.L. and Khalid, W. (2018) ‘Considering Industry 4.0 aspects in the supply chain for an SME’, *Production Engineering*, 12(6), pp. 747–758. doi:10.1007/s11740-018-0851-y.

Barlas, P. and Heavey, C. (2016) ‘Automation of input data to discrete event simulation for manufacturing: A review’, *International Journal of Modeling, Simulation, and Scientific Computing*, 07, p. 1630001. doi:10.1142/S1793962316300016.

Beauchemin, M. *et al.* (2020) ‘Evaluating workers allocation policies through the simulation of a high-precision machining workshop’, in *Winter Simulation Conference*.

Bhulai, S., Koole, G. and Pot, A. (2008) ‘Simple Methods for Shift Scheduling in Multiskill Call Centers’, *Manufacturing & Service Operations Management*, 10(3), pp. 411–420. doi:10.1287/msom.1070.0172.

Borreguero-Sanchidrián, T. *et al.* (2018) ‘Flexible Job Shop Scheduling With Operators in Aeronautical Manufacturing: A Case Study’, *IEEE Access*, 6, pp. 224–233. doi:10.1109/ACCESS.2017.2761994.

Borshchev, A. (2014) ‘Multi-method modelling: AnyLogic’, in, pp. 248–279. doi:10.1002/9781118762745.ch12.

Boschert, S., Heinrich, C. and Rosen, R. (2018) *Next Generation Digital Twin*.

Boschert, S. and Rosen, R. (2016) ‘Digital Twin—The Simulation Aspect’, in Hehenberger, P. and Bradley, D. (eds) *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers*. Cham: Springer International Publishing, pp. 59–74. doi:10.1007/978-3-319-32156-1_5.

Bouajaja, S. and Dridi, N. (2017) ‘A survey on human resource allocation problem and its applications’, *Operational Research*, 17(2), pp. 339–369. doi:10.1007/s12351-016-0247-8.

Bourdais, S., Galinier, P. and Pesant, G. (2003) ‘hibiscus: A Constraint Programming Application to Staff Scheduling in Health Care’, in Rossi, F. (ed.) *Principles and Practice of Constraint Programming – CP 2003*. Berlin, Heidelberg: Springer (Lecture Notes in Computer Science), pp. 153–167. doi:10.1007/978-3-540-45193-8_11.

Brik, B. *et al.* (2019) *Accuracy and Localization-Aware Rescheduling for Flexible Flow Shops in Industry 4.0*. doi:10.1109/CoDIT.2019.8820445.

Bülbül, K. and Kaminsky, P. (2013) ‘A linear programming-based method for job shop scheduling’, *Journal of Scheduling*, 16(2), pp. 161–183. doi:10.1007/s10951-012-0270-4.

Butt, J. (2020) ‘A Strategic Roadmap for the Manufacturing Industry to Implement Industry 4.0’, *Designs*, 4(2), p. 11. doi:10.3390/designs4020011.

Caprara, A., Monaci, M. and Toth, P. (2003) ‘Models and algorithms for a staff scheduling problem’, *Mathematical Programming*, 98(1), pp. 445–476. doi:10.1007/s10107-003-0413-7.

Carlier, J. (1982) ‘The one-machine sequencing problem’, *European Journal of Operational Research*, 11(1), pp. 42–47. doi:10.1016/S0377-2217(82)80007-6.

Costa, A., Cappadonna, F.A. and Fichera, S. (2014) ‘Joint optimization of a flow-shop group scheduling with sequence dependent set-up times and skilled workforce assignment’, *International Journal of Production Research*, 52(9), pp. 2696–2728. doi:10.1080/00207543.2014.883469.

Costigliola, A. *et al.* (2017) ‘Simulation Model of a Quality Control Laboratory in Pharmaceutical Industry’, *IFAC-PapersOnLine*, 50(1), pp. 9014–9019. doi:10.1016/j.ifacol.2017.08.1582.

