Scheduling Hybrid Flow Lines of Aerospace Composite Manufacturing Systems

Aria Azami

A Thesis
in
The Department
of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Industrial Engineering) at
Concordia University
Montreal, Quebec, Canada

January 2016

© Aria Azami, 2016

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Aria Azami

Entitled: Scheduling Hybrid Flow Lines of Aerospace Composite Manufacturing Systems

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science in Industrial Engineering

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Wen-Fang Xie	Chair
Dr. Yong Zeng	Examiner
Dr. Onur Kuzgunkaya	Examiner
Dr. Nadia Bhuiyan	Supervisor
Dr. Kudret Demirli	Co-Supervisor

Approved by ____
Chair of Department or Graduate Program Director

____
Dean of Faculty

Date ____

Abstract

Scheduling Hybrid Flow Lines of Aerospace Composite Manufacturing Systems

Composite manufacturing is a vital part of aerospace manufacturing systems. Applying effective scheduling within these systems can cut the costs in aerospace companies significantly. These systems can be characterized as two-stage Hybrid Flow Shops (HFS) with identical, non-identical and unrelated parallel discrete-processing machines in the first stage and non-identical parallel batch-processing machines in the second stage. The first stage is normally the lay-up process in which the carbon fiber sheets are stacked on the molds (tools). Then, the parts are batched based on the compatibility of their cure recipe before going to the second stage into the autoclave for curing. Autoclaves require enormous capital investment and maximizing their utilization is of utmost importance.

In this thesis, a Mixed Integer Linear Programming (MILP) model is developed to maximize the utilization of the resources in the second stage of this HFS. CPLEX, with an underlying branch and bound algorithm, is used to solve the model. The results show the high level of flexibility and computational efficiency of the proposed model when applied to small and medium-size problems. However, due to the NP-hardness of the problem, the MILP model fails to solve large problems (i.e. problems with more than 120 jobs as input) in reasonable CPU times.

To solve the larger instances of the problem, a novel heuristic method along with a Genetic Algorithm (GA) are developed. The heuristic algorithm is designed based on a careful observation of the behavior of the MILP model for different problem sets. Moreover, it is enhanced by adding a number of proper dispatching rules. As its output, this heuristic algorithm generates eight initial feasible solutions which are then used as the initial population of the proposed GA.

The GA improves the initial solutions obtained from the aforementioned heuristic through its stochastic iterations until it reaches the satisfactory near-optimal solutions. A novel crossover operator is introduced in this GA which is unique to the HFS of aerospace composite manufacturing systems. The proposed GA is proven to be very efficient when applied to large-size problems with up to 300 jobs. The results show the high quality of the solutions achieved by the GA when compared to the optimal solutions which are obtained from the MILP model.

A real case study undertaken at one of the leading companies in the Canadian aerospace industry is used for the purpose of data experiments and analysis.

Acknowledgements

First and foremost, I would like to express my sincere gratitude and appreciation to my supervisors, Professor Nadia Bhuiyan and Professor Kudret Demirli, for their encouragement, understanding, and generosity. I am and always will be grateful to Dr. Bhuiyan for her continuous support, guidance, and patience, as well as for providing a friendly and comfortable research environment for me and all her students. Also, I warmly thank Dr. Demirli for giving me the opportunity to become a member of his research group. It has been a prodigious honour working with a highly professional scholar such as him. Dr. Demirli's intelligence, wisdom, and assiduous commitment to the highest standards was a continuous inspiration and motivation during my studies.

Moreover, I would like to extend my gratitude to Dr. Onur Kuzgunkaya who patiently organized the meetings with the case company and helped me with his invaluable advice and support throughout.

Furthermore, I consider myself extremely fortunate to have many compassionate friends who have supported and encouraged me. I owe my deepest gratitude to Helia Kazerooni for her unlimited kindness, and a special 'thank you' to both Mehdi Karimian and Sajed Kheyrollahi who make time for me regardless of their schedule. I am also indebted to my friend, Dr. Ehsan Mardan, who has constantly assisted me with his insightful advice.

In the end, I owe my profound appreciation to my beloved family. My parents, Baba Hassan and Maman Narges, have taught me the meaning of unconditional love, and it is their boundless affection and patience that made this entire effort possible. Finally, I would like to thank my beloved elder brother who has always believed in me and provided me with such a pure love and support without which I would have never felt strong and confident enough to pursue my goals and become the person I am today. No words can express how very much I love and appreciate you Yashar; this thesis is dedicated to you.

Table of Contents

List of Figures

List of Tables

1 Chapter One – Introduction

This chapter includes an overview on the background of the problem which is dealt with in this thesis along with an introduction of the research problem. A summary of the case company's characteristics and conditions are provided and the objectives and the methodology used in this research are discussed. Finally, the outline of this thesis is presented in the last section.

1.1 Background

In today's global economy with international companies that source products and materials all over the world to get the best quality and the lowest cost, the need for speed is felt more than ever. These conditions led to a new era of production known as mass customization that aims at combining low costs of mass production and high flexibility of individual customization. However, the emergence of global outsourcing has made the management of supply chains even more complicated and challenging. Nowadays, there are a lot of economic, environmental, and political risks associated with global supply chains.

With the advent of new technologies, the efficiency of planning and management of supply chains is highly improved. These technologies (e.g. Electronic Data Interchange (EDI), Enterprise Resource Planning (ERP), Supply Chain Planning along with e-commerce, etc.) have provided a proper ground for today's mass customization. On the other hand, the supply chains are threatened by many financial, environmental and political disruptions as well as many unplanned events. Managing the supply chain with all these changes and disruptions seems quite difficult and complicated but there are also great opportunities for identifying and eliminating waste and creating more value within the supply chains (Myerson, 2012).

Supply Chain Management includes the planning and management of all activities related to sourcing, procurement, conversion and logistics as well as coordination and collaboration with suppliers, intermediaries, service providers and customers. There are also some functional areas such as information management, inventory flow scheduling and control, transportation systems operation and infrastructure, distribution facilities management, and customer service (Wisner et al., 2015).

Aerospace supply chains are among the most complicated systems that normally are characterized by high demand variability and long lead times (Rose-Anderssen et al., 2008). Different layers and tiers of suppliers, procurement, manufacturing centers, assembly plants, and customers (internal and external) are the nodes of a typical aerospace supply chain. A valid question that may come to one's mind here is that why researchers focus on aerospace supply chains and try to identify wasteful resources within these chains.

The first reason is that the aerospace supply chain is a major cost center. It's proven that reducing costs in the supply chain that causes the equivalent profit, is much easier than increasing sales (Myerson, 2012). The second reason is the bullwhip effect which means inventory, operational cost, and demand variability increase as one moves up a supply chain (from customer toward the upstream suppliers). Last but not least, even though the new era of increased outsourcing and global supply chains with shorter product life cycles has created a great opportunity of exposure to the worldwide business, it also has put pressure on the aerospace supply chain (which is sometimes more of a net than a chain) to be more efficient (Myerson, 2012).

Among all the main activities included in an aerospace supply chain, composite manufacturing is one of the most advanced and complex areas that has attracted a great deal of scientific attention in the last few years. The application of composite materials (carbon fiber sheets) in airplanes and helicopters is increasing rapidly (Lubin, 2013).

There are promising opportunities for identifying improvement potentials in composite manufacturing systems in order to cut the supply chain costs for the aerospace industry. In the following section, we will introduce and discuss an example of these great opportunities which is also the core research focus of this thesis; designing an effective production planning and scheduling system which enables the composite manufacturing systems to meet the demand of their customers without keeping large amounts of inventory (safety stock) or imposing frequent production accelerations or decelerations.

1.2 Research Problem

Composite Manufacturing Systems normally consist of several stages and processes with combinatorial relations and interactions among various processes in different stages, which make them complicated manufacturing systems. Normally, there are numerous issues and difficulties associated with managing such complex systems, the most significant of which are variability and instability. High levels of demand variability and long lead times in aerospace manufacturing systems combined with the complex nature of Composite Manufacturing Systems causes different types of variability (i.e. production rate, capacity utilization, amounts of inventory, etc.) in the system. In addition, complicated processes inside composite manufacturing systems which require high levels of operator expertise usually cause high yield variance and considerable scrap rates, which in turn make the system unstable.

Since managing the inventory in these systems is very complicated, the tendency to keep more safety stock is increased. However, keeping inventory has its own problems and is proven to be one of the most important sources of waste. To refrain from keeping too much inventory, it is essential to find the root causes of variability and eliminate them (Heizer et al., 2013).