Cunha, M.M. *et al.* (2019) ‘Dual Resource Constrained Scheduling for Quality Control Laboratories’, *IFAC-PapersOnLine*, 52(13), pp. 1421–1426. doi:10.1016/j.ifacol.2019.11.398.

Dantzig, G.B. and Wolfe, P. (1960) ‘Decomposition Principle for Linear Programs’, *Operations Research*, 8(1), pp. 101–111. doi:10.1287/opre.8.1.101.

Deming, W., Neumann, J. and Morgenstern, O. (1944) ‘Theory of Games and Economic Behavior’. doi:10.2307/2280142.

Demir, Y. and Kürşat İşleyen, S. (2013) ‘Evaluation of mathematical models for flexible job-shop scheduling problems’, *Applied Mathematical Modelling*, 37(3), pp. 977–988. doi:10.1016/j.apm.2012.03.020.

Detty, R.B. and Yingling, J.C. (2000) ‘Quantifying benefits of conversion to lean manufacturing with discrete event simulation: A case study’, *International Journal of Production Research*, 38(2), pp. 429–445. doi:10.1080/002075400189509.

Devezas, T., Leitão, J. and Sarygulov, A. (eds) (2017) *Industry 4.0: Entrepreneurship and Structural Change in the New Digital Landscape*. Cham: Springer International Publishing (Studies on Entrepreneurship, Structural Change and Industrial Dynamics). doi:10.1007/978-3-319-49604-7.

Edi, K.H. and Duquenne, P. (2009) ‘Intégration de la polyvalence et de la modulation d’horaire dans une approche d’affectation flexible de la ressource humaine’, *revue ivoirienne des sciences et technologies*, pp. 1–20.

Eiselt, H.A. and Marianov, V. (2008) ‘Employee positioning and workload allocation’, *Computers & Operations Research*, 35, pp. 513–524. doi:10.1016/j.cor.2006.03.014.

Endsley, M. (1995) ‘Toward a Theory of Situation Awareness in Dynamic Systems’, *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37, pp. 32–64. doi:10.1518/001872095779049543.

Eriksson, S. (2020) ‘Optimal Multi-Skilled Workforce Scheduling for Contact Centers Using Mixed Integer Linear Programming’. KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF ENGINEERING SCIENCES.

Erming Zhou, Jin Zhu, and Ling Deng (2017) ‘Flexible job-shop scheduling based on genetic algorithm and simulation validation’, in *13th Global Congress on Manufacturing and Management (GCMM 2016), 28-30 Nov. 2016. MATEC Web of Conferences*, France: EDP Sciences (MATEC Web Conf. (France)), p. 02047 (8 pp.). doi:10.1051/matecconf/201710002047.

Evangelou, G. *et al.* (2021) ‘An approach for task and action planning in Human–Robot Collaborative cells using AI’, *Procedia CIRP*, 97, pp. 476–481. doi:10.1016/j.procir.2020.08.006.

Fattahi, P., Jolai, F. and Arkat, J. (2009) ‘Flexible job shop scheduling with overlapping in operations’, *Applied Mathematical Modelling*, 33(7), pp. 3076–3087. doi:10.1016/j.apm.2008.10.029.

Fattahi, P., Saidi Mehrabad, M. and Jolai, F. (2007) ‘Mathematical modeling and heuristic approaches to flexible job shop scheduling problems’, *Journal of Intelligent Manufacturing*, 18(3), pp. 331–342. doi:10.1007/s10845-007-0026-8.

Ferjani, A. *et al.* (2017) ‘A simulation-optimization based heuristic for the online assignment of multi-skilled workers subjected to fatigue in manufacturing systems’, *Computers & Industrial Engineering*, 112, pp. 663–674. doi:10.1016/j.cie.2017.02.008.

Fourer, R. (no date) ‘AMPL: A Mathematical Programming Language’, p. 65.

Framinan, J., Fernandez-Viagas, V. and Perez-Gonzalez, P. (2019) ‘Using real-time information to reschedule jobs in a flowshop with variable processing times’, *Computers & Industrial Engineering*, 129. doi:10.1016/j.cie.2019.01.036.

Garey, M.R., Johnson, D.S. and Sethi, R. (1976) ‘The Complexity of Flowshop and Jobshop Scheduling’, *Mathematics of Operations Research*, 1(2), pp. 117–129.

Garrido, J. and Sáez, J. (2019) ‘Integration of automatic generated simulation models, machine control projects and management tools to support whole life cycle of industrial digital twins.’, *IFAC-PapersOnLine*, 52(13), pp. 1814–1819. doi:10.1016/j.ifacol.2019.11.465.

Ghaleb, M., Taghipour, S. and Zolfagharinia, H. (2020) ‘Real-Time Optimization of Maintenance and Production Scheduling for an Industry 4.0-Based Manufacturing System’, in *2020 Annual Reliability and Maintainability Symposium (RAMS). 2020 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–8. doi:10.1109/RAMS48030.2020.9153721.

Ghaleb, M., Zolfagharinia, H. and Taghipour, S. (2020) ‘Real-time production scheduling in the Industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns’, *Computers & Operations Research*, 123, p. 105031. doi:10.1016/j.cor.2020.105031.

Ghobakhloo, M. (2018) ‘The future of manufacturing industry: a strategic roadmap toward Industry 4.0’, *Journal of Manufacturing Technology Management*, 29(6), pp. 910–936. doi:10.1108/JMTM-02-2018-0057.

Gleixner, A. (2018) ‘Computational Mixed-Integer Programming’.

Goldsman, D., Nance, R. and Wilson, J. (2010) *A brief history of simulation*, *Proceedings - Winter Simulation Conference*, p. 313. doi:10.1109/WSC.2009.5429341.

Greis, N.P. *et al.* (2019) ‘Manufacturing-Uber: Intelligent Operator Assignment in a Connected Factory’, *IFAC-PapersOnLine*, 52(13), pp. 2734–2739. doi:10.1016/j.ifacol.2019.11.621.

Grieco, A. *et al.* (2017) ‘An Industry 4.0 Case Study in Fashion Manufacturing’, *Procedia Manufacturing*, 11, pp. 871–877. doi:10.1016/j.promfg.2017.07.190.

Gruzauskas, V. (2016) ‘Labour and machine efficient utilization importance to the enterprise profit’, *Journal of Management*, 28, pp. 127–125.

Gunal, M.M. (ed.) (2019) *Simulation for Industry 4.0: Past, Present, and Future*. Cham: Springer International Publishing (Springer Series in Advanced Manufacturing). doi:10.1007/978-3-030-04137-3.

Ham, A. and Cakici, E. (2016) ‘Flexible job shop scheduling problem with parallel batch processing machines: MIP and CP approaches’, *Computers & Industrial Engineering*, 102. doi:10.1016/j.cie.2016.11.001.

Ham, A., Park, M.-J. and Kim, K.M. (2021) ‘Energy-Aware Flexible Job Shop Scheduling Using Mixed Integer Programming and Constraint Programming’, *Mathematical Problems in Engineering*, 2021, p. e8035806. doi:10.1155/2021/8035806.

Hammann, J.E. and Markovitch, N.A. (1995) ‘Introduction to Arena [simulation software]’, in *Winter Simulation Conference Proceedings, 1995. Winter Simulation Conference Proceedings, 1995.*, pp. 519–523. doi:10.1109/WSC.1995.478785.

Hamrol, A. *et al.* (2018) ‘Analysis of the Conditions for Effective Use of Numerically Controlled Machine Tools’, in Hamrol, A. et al. (eds) *Advances in Manufacturing*. Cham: Springer International Publishing (Lecture Notes in Mechanical Engineering), pp. 3–12. doi:10.1007/978-3-319-68619-6_1.