As a matter of fact, batch size reduction is a very critical principle of today's production systems that are associated with mass customization. Unlike traditional systems in which production in large quantities are desirable (because it will spread the fixed costs among a large number of products and reduce the per-unit cost), in modern production systems, planners try to schedule the production according to the customer's actual demand. The ideal situation is one-piece flow and although it may not be attained, heading toward it is beneficial. The benefits include reduced lead times, less inventories, more flexibility to meet the uncertain demand, better quality with less scrap, rework and space (Liker, 2005).

These are all possible by means of effective scheduling systems. Scheduling is known as a decision-making process that deals with the allocation of the available resources to tasks over specified time periods (Pinedo, 2012).

This decision-making process plays a pivotal role in managing the resources within complex systems such as composite manufacturing systems and it can cut the costs of aerospace companies significantly. Therefore, in order to reduce the variability in composite manufacturing systems, a major step is to develop a powerful production planning and scheduling system which enables the system to meet the demand of the customers without imposing frequent production accelerations or decelerations and without feeling the need to keep large amounts of inventory.

The composite manufacturing systems in aerospace companies provide their production lines with the required composite parts. Thus, they are usually considered internal suppliers of aerospace companies' production lines.

Improving the efficiency of these suppliers can shorten the overall lead times of the aerospace supply chains dramatically and improve the efficiency of the resources inside the composite manufacturing systems while reducing the inventories and unnecessary waiting times. There are normally various processes and work stations within these systems. The most significant challenge in these manufacturing systems is to identify the processes where the jobs need to be scheduled and to schedule the jobs in those particular processes in a way that other processes can be synchronized with them and work at the rate of customer's demand.

The problem that we face in this research is to observe the Composite Manufacturing Systems of aerospace companies in order to provide visibility on their operations and identify the opportunities for improvement. Afterwards, reducing the sources of variability by a proper method of scheduling the jobs inside these systems is tackled. In other words, in this research we aim at developing a scheduling system for Composite Manufacturing Systems.

These systems can be characterized as two-stage Hybrid Flow Shops (HFS) with identical and unrelated parallel discrete-processing machines in the first stage and identical parallel batch-processing in the second stage. The first stage is normally the lay-up process in which the carbon fiber sheets are stacked on the molds (tools). Then, the parts are batched based on the compatibility of their cure recipe before going to the second stage into the autoclave for curing. Autoclaves require enormous capital investment and maximizing their utilization is of utmost importance.

A case study of a leading company in the Canadian aerospace industry, which we will hereinafter refer to as the ABC Company for the purpose of confidentiality, is used to validate our model.

1.3 Objectives and Methodology

The objective of this thesis is to schedule jobs in a two-stage Hybrid Flow Shop (HFS) with identical, non-identical and unrelated parallel discrete-processing machines in the first stage (Lay-up Cells) and non-identical parallel batch-processing machines in the second stage (Autoclaves). To the best of our knowledge, this is the first time in the literature of scheduling problems that this particular type of HFS is studied.

The optimal schedule will result in synchronizing the sequence of work done in the lay-up process (first stage) to the autoclaves' cure sequence (second stage). Autoclaves require enormous capital investment and maximizing their utilization is of utmost importance. Therefore, the main objective of this study is to maximize the utilization of the resources (autoclaves) in the second stage of the described Hybrid Flow Shop.

Minimizing the tooling cycle time is another result of such an optimal schedule that in turn will open up extra capacity in the lay-up and cure processes. In addition, providing visibility on the utilization of the resources in the lay-up cells and the capacity usage of autoclaves is of interest.

As the first step of this research, we have identified and reviewed the best practices which are done in the area of two-stage Hybrid Flow Shops with discrete and/or batch processes. A number of similar industrial cases have been studied to find the ways researchers approach these sorts of problems. This helped us identify the methods used to model the real problems of Hybrid Flow Shop scheduling and the techniques used to solve such models.

Value Stream Mapping (VSM) is used to depict the process of composite manufacturing in the aerospace industry from the beginning to the end in order to study the flow of material and information within these systems.

The next step was to come up with a mathematical model that represents all the objectives and constraints we are dealing with in this problem and in this particular case study. The model should imitate the actual conditions of the case study considering all the limitations and special characteristics of the Composite Manufacturing System.

A Time-indexed Mixed Integer Linear Programming model is developed for this purpose. CPLEX with an underlying branch and bound algorithm is used to solve the model. The results show the high level of flexibility and computational efficiency of the proposed model when applied to the real case study of Composite Manufacturing Center at the ABC Company, especially for small and medium-size problems.

Afterwards, for solving the large instances of the problem, a heuristic solution method is proposed. The heuristic method is designed based on a careful observation of the behavior of the MILP model and using a number of dispatching rules drawn from the literature of scheduling problems. Combining this knowledge with the information we have about the special characteristics of Composite Manufacturing Systems, this novel heuristic is established to generate eight feasible initial solutions to the original problem. Afterwards, a Genetic Algorithm is developed that uses the outputs of the aforementioned heuristic method as its initial population and generates improved solutions in its stochastic iterations and finally achieves near-optimal solutions.

A case study of a leading company in the Canadian aerospace industry is used to validate the proposed models. Various sets of data from the Composite Manufacturing Center (CMC) of the case company are used for numerical experiments and data analysis.

A thorough data experiment/analysis is conducted to compare the results achieved from the MILP model with those obtained from the heuristic method and the GA. Another main objective of this research is to develop a scheduling tool (according to the hypotheses and assumptions made by the case company) that receives data from the MRP system as input and delivers the sequence of work and the work schedule for the two stages of the Composite Manufacturing System.

1.4 Organization of the Thesis

This thesis is organized into six chapters. Following the introductory chapter, the second chapter provides a comprehensive literature review. The third chapter begins with a description of the case study, followed by the detailed problem description and the MILP model formulation. The solution approach for small and medium-size problems are also discussed in this chapter. In chapter four, a heuristic method along with a GA are presented to show how the near-optimal solutions for larger instances of the problem are achieved.

Chapter five includes the numerical experiments. Some numerical examples of the MILP model as well as the heuristic and the GA are presented and solved and the results are analyzed in order to compare different methods. Finally, in chapter six, conclusions, limitations and the possible avenues for future research are discussed.

2 Chapter Two – Literature Review

This chapter reviews and discusses the existing research in the area of Hybrid Flow Shop scheduling. It starts with a definition of HFS environments and batch scheduling problems and continues with a categorization of the various methodologies used to model and solve such problems. Finally, a short summary along with the detected gaps in the literature and the potential research opportunities are presented.

2.1 Hybrid Flow Shops

Hybrid Flow Shops are manufacturing environments in which there is a group of machines and they are arranged into s stages in series. At stage l, l=1, …, s, there are M(l) identical machines in parallel. Job j, j=1, …, n, has to be processed on any one machine at each stage. The processing times of jobs j at the various stages are P (1, j), P (2, j),…, P (s, j). Preemption is allowed in some HFS problems and is not allowed in others. If preemption is not allowed, then once a job starts being processed on a machine, it has to be finished before any other job can be processed on that machine. Each machine can process at most one job at a time. The buffer that stores the jobs waiting for processing at each stage has limited storage capacity in some HFS problems and unlimited storage capacity in others (Pinedo, 2012).

In this research, we aim at studying the scheduling problem of a two-stage hybrid flow shop with identical, non-identical and unrelated parallel discrete-processing machines in the first stage and non-identical parallel batch-processing machines in the second stage. To do so, we have conducted a literature review focusing on scheduling hybrid flow shops that contain discrete and/or batch processing machines.

The objective of HFS scheduling problems is typically to minimize the make-span, C_{max}, or the completion time of the last job to leave the system. This objective implicitly attempts to maximize the utilization of the system. However, in the case of this research, the objective is to maximize the utilization of the resources in the cure process explicitly.

2.2 Batch Scheduling Problem

There are two main types of batching in scheduling problems. One situation is when different jobs require different setups. These setups may call for changing tools or cleaning the machine. In these cases, jobs are usually grouped in families based on their similarities in a way that no setup is required for a job if the previously processed job was in the same family. These are called family scheduling models (*serial-batching scheduling*). Another situation is where a batching machine can process a number of jobs simultaneously. These are called batching machine models in which a machine processes a batch of jobs at the same time (Potts and Kovalyov, 2000). Scheduling of batch processing machine is also referred to as *parallel-batching scheduling* in the literature (Mathirajan and Sivakumar, 2006).

Many researchers have studied serial batching scheduling problems. They have used various modeling approaches and diverse solution methodologies to address this type of problems (Baker and Schrage, 1978; Zdrzalka, 1991; Kovalyov and Potts, 1992; Ng et al., 2003; Yuan et al., 2007; Yimer and Demirli, 2009; Tang and Liu, 2009; Shen and Buscher; 2012). However, in this research we will focus on the parallel-batching scheduling problems in flow shops and hybrid flow shops.

In the following, we have categorized the previous studies based on their modeling and solution methodology. The methodologies used in the literature mainly fall in four main categories of Heuristic Algorithms and Dispatching Rules (simple/or combined with local search algorithms), Dynamic Programming, Mixed and Pure Integer Programming, and Meta-heuristic methods. The table in the next page shows the studies done in each of the aforementioned categories.

Table 1. Four Main Methodologies to Model and Solve HFS Scheduling Problems

Number	Modeling/Solution Methodology	Research (Authors and Year)
1	Heuristic Algorithms and Dispatching Rules	Ahmadi et al. (1992) Ovacik and Uzsoy (1995) Sawik (1995) Sotskov et al. (1996) Chandra and Gupta (1997) Hoogeveen and Velde (1998) Danneberg et al. (1999) Blomer and Gunther (2000) Sung et al. (2000) Wang et al. (2001) Wang and Li (2002) Sung and Kim (2003) Lin and Cheng (2005) Oulamara et al. (2009) Yurtsever et al. (2009) Bellanger and Oulamara (2009) Wang et al. (2012)
2	Dynamic Programming	Sung and Yoon (1997) Brucker et al. (1997) Sung and Min (2001) Xuan and Tang (2007)
3	Mixed and Pure Integer Programming	Pinto and Grossmann (1995) Guinet and Solomon (1996) Bhatnagar et al. (1999) Sawik (2000) Blomer and Gunther (2000) Méndez et al. (2001) Sawik (2002) Su (2003) Damodaran and Srihari (2004) Sawik (2005) Low (2005) Sawik (2006) Voss and Witt (2007) Jenabi et al. (2007) Klemmt and Hielscher (2008) Ruiz et al. (2008) Klemmt et al. (2009) Amin-Naseri and Beheshti-Nia (2009) Gong et al. (2010) Behnamian and Fatemi Ghomi (2011) Ziaeifar et al. (2012) Klemmt and Mönch (2012) Gicquel et al. (2012) Wang et al. (2012) Rossi et al. (2013) Rossi et al. (2014)

Number	Modeling/Solution Methodology		Research (Authors and Year)
4	Meta-heuristic Methods	Genetic Algorithm	Kim and Kim (2002) Wang and Li (2002) Balasubramanian et al. (2004) Reichelt and Mönch (2006) Jenabi et al. (2007) Luo et al. (2009) Luo et al. (2011) Behnamian and Fatemi Ghomi (2011) Ziaeifar et al. (2012)
		Simulated Annealing	Danneberg et al. (1999) Mathirajan et al. (2004) Low (2005) Jenabi et al. (2007)
		Ant Colony System	Raghavan and Venkataramana (2006) Li et al. (2008) Mönch and Almeder (2009)
		Neural Networks	Mönch et al. (2006)
		Variable Neighborhood Search (VNS)	Klemmt et al. (2009) Behnamian and Fatemi Ghomi (2011)
		Tabu Search	Wang and Tang (2009)
		Hybrid Particle Swarm Optimization (HPSO) Algorithm	Liu et al. (2010)
		Greedy Randomized Adaptive Search Procedure (GRASP)	Damodaran et al. (2011)

2.3 Heuristic Algorithms and Dispatching Rules

Ahmadi et al. (1992) study the complexity of a class of two-machine batching and scheduling problems for the objectives of minimizing Cmax and ∑Cj. In the two-stage flow shops that they analyze, one or both of the machines may be a batching machine. According to their notation, β denotes a batch processor, δ denotes a discrete processor and → denotes the system configuration. For instance, δ→β means a discrete processor followed by a batching machine. They show that there exist polynomial time procedures to solve δ→β and β→β for both objectives of minimizing Cmax and ∑Cj as well as β→δ with the objective of minimizing Cmax. For β→δ with the objective of minimizing ∑Cj, they prove the problem is NP-complete. Besides the proposed Integer Programming formulation, they present a heuristic and determine an upper bound on the worst case performance ratio of their heuristic. In addition, they analyze the case of three machine flow shop as well as multiple family problems.

Ovacik and Uzsoy (1995) develop a number of rolling horizon algorithms to minimize maximum lateness of jobs in a parallel machine environment with sequence dependent setup times and unequal job arrival times. The numerical experiments of their research show that their Rolling Horizon Procedures (RHP) outperforms the famous Early Due Date (EDD) dispatching rule by 69% on average. They also give schedules 38% better than those which are derived from EDD coupled with local search.

Potts et al. (2001) analyze the complexity of two-machine batching problems in open shop, job shop and flow shop environments. They study cases of bounded and unbounded batch sizes on a machine with the objective of minimizing the completion time of the last job (make-span), and establish the complexity status of the problem in each case.

Sung and Kim (2003) study a two-machine flow shop composed of two batch processes and develop three heuristic algorithms to minimize maximum tardiness, number of tardy jobs and total tardiness, respectively.

Lin and Cheng (2005) study a flow shop with two stages in which the first stage is a discrete processing machine and the second stage is a batch processing machine with a constant setup time for each batch it processes. They prove the strong NP-hardness of the problem and show some special polynomial time solvable cases of the problem. They also suggest that an optimal solution for these special cases is a lower bound for the general problem. They propose some heuristics to obtain near optimal solutions to the general problem and through numerical experiments show that the error ratios are small.

Oulamara et al. (2009) analyze a scheduling problem of a two machine flow shop in which the first machine is a discrete processor (tire building process) and the second machine is a batch processor (tire curing process). They also define compatibility relations between each pair of jobs and draw an undirected compatibility graph to demonstrate these compatibility relations. The batch processing time in the second machine is defined as the longest processing time of the jobs contained in the batch. They show the NP-hardness of the problem when the objective is to minimize the make-span and develop three heuristic algorithms to make batching and sequencing decisions.

Bellanger and Oulamara (2009) study a two stage hybrid flow shop in which the machines in the first stage are identical parallel discrete processors and the second stage consists of several identical parallel batch processing machines. The batch processors in the second stage can only process compatible jobs in each batch. So, they define a compatibility relation between each pair of tasks and obtain an undirected compatibility graph. They proposed three heuristic algorithms in order to minimize the make-span. They also conduct computational experiments to compare the average and worst case performance of the different proposed heuristics.

Wang et al. (2012) study the scheduling problem of a two-stage hybrid flow shop in which the first stage consists of parallel discrete processing machines while the second stage includes parallel batch processing machines. They develop a MIP model with the objective of minimizing the make-span which takes into account different aspects of the problem such as release times and due date constraints. They propose several heuristic algorithms to solve the problem most efficient of which is a dispatching rule called BFIFO which assigns batches to machines based on first-in-first-out rule and then employs a local search heuristic called Interchange, Translocation, and Transposition (ITT) for re-optimization (relocating the job positions in the sequence to obtain better solutions). The computational experiments suggest that the proposed technique is capable of generating good schedules for problems with up to 160 jobs on 40 machines within ten minutes.

2.4 Dynamic Programming

Sung and Yoon (1997) consider a system of two-machine flow shop with two batch processes in which a finite number of jobs are released dynamically and the objective is to minimize the maximum completion time of all jobs (make-span). They develop a Dynamic Programming algorithm based on different solution properties they analyzed.

Brucker et al. (1997) study the scheduling of a single batching machine with bounded and unbounded batch sizes. They develop different Dynamic Programming models for both cases with objective functions of minimizing total weighted completion time, maximum lateness, weighted number of late jobs and total weighted tardiness. They have mapped the time complexities of various single-machine batching problems with the same release dates of jobs.

Sung and Min (2001) aim at minimizing earliness/tardiness of jobs with a common due date in a two machine flow shop environment with three different structures. In the first case, a discrete processing machine is followed by a batching processing machine. In the second case, machines of both stages are batching processing machines. And, the third case consists of a batching processing machine which is followed by a discrete processing machine. For the first two cases, the authors propose polynomial time heuristic algorithms. For the third case, however, they prove that the problem is NP-complete and show that the optimal sequence of the problem is the same as the optimal sequence for the problem of a single discrete machine. Consequently, based on the work of Ventura and Weng (1995), they propose a pseudo-polynomial dynamic programming algorithm.