Haralick, R.M. and Elliott, G.L. (1980) ‘Increasing tree search efficiency for constraint satisfaction problems’, *Artificial Intelligence*, 14(3), pp. 263–313. doi:10.1016/0004-3702(80)90051-X.

Harmonosky, C.M. and Robohn, S.F. (1991) ‘Real-time scheduling in computer integrated manufacturing: a review of recent research’, *International Journal of Computer Integrated Manufacturing*, 4(6), pp. 331–340. doi:10.1080/09511929108944511.

Henning, K. (2013) *Recommendations for implementing the strategic initiative INDUSTRIE 4.0*. Available at: https://www.semanticscholar.org/paper/Recommendations-for-implementing-the-strategic-4.0-Henning/80d4b3eccd6951cf6dba61f1b33889a0edbf0407 (Accessed: 1 February 2022).

Ho, T.-W. *et al.* (2018) ‘A Platform for Dynamic Optimal Nurse Scheduling Based on Integer Linear Programming along with Multiple Criteria Constraints’, in *Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference on ZZZ - AICCC ’18. the 2018 Artificial Intelligence and Cloud Computing Conference*, Tokyo, Japan: ACM Press, pp. 145–150. doi:10.1145/3299819.3299825.

van Hoeve, W.-J. and Katriel, I. (2006) ‘Global constraints’, in.

Hollocks, B. (1992) ‘A well-kept secret?’, *OR Insight*, 5(4), pp. 12–17. doi:10.1057/ori.1992.29.

Hughes, K. and Jiang, X. (2010) ‘Using discrete event simulation to model excavator operator performance’, *Human Factors and Ergonomics in Manufacturing & Service Industries*, 20(5), pp. 408–423. doi:10.1002/hfm.20191.

Huynh, B.H., Akhtar, H. and Li, W. (2020) ‘Discrete Event Simulation for Manufacturing Performance Management and Optimization: A Case Study for Model Factory’, in *2020 9th International Conference on Industrial Technology and Management (ICITM)*. *2020 9th International Conference on Industrial Technology and Management (ICITM)*, pp. 16–20. doi:10.1109/ICITM48982.2020.9080394.

Idrisov, G.I. *et al.* (2018) ‘New technological revolution: Challenges and opportunities for Russia’, *Voprosy Ekonomiki*, pp. 5–25. doi:10.32609/0042-8736-2018-4-5-25.

Ingemansson, A., Bolmsjö, G.S. and Harlin, U. (2002) ‘A Survey of the Use of the Discrete-Event Simulation in Manufacturing Industry’, p. 5.

Ivanov, V. *et al.* (2019) ‘Technology for complex parts machining in multiproduct manufacturing’, *Management and Production Engineering Review*, Vol. 10, No. 2, pp. 25–36. doi:10.24425/mper.2019.129566.

Jung, W.-K. *et al.* (2020) ‘Real-time data-driven discrete-event simulation for garment production lines’, *Production Planning & Control*, 0(0), pp. 1–12. doi:10.1080/09537287.2020.1830194.

Kagermann, H. (2013) *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry ; Final Report of the Industrie 4.0 Working Group*. Forschungsunion.

Keller, J. (2002) ‘Human performance modeling for discrete-event simulation: workload’, in *Proceedings of the Winter Simulation Conference*. *Proceedings of the Winter Simulation Conference*, pp. 157–162 vol.1. doi:10.1109/WSC.2002.1172879.

Kelley, James E. and Walker, Morgan R. (1959) ‘Critical-path planning and scheduling’, in. *IRE-AIEE-ACM computer conference*. Available at: https://dl-acm-org.acces.bibl.ulaval.ca/doi/abs/10.1145/1460299.1460318 (Accessed: 22 November 2021).

Khayal, O. (2018) *A review for Dynamic Scheduling in Manufacturing*. doi:10.13140/RG.2.2.15345.33129.

Kher, H.V. and Fry, T.D. (2001) ‘Labour flexibility and assignment policies in a job shop having incommensurable objectives’, *International Journal of Production Research*, 39(11), pp. 2295–2311. doi:10.1080/00207540110036704.