Xuan and Tang (2007) consider a hybrid flow shop problem in iron and steel industry with *s* stages that the last stage is a batch processing machine. They develop an integer Programming model with a time-indexed variable and use a Lagrangian Relaxation algorithm to decompose the problem into a set of batch-level sub-problems. In order to solve these sub-problems, they propose a forward dynamic programming algorithm.

2.5 Mixed/Pure Integer Programming

Pinto and Grossmann (1995) study the problem of scheduling multi-stage batch plants with parallel processing equipment and unlimited intermediate buffers. They develop a MIP model with continuous time variables in order to minimize the time interval between the end of processing of the jobs and their due dates (minimizing the earliness). The model is capable of solving small instances with less than 500 binary variables. However, for the larger instances of the problem, two methods are suggested. First, they propose a number of preordering constraints to reduce the number of nodes (decisions to be made in the branch and bound algorithm). Second, they develop a decomposition method that determines feasible assignments of the jobs with the objective of minimizing in-process time and then determines a schedule that eliminates unnecessary setups to minimize earliness. The numerical experiments on a real-world case demonstrate the capability of the proposed approach to solve a problem with 5 stages, 25 machines and up to 50 jobs.

Méndez et al. (2001) consider the problem of scheduling a resource-constrained flow shop with multiple stages and several batch processing machines working in parallel at each stage. They propose a continuous-time Mixed Integer Programing model to minimize the earliness of jobs with maximizing the weighted completion times for all orders in all stages with higher weights for later stages. The formulation structure of the model enables the possibility of adding topological constraints and sequence-dependent setup times without defining any additional variable or constraint. The model is applied to real-world problems and solved by CPLEX Mixed-Integer Optimizer and the results show that it is significantly powerful in terms of computational efficiency.

Sawik (2002) studied the problem of batch scheduling in hybrid flow shops with limited intermediate buffers. The hybrid flow shop under study consists of i consecutive processing stages and m parallel identical machines in each stage. He proposes a MIP model in order to minimize the make-span by processing the parts of one type consecutively (in serial batches). The numerical experiments show that the proposed MIP formulation can be effectively applied to small size problems. However, the CPU time required to find optimal solutions for realistic large size problems using the proposed MIP model can be very high.

Damodaran and Srihari (2004) analyze a two-machine flow shop with batch processes and develop two MIP models for the case of unlimited buffer capacity and no buffer capacity. The objective function of the model is to minimize Cmax. They have defined two sets of decision variables one of which is a binary assignment variable Zbk which is 1 if the batch b is scheduled as the kth scheduled batch and the other one is Xjb which is 1 if the job 1 is included in the batch b. They solve the problem with AMPL and CPLEX only for ten jobs and suggest using techniques such as branch-and-price and branch-and-cut for solving larger instances of the problem.

Sawik (2006) develops multi-objective Integer Programming formulations for obtaining Master Production Schedule of customer orders during long-term planning horizons in a hybrid flow shop environment with several parallel batch processing machines in each stage. He also develops a MIP model for machine assignments and scheduling of jobs over short-term planning horizons. To enhance the proposed models, he also adds a number of cutting constraints to each of them. For instance, in the case of the multi-objective Integer Programming model, the new cutting constraints relate the required capacity (to meet the customer demand) to the available capacity for each processing stage. They conduct computational experiments in a real-world large-sized case of a make-to-order assembly system in the electronics industry and observe that their hierarchical approach is capable of obtaining long-term master production schedules and short-term machine schedules.

Voss and Witt (2007) develop an Integer Programming model for their problem which is a combination of a Hybrid Flow Shop and Resource Constrained Project Scheduling Problem (RCPSP). The objective function of their model is to minimize the weighted tardiness. They divide the planning horizon T into t periods with equal length and define a time-indexed decision variable. They propose a heuristic solution procedure using dispatching rules and modify the rules in order to create batches.

Klemmt and Hielscher (2008) study different MIP as well as simulation-based optimization approaches to solve the problem of scheduling jobs in batch processes of semiconductor fabrication industry. They develop MIP models for two different problems with objectives of minimizing the make-span and minimizing the job cycle time. They also study the simulation-based optimization approaches for both of the problems. The results of the computational experiments demonstrate that both approaches give better results compared to the dispatching rules strategy taken by a real manufacturing case. The results also show that the MIP-based approaches give exact solutions or very good near-optimal solutions within the first five minutes of running the models. The simulation-based optimization approaches yield very good results especially for higher problem dimensions. In cases that problem dimensions of the MIP are too high, the simulation-based approaches are advisable.

Ruiz et al. (2008) study the problem of scheduling a hybrid flow shop with unrelated parallel machines, sequence-dependent setup times and unlimited intermediate buffers. They propose a MIP model as well as a number of heuristic methods to solve the large instances of the problem. They also propose an advanced statistical tool called AID which employs decision trees and Design of Experiments (DOE) in order to evaluate the effects of different factors on the performance of the MIP model and the heuristics algorithms. The results of their analysis suggest that many of the factors which are usually considered critical (ex. the setup to processing time ratio) have no effect on the difficulty of the MIP model and the proposed heuristics.

Amin-Naseri and Beheshti-Nia (2009) study a parallel batch scheduling problem in a hybrid flow shop. They formulate the problem as a MIP model and prove its NP-hardness. Then, in order to obtain near optimal solutions, they propose three heuristics which are developed based on Johnson's rule, parallel machine scheduling methods, and theory of constraints, respectively. They also introduce a novel GA with three dimensional chromosome structures and through numerical experiments show that the 3D GA outperforms all the aforementioned heuristics.

Klemmt and Mönch (2012) discuss different classes of nested time constrains in semiconductor wafer fabrication environments and study a scheduling problem of jobs in a flow shop with time constraints between consecutive stages. They propose a simple heuristic based on List Scheduling and a decomposition approach based on MIP formulation with the objective of minimizing total tardiness. Then, they design computational experiments and show that the MIP-based heuristic outperforms the simple list scheduling heuristic.

Gicquel et al. (2012) study a real-world case of hybrid flow shops in fermentation areas of a bio-process industry. The HFS consists of batch processing machines, multiprocessor tasks, and sequence-dependent setup and removal times. There are several additional constraints such as zero intermediate buffer capacity and limited waiting time of jobs between two successive production operations due to the formation of undesired by-products in case of waiting too long. In order to minimize the total weighted tardiness, they develop a MIP model with a time-indexed binary variable Y_{bst} which is equal to 1 if processing of batch b at stage s starts in time period t and equals to zero otherwise. Computational experiments of the research demonstrate that the model is capable of obtaining optimal schedules for real industrial size cases with up to 35 jobs in reasonable computational times. However, the authors recommend the use of meta-heuristics for future work so as to obtain solutions for larger instances within shorter times.

Rossi et al. (2014) consider MIP formulations for hybrid flow shop scheduling problem with parallel batching machines and compatible product families with the objective of minimizing the number of tardy jobs. They prioritize and sort the jobs based on the critical ratio (CR_{setup}) which is the available time between current time and due date divided by the processing time of the job. To obtain the critical ratios, they propose two heuristic methods one of which considers individual stages while the other one takes all stages into account. Numerical experiments of their research show that the combination of the critical ratio priority rule and the rolling horizon mechanism of the algorithm enables the proposed heuristics to efficiently reduce the number of tardy jobs as well as the make-span.

2.6 Meta-heuristic Methods

Due to the critical fact that hybrid flow shop scheduling problems are mainly NP-hard and finding the exact solution for such problems calls for very long computational times and costs (and even it is sometimes impossible), the use of meta-heuristic algorithms to find near-optimal solutions in relatively short times is very beneficial.

Therefore, during the last two decades, many researchers have tried to employ different meta-heuristic approaches to solve complicated scheduling problems. In the following, a review of research in this direction will be presented.

2.6.1 Genetic Algorithm

Kim and Kim (2002) study a two-machine flow shop in which the first stage is a batch machine and the second stage is a discrete machine. They propose a heuristic based on the GA approach to minimize the total completion time of the jobs and demonstrate the effectiveness of the GA-based algorithms in solving scheduling problems through computational experiments. Su (2003) studies the same problem and formulates it as a Mixed Integer Programing model with the objective of minimizing Cmax. He then develops a heuristic algorithm and presents its effectiveness through experimental investigations.

Reichelt and Mönch (2006) aim at minimizing total weighted tardiness and make-span at the same time in a semiconductor manufacturing environment with batch processing machines. They propose a multi-objective GA to develop Pareto efficient solutions and combine it with a local search technique to improve the quality of the solutions. The computational experiments of this research show the good solution quality of their proposed approach.

Luo et al. (2009) study the scheduling of a two-stage hybrid flow shop in metal-working company that has several parallel batching machines in the first stage and one single discrete machine in the second stage. There is no intermediate buffer between the two stages and there are times that machines are unavailable due to break downs or preventive maintenance practices. They consider two types of machine unavailability; deterministic and stochastic. They develop a mathematical model for the problem with objective function of minimizing the make-span and time-indexed variables for both stages. Then, they propose a GA with two decoding methods in order to solve the problem in both cases of deterministic and stochastic machine unavailability.

Luo et al. (2011) consider a real-world case in a metal working company which is a two-stage hybrid flow shop with parallel batch processing machines in the first stage and a single machine with sequence dependent setup times in the second stage. There are some additional constraints arising from the special characteristics of the studied problem such as blocking (no intermediate buffer between two stages) and conducting machine maintenance practices in a timely manner during the scheduling horizon. They develop linear programming formulations to model the problem. To obtain solutions, they propose a two-step algorithm. In the first step, jobs are grouped in batches based on their similarities and some grouping constraints. In the second stage, a GA is proposed which treats each batch like a job unit and determines the sequence of these units to be processed in the first stage. This approach is compared to the manual scheduling practices which are done currently by the engineers in the company and the results show the high superiority of the proposed algorithm with respect to the solution quality and the required computation time.

Ziaeifar et al. (2012) study the problem of scheduling a hybrid flow shop with unlimited intermediate buffers between stages and processors that move among stages and cooperate with processors of each stage in order to perform the operations. They develop a MIP model with the objective of minimizing the make-span and cost of assigning processors to stages. In order to compute the make-span, they propose a novel heuristic algorithm. Then, they propose a GA that uses the result of the aforementioned heuristic as a part of its fitness function. They apply their proposed algorithm to 20 test problems and compare the results with a lower bound in the literature (Shiau and Huang, 2011). The efficiency of the proposed algorithm is evidenced by the computational experiments.

2.6.2 Simulated Annealing

Low (2005) study the scheduling of a hybrid flow shop with unrelated parallel machines. He develops a MIP model with the objective of minimizing total flow time in the system. Since the MIP model is not a practical solution method for large size problems, he proposes a Simulated Annealing meta-heuristic to obtain near optimal solutions. He uses SPT/FCFS rule to generate initial solutions for the SA algorithm. Then he compares the proposed algorithm with two other SA algorithms one of which is standard SA with SPT/FCFS initial solutions heuristic and the other one is his proposed SA with randomly generated initial solutions. The results of the experimental analysis show that his proposed Simulated Annealing heuristic with SPT/FCFS initial solution heuristic outperforms the other two SA-based heuristics with respect to solution quality and efficiency.

Jenabi et al. (2007) study the Economic Lot-Scheduling Problem (ELSP) in a hybrid flow shop with unrelated parallel machines over a limited planning horizon. They develop a MIP model to minimize the sum of setup and inventory holding costs while preventing stock-outs. Since the MIP model is not practically solvable for large size problems, they propose two meta-heuristics based on Hybrid Genetic Algorithm (HGA) and Simulated Annealing approaches. They also propose two constructive heuristic algorithms that provide good initial solutions for the meta-heuristics. The computational experiments of the research reveal that the proposed HGA meta-heuristic is superior to the SA algorithm from the viewpoint of solution quality while the SA algorithm requires less computational time.

2.6.3 Ant Colony System

Raghavan and Venkataramana (2006) try to minimize the Total Weighted Tardiness in a parallel batch processor scheduling environment with incompatible job families, identical job sizes and arbitrary job weights. They divide the problem into two sub-problems of Batch Formation and Batch Scheduling and propose an Ant Colony Optimization approach to solve the batch scheduling problem. They apply their proposed algorithm to randomly generated data sets and show that the ant colony optimization based algorithm provides higher quality solutions compared to the well-known Apparent Tardiness Cost-Batched Apparent Tardiness Cost (ATC-BATC) rule.

Mönch and Almeder (2009) propose an Ant Colony System (ACS) approach to solve the same problem. They compare their proposed algorithm to the GA algorithms proposed by Balasubramanian et al. (2004) and suggest that the ACS approach provides higher quality solutions in considerably less computational time.

2.6.4 Neural Networks

Mönch et al. (2006) aim at minimizing total weighted tardiness on parallel batch machines with incompatible job families and unequal release time of the jobs. They develop a heuristic based on the famous Apparent Tardiness Cost (ATC) Dispatching Rule suggested by Vepsalainen and Morton (1987). They use two different machine learning approaches (Neural Networks and Inductive Decision Trees) for choosing the look-ahead parameter which is needed for using the ATC rule. The results on the performance of the machine learning approaches for choosing the look-ahead value indicate their higher solution quality and less computation time. Numerical experiments show that in around eighty percent of the cases, a better total weighted tardiness is obtained from applying machine learning approaches instead of using a fixed look-ahead parameter approach.

2.6.5 Variable Neighborhood Search (VNS)

Klemmt et al. (2009) study the problem of scheduling an unrelated parallel batch machine with unequal release dates and incompatible job families. In this problem which is motivated by a real case in diffusion and oxidation operations in semiconductor industry, only jobs belonging to the same family can be processed simultaneously on a batch processor and since the machines are dedicated to specific product families, not all families can be processed by every machine. They develop a MIP model with the objective of minimizing the total weighted tardiness. They use time window decomposition, machine pool decomposition and job list reduction to get the smaller instances of the problem and solve them. They also propose a Variable Neighborhood Search (VNS) meta-heuristic to solve the problem. The results show that the VNS approach is slightly superior to MIP with regard to solution quality and computation time. However, The MIP formulation is more flexible especially in case of considering additional constraints such as time constraints between consecutive process steps.

Behnamian and Fatemi Ghomi (2011) consider the problem of scheduling a hybrid flow shop with sequence-dependent setup times and resource-dependent processing times. They develop a MIP model with the objective of minimizing make-span and total resource allocation costs. They propose a hybrid meta-heuristic which is built based on a combination of GA and Variable Neighborhood Search (VNS). They test the proposed algorithm on more than 250 benchmark problems with up to 100 jobs. The results show the efficiency of the hybrid meta-heuristic. They also compare their algorithm with the Random Initial Population Simulated Annealing (RNDSA) proposed by Jungwattanakit et al. (2009) and demonstrate the superiority of their algorithm.

2.6.6 Tabu Search

Wang and Tang (2009) study hybrid flow shop scheduling problem with limited intermediate buffers. They aim at minimizing the sum of weighted completion time of all jobs. Since the problem is proven to be strongly NP-hard, they propose a Tabu Search heuristic combined with a modified scatter search with two reference sets in order to obtain near-optimal solutions. They generalize the algorithm of Nawaz et al. (1983) which was originally proposed for pure flow shop sequencing problem so as to obtain good initial solutions for their hybrid flow shop problem. The computational results of their research show that the proposed Tabu Search heuristic outperforms the Lagrangian relaxation algorithm proposed by Tang and Xuan (2006).

2.6.7 Hybrid Particle Swarm Optimization (HPSO) Algorithm

Liu et al. (2010) study a hybrid flow shop scheduling problem with five stages and several parallel batch processing machines in each stage in polypropylene industry. They propose a Hybrid Particle Swarm Optimization (HPSO) algorithm to minimize the make-span. There are different inventory storage policies in different intermediate buffer areas. They use the algorithm of Nawaz et al. (1983) known as Nawaz–Enscore–Ham (NEH) heuristic to generate initial solutions and modify this algorithm to develop a new local search heuristic. They also develop a local search heuristic based on Simulated Annealing approach with an adaptive meta-Lamarckian learning strategy. The computation experiments and simulation results of this research suggest that the proposed HPSO algorithm is more effective and robust compared to the existing hybrid PSO proposed by Tasgetiren et al. (2004) and existing constructive heuristics.

2.6.8 Greedy Randomized Adaptive Search Procedure (GRASP)

Damodaran et al. (2011) study the problem of scheduling several identical batch processing machines in parallel. They propose A Greedy Randomized Adaptive Search Procedure (GRASP) approach in order to minimize the make-span. Their GRASP algorithm obtains optimal solutions for problems with up to 10 jobs and near-optimal solutions for larger instances. The numerical experiments show the high efficiency and superiority of the algorithm compared to previously proposed heuristics in the literature.