Kress, D. and Müller, D. (2019) ‘Mathematical Models for a Flexible Job Shop Scheduling Problem with Machine Operator Constraints ∗∗This work has been supported by the European Union and the state North Rhine-Westphalia through the European Fund for Regional Development (EFRD). It has been conducted as part of the project “EKPLO: Echtzeitnahes kollaboratives Planen und Optimieren” (EFRE-0800463).’, *IFAC-PapersOnLine*, 52(13), pp. 94–99. doi:10.1016/j.ifacol.2019.11.144.

Lan, C.-H. and Lan, T.-S. (2005) ‘A combinatorial manufacturing resource planning model for long-term CNC machining industry’, *The International Journal of Advanced Manufacturing Technology*, 26(9–10), pp. 1157–1162. doi:10.1007/s00170-004-2090-y.

Lanzarone, E. and Matta, A. (2014) 'Robust nurse-to-patient assignment in home care services to minimize overtimes under continuity of care', *Operations Research for Health Care*, 3(2), pp. 48–58. doi:10.1016/j.orhc.2014.01.003.

Lasi, H. *et al.* (2014) 'Industry 4.0', *Business & Information Systems Engineering*, 6(4), pp. 239–242. doi:10.1007/s12599-014-0334-4.

Law, A.M. and Kelton, W.D. (1991) *Simulation modeling and analysis*. 2nd ed. New york: McGraw-Hill (McGraw-Hill series in industrial engineering and management science).

Le, V.T. *et al.* (2013) 'Dynamic control of skilled and unskilled labour task assignments', in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 955–960. doi:10.1109/AIM.2013.6584217.

Leusin, M.E. *et al.* (2018) 'Solving the Job-Shop Scheduling Problem in the Industry 4.0 Era', *Technologies*, 6(4), p. 107. doi:10.3390/technologies6040107.

Li, K., Xu, S. and Fu, H. (2020) 'Work-break scheduling with real-time fatigue effect and recovery', *International Journal of Production Research*, 58(3), pp. 689–702. doi:10.1080/00207543.2019.1598600.

Lidberg, S., Pehrsson, L. and Ng, A.H.C. (2018) 'Using aggregated discrete event simulation models and multi-objective optimization to improve real-world factories', in *2018 Winter Simulation Conference, WSC 2018, December 9, 2018 - December 12, 2018*. Gothenburg, Sweden: Institute of Electrical and Electronics Engineers Inc. (Proceedings - Winter Simulation Conference), pp. 2015–2024. doi:10.1109/WSC.2018.8632337.

Lin, Y. and Schrage, L. (2009) 'The global solver in the LINDO API', *Optimization Methods and Software*, 24(4–5), pp. 657–668. doi:10.1080/10556780902753221.

Liu, F. *et al.* (2017) 'On the Robust and Stable Flowshop Scheduling Under Stochastic and Dynamic Disruptions', *IEEE Transactions on Engineering Management*, 64(4), pp. 539–553. doi:10.1109/TEM.2017.2712611.

Manne, A.S. (1960) 'On the Job-Shop Scheduling Problem', *Operations Research*, 8(2), pp. 219–223.

Matt, D.T., Modrák, V. and Zsifkovits, H. (eds) (2020) *Industry 4.0 for SMEs: Challenges, Opportunities and Requirements*. Springer Nature. doi:10.1007/978-3-030-25425-4.

Meisels, A. and Schaerf, A. (2003) 'Modelling and Solving Employee Timetabling Problems', *Annals of Mathematics and Artificial Intelligence*, 39(1), pp. 41–59. doi:10.1023/A:1024460714760.

Meng, L. *et al.* (2020) 'Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem', *Computers & Industrial Engineering*, 142, p. 106347. doi:10.1016/j.cie.2020.106347.