2.7 Summary

According to the conducted literature review, there are many general forms to model Hybrid Flow Shop manufacturing systems. The more complex manufacturing environment gets the more complicated and advance the scheduling models become. This is also true for the solution methodologies.

There is an exponentially growing research focus on the subject of flow shops scheduling but only a small proportion of these research attempts are dealing with real-world cases (Reisman et al., 1995). Classical scheduling theory sometimes fails to respond to the needs of practical environments, and the new trends in scheduling research attempt to make it more relevant and applicable to the real world problems (MacCarthy and Liu, 1993). As a result, bridging the gap between the scheduling theory and the practical problems has become a matter of significant importance.

Generally, researchers tend to follow the typical and conventional objectives that are quite well-known in the literature of the scheduling problems. The most popular objective function is Minimizing the make-span, C_{max}, or the completion time of the last job to leave the system. Minimizing other KPIs such as Sum of Completion Times, Total (Weighted) Completion Time, Total (Weighted) Tardiness, Maximum Tardiness and Number of Tardy Jobs are among the other typical objective functions. The following table shows different objective functions considered in the research in the area of Hybrid Flow Shop scheduling. As it can be observed, most of them revolve around the aforementioned objectives.

This is exactly because of the lack of proper connection between the scheduling theory and the practical cases because as one can easily suspect, not all the real manufacturing cases necessarily seek one of these few objectives. The objectives of the real case problems can be very diverse and very different from these typical objectives in the literature of Hybrid Flow Shop scheduling problems. This is one of the points that differentiate this thesis from the previous studies in the area of HFS as its objective is to maximize the utilization of the resources in the cure process which was explained before in section 1.3.

Table 2. Different Typical Objective Functions for HFS Scheduling Problems

No.	Authors and Year	Minimizing Sum of Completion Times	Minimizing Total (Weighted) Completion Time	Minimizing the Make-span	Minimizing Total (Weighted) Tardiness	Minimizing Maximum Tardiness	Minimizing Number of Tardy Jobs	Other Objectives
1	Ahmadi et al. (1992)	✓		✓				
2	Ovacik and Uzsoy (1995)					✓		
3	Sawik (1995)							✓
4	Pinto and Grossmann (1995)							✓
5	Sotskov et al. (1996)	✓		✓				
6	Guinet and Solomon (1996)					✓		
7	Sung and Yoon (1997)			✓				
8	Brucker et al. (1997)		✓		✓	✓	✓	
9	Hoogeveen and Velde (1998)			✓				
10	Danneberg et al. (1999)	✓		✓				
11	Bhatnagar et al. (1999)			✓				
12	Sung et al. (2000)	✓		✓				
13	Sawik (2000)			✓				
14	Blomer and Gunther (2000)			✓				
15	Potts et al. (2001)			✓				
16	Sung and Min (2001)				✓	✓		
17	Wang et al. (2001)			✓				
18	Méndez et al. (2001)							✓
19	Kim and Kim (2002)		✓					
20	Wang and Li (2002)			✓				
21	Sawik (2002)			✓				
22	Su (2003)			✓				
23	Sung and Kim (2003)				✓	✓	✓	
24	Damodaran and Srihari (2004)			✓				
25	Mathirajan et al. (2004)			✓				

No.	Authors and Year	Minimizing Sum of Completion Times	Minimizing Total (Weighted) Completion Time	Minimizing the Make-span	Minimizing Total (Weighted) Tardiness	Minimizing Maximum Tardiness	Minimizing Number of Tardy Jobs	Other Objectives
26	Balasubramanian et al. (2004)				✓			
27	Lin and Cheng (2005)			✓				
28	Low (2005)						✓	
29	Sawik (2005)				✓	✓	✓	
30	Mönch et al. (2006)				✓			
31	Reichelt and Mönch (2006)			✓	✓			
32	Raghavan and Venkataramana (2006)				✓			
33	Sawik (2006)				✓			
34	Voss and Witt (2007)				✓			
35	Xuan and Tang (2007)		✓					
36	Li et al. (2008)				✓			
37	Ruiz et al. (2008)			✓				
38	Klemmt and Hielscher (2008)			✓				
49	Amin-Naseri and Beheshti-Nia (2009)			✓				
40	Luo et al. (2009)			✓				
41	Klemmt et al. (2009)				✓			
42	Mönch and Almeder (2009)				✓			
43	Bellanger and Oulamara (2009)			✓				
44	Oulamara et al. (2009)			✓				
45	Wang and Tang (2009)	✓						
46	Gong et al. (2010)	✓		✓				
47	Liu et al. (2010)			✓				
48	Damodaran et al. (2011)			✓				
49	Behnamian and Fatemi Ghomi (2011)			✓				

No.	Authors and Year	Minimizing Sum of Completion Times	Minimizing Total (Weighted) Completion Time	Minimizing the Make-span	Minimizing Total (Weighted) Tardiness	Minimizing Maximum Tardiness	Minimizing Number of Tardy Jobs	Other Objectives
50	Luo et al. (2011)			✓				
51	Klemmt and Mönch (2012)				✓			
52	Gicquel et al. (2012)				✓			
53	Wang et al. (2012)			✓				
54	Ziaeifar et al. (2012)			✓				
55	Rossi et al. (2013)			✓			✓	
56	Rossi et al. (2014)			✓			✓	

Although there are many attempts to model the complex HFS in the literature, to the best of our knowledge, the constraint of physical capacity of the buffers is something missing. Some researchers (Leisten, 1990; Agnetis et al., 1997; Pranzo, 2004; Wang et al., 2006; Liu et al., 2008) have attempted to study the scheduling problems with limited buffers but they did not consider the physical capacity (volume) of the buffer as a constraint which we try to cover in this research.

The limited number of tools is also a constraint which has not been studied deeply in the literature. Unlike the constraints about the limited number of resources (machines) in different stages, limited number of tools does not seem to be seriously taken into account in the theory of scheduling problems. However, in reality, it is a very important constraint that can totally affect the performance of a hybrid flow shop manufacturing system. In this research, we aim at including this constraint into the modeling and formulations.

3 Mathematical Model Formulation and Analysis

In this section, we will try to develop an optimization model for the case of the Composite Manufacturing Center. In the following, first we will describe the case study setting. We then define the problem and then, the parameters, variables and the equations which are needed to model it. Finally, we will discuss the solution methodology and the results we have obtained from CPLEX.

3.1 Case Study Characteristics

The Composite Manufacturing Center (CMC) of the case company of this research (ABC Company) is producing hundreds of composite parts both for the assembly line and for the spares usage. The center consists of different processes:

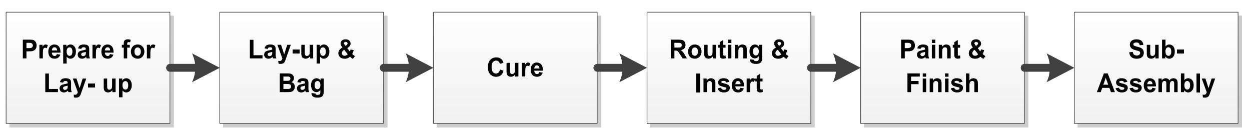

Figure 1. Various Processes inside the Composite Manufacturing Center

Currently, CMC does not fail to meet the demand of its customers apparently because the excess capacity in the system is absorbing the variations. However, the demand is going to increase in the future and the need for providing visibility on the capacity of the system in general, and the capacity of the lay-up and curing processes in particular, is seriously felt.

One of the most important factors to boost the efficiency of Composite Manufacturing Systems successfully is visibility. Visibility enables the supply chain participants to have access to reliable information regarding inventory, product availability, order status, etc. and helps to manage the flow of products and services more effectively. Hence, the benefit of providing this visibility is to have reliability throughout the chain (Myerson, 2012).

To provide such a visibility on a system (i.e. a supply chain), VSM is proven to be a very powerful technique. There are many opportunities to look for waste and non-value-added activities in composite manufacturing systems and eliminate them by the aid of VSM.

VSM is a team-based technique of depicting the process from the beginning to the end (current state map) and covering both value-added and non-value-added activities from the customer's perspective, numerical and visual data regarding the performance of these activities and then, eliminating the wastes, defects, failures and non-value-added activities to acquire the future state map. This will enable the company to better prioritize the improvement efforts, provide a common language for people involved, and prepare the ground for an implementation plan (Liker, 2005).