Munasingha, K. and Adikariwattage, V. (2020) ‘Discrete Event Simulation Method to Model Passenger Processing at an International Airport’, in *2020 Moratuwa Engineering Research Conference (MERCon)*. *2020 Moratuwa Engineering Research Conference (MERCon)*, pp. 401–406. doi:10.1109/MERCon50084.2020.9185370.

Nehme, C., Crandall, J. and Cummings, M. (2008) *Using Discrete-Event Simulation to Model Situational Awareness of Unmanned-Vehicle Operators*. Available at: https://www.semanticscholar.org/paper/Using-Discrete-Event-Simulation-to-Model-Awareness-Nehme-Crandall/cdb956db32c9e46158cf4305eef7c6534477c46e (Accessed: 31 January 2022).

Nethercote, N. *et al.* (2007) ‘MiniZinc: Towards a standard CP modelling language’, in *Proceedings of the 13th International Conference on the Principles and Practice of Constraint Programming*. *International Conference on Principles and Practice of Constraint Programming 2007*, Springer-Verlag London Ltd., pp. 529–543. Available at: https://research.monash.edu/en/publications/minizinc-towards-a-standard-cp-modelling-language (Accessed: 21 April 2021).

Oliveira, E., Smith, B.M. and Jt, L. (2000) *A Job-Shop Scheduling Model for the Single-Track Railway Scheduling Problem*. Problem, Research Report 2000.21, University of Leeds. OpenTrack Railway Technology, Railway Simulation. http://www.opentrack.ch.

Onggo, B., Hill, J. and Brooks, R. (2013) ‘A pilot survey on data identification and collection in simulation projects’, in. *Modelling and Simulation 2013 - European Simulation and Modelling Conference, ESM 2013*.

Özkul, A.O. *et al.* (2021) ‘An Implementation of Flexible Job Shop Scheduling Problem in a Metal Processing Company’, in Durakbasa, N.M. and Gençyılmaz, M.G. (eds) *Digital Conversion on the Way to Industry 4.0*. Cham: Springer International Publishing (Lecture Notes in Mechanical Engineering), pp. 817–830. doi:10.1007/978-3-030-62784-3_68.

Pabla, B.S. and Adithan, M. (1994) *CNC Machines*. New Age International.

Palombarini, J.A. and Martínez, E.C. (2019) ‘Closed-loop Rescheduling using Deep Reinforcement Learning’, *IFAC-PapersOnLine*, 52(1), pp. 231–236. doi:10.1016/j.ifacol.2019.06.067.

de Paula Ferreira, W., Armellini, F. and De Santa-Eulalia, L.A. (2020) ‘Simulation in industry 4.0: A state-of-the-art review’, *Computers & Industrial Engineering*, 149, p. 106868. doi:10.1016/j.cie.2020.106868.

Pegden, C.D. (2008) ‘Introduction to Simio’, in *2008 Winter Simulation Conference*. *2008 Winter Simulation Conference*, pp. 229–235. doi:10.1109/WSC.2008.4736072.

Pegden, C.D. (2009) ‘An Introduction to Simio® for Beginners’, p. 7.

Pfitzer, F. *et al.* (2018) ‘Event-Driven Production Rescheduling in Job Shop Environments’, in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. *2018*

IEEE 14th International Conference on Automation Science and Engineering (CASE), pp. 939–944. doi:10.1109/COASE.2018.8560523.

Popkova, E.G., Ragulina, Y.V. and Bogoviz, A.V. (eds) (2019) *Industry 4.0: Industrial Revolution of the 21st Century*. Cham: Springer International Publishing (Studies in Systems, Decision and Control). doi:10.1007/978-3-319-94310-7.

Rahmani, D. and Ramezanian, R. (2016) 'A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study', *Computers & Industrial Engineering*, 98, pp. 360–372. doi:10.1016/j.cie.2016.06.018.

Robinson, S. (2005) 'Discrete-event simulation: from the pioneers to the present, what next?', *Journal of the Operational Research Society*, 56(6), pp. 619–629. doi:10.1057/palgrave.jors.2601864.

Rohde, D. (2019) *Dynamic simulation of future integrated energy systems*.

Rossi, E.F., van Beek, P. and Walsh, T. (2006) 'Handbook of Constraint Programming', p. 969.

Salido, M.A. *et al.* (2017) 'Rescheduling in job-shop problems for sustainable manufacturing systems', *Journal of Cleaner Production*, 162, pp. S121–S132. doi:10.1016/j.jclepro.2016.11.002.

Sargent, R.G. (2004) 'Validation and verification of simulation models', in *Proceedings of the 36th conference on Winter simulation*. Washington, D.C.: Winter Simulation Conference (WSC '04), pp. 17–28.

Schlesinger (1979) 'Terminology for model credibility', *SIMULATION*, 32(3), pp. 103–104. doi:10.1177/003754977903200304.

Schmidt, J.W. and Taylor, R.E. (1970) *Simulation and analysis of industrial systems*. Homewood, Ill: R. D. Irwin (Irwin series in quantitative analysis for business).

Semini, M., Fauske, H. and Strandhagen, J.O. (2006) 'Applications of Discrete-Event Simulation to Support Manufacturing Logistics Decision-Making: A Survey', in *2006 Winter Simulation Conference*. *2006 Winter Simulation Conference*, pp. 1946–1953. doi:10.1109/WSC.2006.322979.

Sicong, T., Weng, W. and Shigeru, F. (2009) 'Scheduling of Worker Allocation in the Manual Labor Environment with Genetic Algorithm', *Lecture Notes in Engineering and Computer Science*, 2174.

Sierra, M.R., Mencía, C. and Varela, R. (2015) 'New schedule generation schemes for the job-shop problem with operators', *Journal of Intelligent Manufacturing*, 26(3), pp. 511–525. doi:10.1007/s10845-013-0810-6.

Song, T. *et al.* (2020) 'Multi-skill aware task assignment in real-time spatial crowdsourcing', *GeoInformatica*, 24(1), pp. 153–173. doi:10.1007/s10707-019-00351-4.

Sotskov, Yu.N. and Shakhlevich, N.V. (1995) ‘NP-hardness of shop-scheduling problems with three jobs’, *Discrete Applied Mathematics*, 59(3), pp. 237–266. doi:10.1016/0166-218X(95)80004-N.

Sreekara Reddy, M.B.S. *et al.* (2018) ‘An effective hybrid multi objective evolutionary algorithm for solving real time event in flexible job shop scheduling problem’, *Measurement*, 114, pp. 78–90. doi:10.1016/j.measurement.2017.09.022.

Trilling, L., Guinet, A. and Magny, D.L. (2006) ‘Nurse scheduling using integer linear programming and constraint programming’, *IFAC Proceedings Volumes*, 39(3), pp. 671–676. doi:10.3182/20060517-3-FR-2903.00340.

Tsarouchi, P., Makris, S. and Chryssolouris, G. (2016) ‘Human–robot interaction review and challenges on task planning and programming’, *International Journal of Computer Integrated Manufacturing*, 29(8), pp. 916–931. doi:10.1080/0951192X.2015.1130251.

Turker, A.K. *et al.* (2019) ‘A Decision Support System for Dynamic Job-Shop Scheduling Using Real-Time Data with Simulation’, *Mathematics*, 7(3), p. 278. doi:10.3390/math7030278.

Turner, C.J. *et al.* (2016) ‘Discrete Event Simulation and Virtual Reality Use in Industry: New Opportunities and Future Trends’, *IEEE Transactions on Human-Machine Systems*, 46(6), pp. 882–894. doi:10.1109/THMS.2016.2596099.

Uhlmann, I.R. and Frazzon, E.M. (2018) ‘Production rescheduling review: Opportunities for industrial integration and practical applications’, *Journal of Manufacturing Systems*, 49, pp. 186–193. doi:10.1016/j.jmsy.2018.10.004.