VSM is a very useful technique to reveal the areas of potential waste in supply chains and networks. Supplier performance and supplier lead times, distribution network's efficiency, and transfer facilities are different areas to look for waste. Thus, in order to better explain the circumstances of the Composite Manufacturing Center at the ABC Company, we employ this technique.

In the following page, the Value Stream Map of the Composite Manufacturing Center at ABC Company is depicted. There is a Production Control department over CMC that receives 3-5 years forecast of demand from the customers and sends an 18-24-month forecast of CMC's needs to the suppliers.

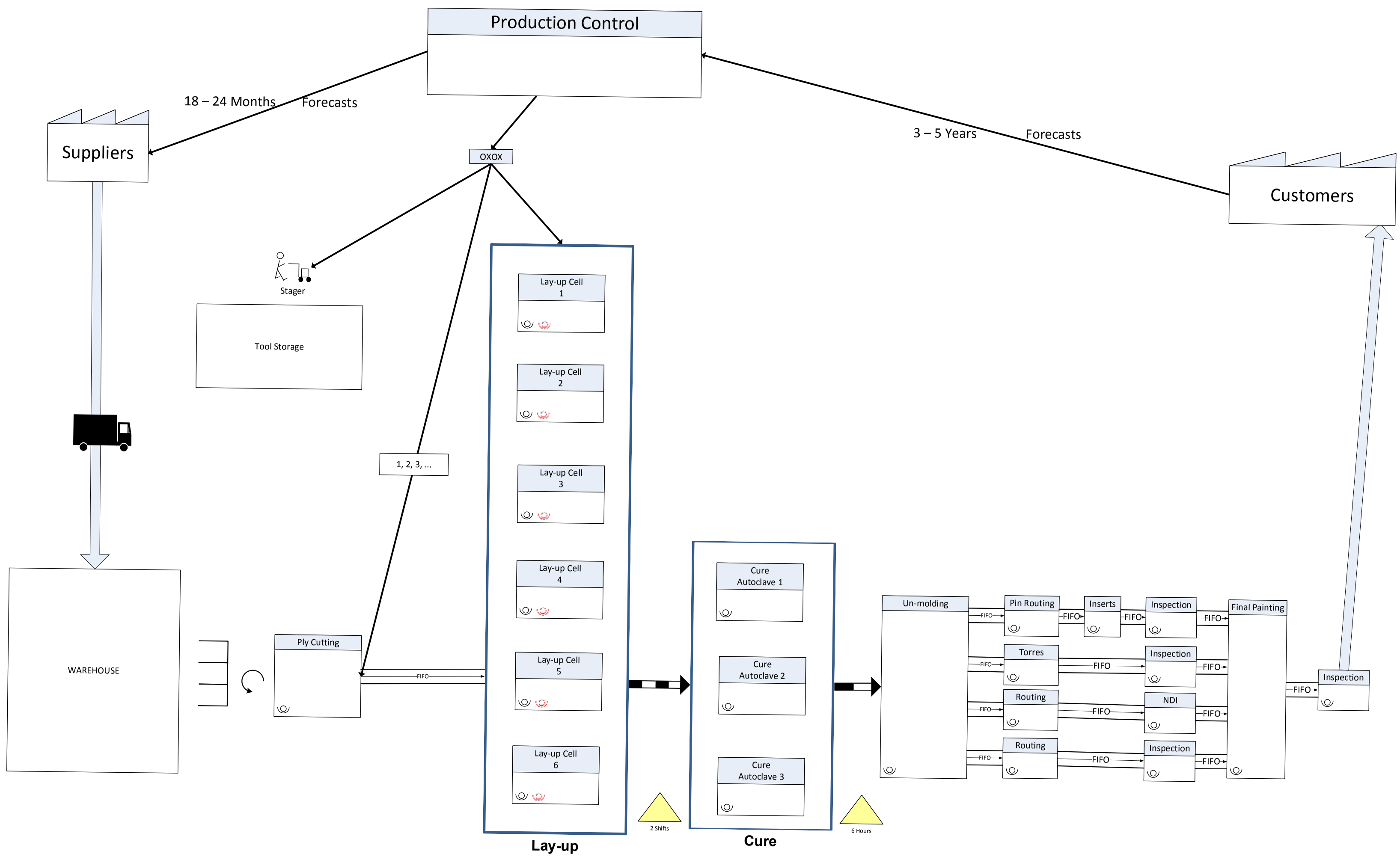

Figure 2. Value Stream Map of the Composite Manufacturing Center

In the Ply-cutting process, the sheets of composite are cut in the required sizes and shapes. Then, they are sent to the Lay-up process. In the Lay-up process, there are six cells for different product families. Inside each cell, a number of regular and laser-equipped work stations exist. Each work station (regular or laser-equipped) is managed by one Lay-up Operator.

The operators of the lay-up cells stack the sheets of composite on tools (molds) and vacuum the mold so that the part can be ready for the cure process. The aforementioned tools or molds are brought in the cell by a person whose responsibility is to receive the production schedule and bring the required mold to the lay-up cell exactly on time. This person is called Stager in this Composite Manufacturing Center.

After being processed in the lay-up cell, parts have to wait in a buffer area between the lay-up process and the cure process. This buffer is where the parts are grouped in batches. It is worth mentioning that this buffer has a limited physical capacity (in terms of volume). Therefore, at any given time, only a certain volume of jobs can be present in the buffer area. There is also a time limitation in this buffer area. Due to some technical reasons, parts cannot remain attached to their tool (mold) and stay in this buffer area for more than two working shifts.

When the required parts for creating a batch are present in the buffer and the batch is complete, parts which are compatible from the viewpoint of the cure recipe are grouped and are sent to the autoclave.

The cure process accommodates various cure recipes for different products. There are three autoclaves which are capable of running different cure recipes. The schedule of the different cure recipes are fixed for two of the autoclaves and the third one is allocated for custom orders.

After the parts are cured in one of the autoclaves, they are sent to the unmolding process so that the operators in this work station separate the parts from their molds. As the next step, the mold will be sent to a tool preparation process and will be cleaned and prepared to be used again. The cured part is sent to the next process which is linked to the unmolding process with a FIFO lane.

All the processes downstream of the unmolding process are linked together with FIFO lanes. Hence, downstream of the unmolding process, continuous flow of the materials is achieved.

However, certain conditions of the processes upstream of the unmolding process make the value stream quite complicated.

The production requirements are sent to three points. The first and foremost process that receives the production requirements is the lay-up process. The concept of Offset Scheduling is used to offset the production requirements by two days and send them to the ply-cutting process so that they can cut the required sheets for the lay-up process in advance and make sure they are ready when needed. The third point that receives the production requirements is the Stager. He should know about the production schedule so that he can bring the required molds to the lay-up cells exactly on time.

Based on this procedure, one may conclude that the pacemaker of this value stream is the lay-up process which is a valid statement to some extent. However, certain conditions of the composite manufacturing center add some complexities to the problem and make the task of identifying the pacemaker more challenging.

One of the most significant constraints of this composite manufacturing center is the limited number of molds which are used in the lay-up, cure and unmolding processes. This constraint is a common limitation of every composite manufacturing system because molds are expensive to build and they are unique for each product type. This limitation can be explained better by introducing the concept of Tooling Cycle in the composite manufacturing center.

The Tooling Cycle is the path that each mold goes through every time it is used to produce a part. It starts from the Tool Storage area when the Stager takes the tool and brings it to the lay-up cell. After being processed in the lay-up cell, the mold and the attached carbon fiber sheets depart from the lay-up cell and go to the autoclave. After the cure process, the part will be separated from its mold and the mold will be free to go to the Tool Preparation area. After being cleaned and prepared, the mold is sent to the Tool Storage and the Tooling Cycle terminates. The red path in Figure 3 depicts this cycle.