Vieira, G., Herrmann, J. and Lin, E. (2003) ‘Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods’, *J. Scheduling*, 6, pp. 39–62. doi:10.1023/A:1022235519958.

Vielma, J.P. (2015) ‘Mixed Integer Linear Programming Formulation Techniques’, *SIAM Review*, 57(1), pp. 3–57. doi:10.1137/130915303.

Walsh, T. (2001) ‘Stochastic Constraint Programming’, p. 7.

Waschneck, B. *et al.* (2016) ‘Production Scheduling in Complex Job Shops from an Industry 4.0 Perspective: A Review and Challenges in the Semiconductor Industry’, in *SAMI@iKNOW*.

Watson, J.-P. and Beck, J.C. (2008) ‘A Hybrid Constraint Programming / Local Search Approach to the Job-Shop Scheduling Problem’, in Perron, L. and Trick, M.A. (eds) *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Berlin, Heidelberg: Springer (Lecture Notes in Computer Science), pp. 263–277. doi:10.1007/978-3-540-68155-7_21.

Wichmann, R.L., Eisenbart, B. and Gericke, K. (2019) ‘The Direction of Industry: A Literature Review on Industry 4.0’, *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1), pp. 2129–2138. doi:10.1017/dsi.2019.219.

Xia, W. and Wu, Z. (2005) ‘An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems’, *Computers & Industrial Engineering*, 48(2), pp. 409–425. doi:10.1016/j.cie.2005.01.018.

Xie, J. *et al.* (2019) ‘Review on flexible job shop scheduling’, *IET Collaborative Intelligent Manufacturing*, 1(3), pp. 67–77. doi:10.1049/iet-cim.2018.0009.

Xu, J., Xu, X. and Xie, S.Q. (2011) ‘Recent developments in Dual Resource Constrained (DRC) system research’, *European Journal of Operational Research*, 215(2), pp. 309–318. doi:10.1016/j.ejor.2011.03.004.

Yao, N. and Zhang, F. (2020) *Optimal Real-time Scheduling of Human Attention for a Human and Multi-robot Collaboration System*, p. 35. doi:10.23919/ACC45564.2020.9147782.

Yow, A. *et al.* (2005) ‘Predicting nuclear power-plant operator performance using discrete event simulation’, *Cognition, Technology & Work*, 7(1), pp. 29–35. doi:10.1007/s10111-004-0167-x.

Zammori, F. and Bertolini, M. (2015) *A Conceptual Framework for Project Scheduling with Multi-Skilled Resources*. doi:10.2991/aiie-15.2015.103.

Zhang, J. *et al.* (2019) ‘Review of job shop scheduling research and its new perspectives under Industry 4.0’, *Journal of Intelligent Manufacturing*, 30(4), pp. 1809–1830. doi:10.1007/s10845-017-1350-2.

Zhang, S. and Wang, S. (2018) ‘Flexible Assembly Job-Shop Scheduling With Sequence-Dependent Setup Times and Part Sharing in a Dynamic Environment: Constraint Programming Model, Mixed-Integer Programming Model, and Dispatching Rules’, *IEEE Transactions on Engineering Management*, 65(3), pp. 487–504. doi:10.1109/TEM.2017.2785774.

Zheng, P. *et al.* (2018) ‘Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives’, *Frontiers of Mechanical Engineering*, 13(2), pp. 137–150. doi:10.1007/s11465-018-0499-5.

Zhu, H. *et al.* (2019) ‘An Adaptive Real-Time Scheduling Method for Flexible Job Shop Scheduling Problem With Combined Processing Constraint’, *IEEE Access*, 7, pp. 125113–125121. doi:10.1109/ACCESS.2019.2938548.

Zupan, H. and Herakovic, N. (2015) ‘Production line balancing with discrete event simulation: A case study’, *IFAC-PapersOnLine*, 48(3), pp. 2305–2311. doi:10.1016/j.ifacol.2015.06.431.