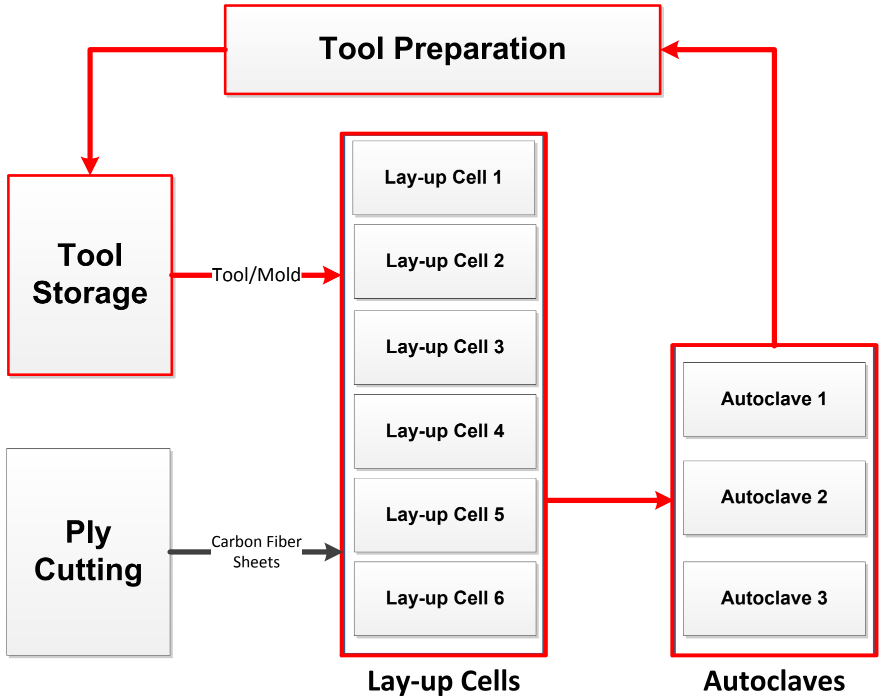

Figure 3. The Tooling Cycle

As long as a mold is within its tooling cycle, it cannot be used for any other product. This puts a constraint on scheduling the jobs in this value stream because only the jobs with different mold requirements can be scheduled at any given time.

As a result, one should look at the system (which is composed of the lay-up cells, cure process, unmolding process and tool preparation process) as a whole. This system (instead of a single process) is where the jobs need to be scheduled in this composite manufacturing center. Therefore, the task of scheduling the jobs within this system becomes the most significant challenge of this research.

This system in the literature of scheduling problems can be characterized as a two-stage Hybrid Flow Shop. This HFS is a common manufacturing environment in composite manufacturing systems of aerospace companies. Therefore, the proposed scheduling systems in this research can be applied to aerospace composite manufacturing systems.

3.2 Scheduling Hybrid Flow Lines of Aerospace Composite Manufacturing Systems

First, we need to define the main problem of this research. The VSM in Figure 4 shows how the work is done inside the CMC.

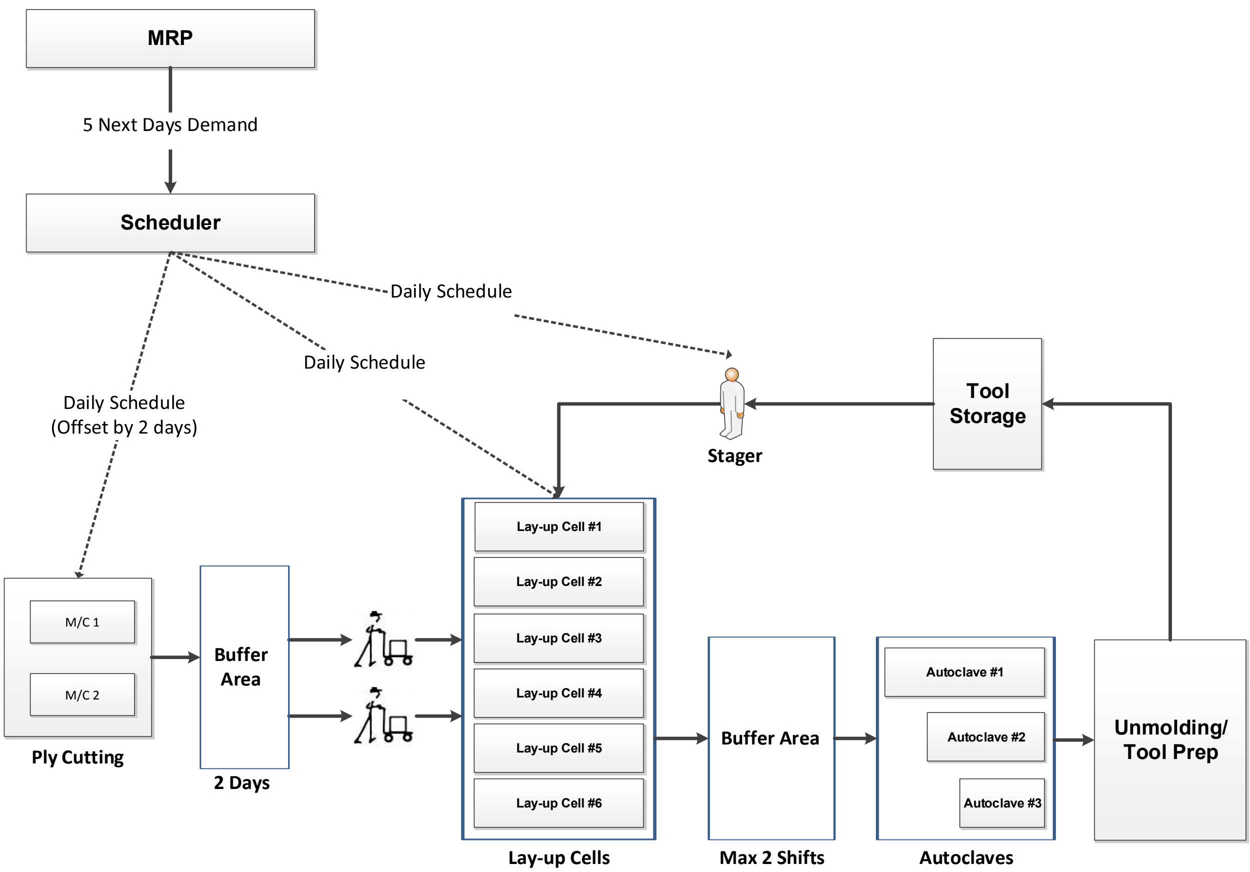

Figure 4. Partial Value Stream Map of the CMC

Every day, the scheduler looks at the MRP requirements for the next 5 days (in case of custom orders for the next 15 days). Then, he plans the schedule of the day for the available manpower. Each required unit (job) has a due date specified by the MRP system.

To schedule the jobs, the scheduler has to consider these points:

- The lay-up process is composed of a number of unrelated cells, each one of which performs a different process. The parts which are sent to each one of the cells are from different product families. For example, one cell is allocated only for processing the wings while the other one is for doors. However, inside each cell, there are a number of work stations which are either regular or laser-equipped. Each one of these two types can include a number of identical work stations
- The number of regular and/or laser-equipped work stations within each lay-up cell is limited.
- The buffer area between the lay-up process and the cure process cannot keep any part for more than a specific time period because carbon fiber sheets cannot get attached to the molds for more than a certain amount of time.
- The buffer also has a limited physical capacity. Therefore, at any given time, no more than a certain volume of jobs can exist in the buffer.
- Each type of tool is unique and there are normally only a very limited number of tools for each part. Therefore, it's not possible to schedule the jobs which need the same tool simultaneously or in a row.
- Different parts need to be grouped before being sent to the autoclaves for the cure process. Therefore, forming the optimal batches is one of the requirements that the scheduling system should satisfy.
- The autoclaves' operating plan is fixed. Each type of cure has its own fixed schedule.
- The physical capacities (volume) of the autoclaves are limited.

A buffer of two days is kept between the ply cutting and the lay-up cells. After lay-up, products are batched according to their cure types and then they are sent to the autoclaves. The autoclaves #1 and #2 are used for the 4 main types of curing and they cover around 95 percent of the demand. Autoclave #3, which is the small one, is used for the custom orders.

After curing, the tools are unmolded and then prepared to be sent to the tooling storage, this is where the tooling cycle terminates. The parts are then sent to the next processes in a continuous flow manner (with FIFO lanes).

If we look at this as a classic scheduling problem, we have a scheduling system which is characterized as a Hybrid Flow Shop composed of the lay-up cells, curing process and the buffer between them. The orders are sent to this system and the parts are processed inside it, and then they go through all the processes downstream which are working in continuous flow with FIFO lanes (Figure 2).

3.3 Developing the Mixed Integer Linear Programming Model

First, the scope of the model should be explained. As mentioned before, a buffer of two days is kept between the Ply Cutting and the Lay-up cells. The idea of putting a supermarket (size=2 days) seems logical and it is proven to be effective in the current system. Therefore, we keep this pull system as it is and do not include Ply Cutting in the optimization model.

The model will include lay-up cells, curing process, the buffer between lay-up and cure, and the unmolding process (Figure 5).

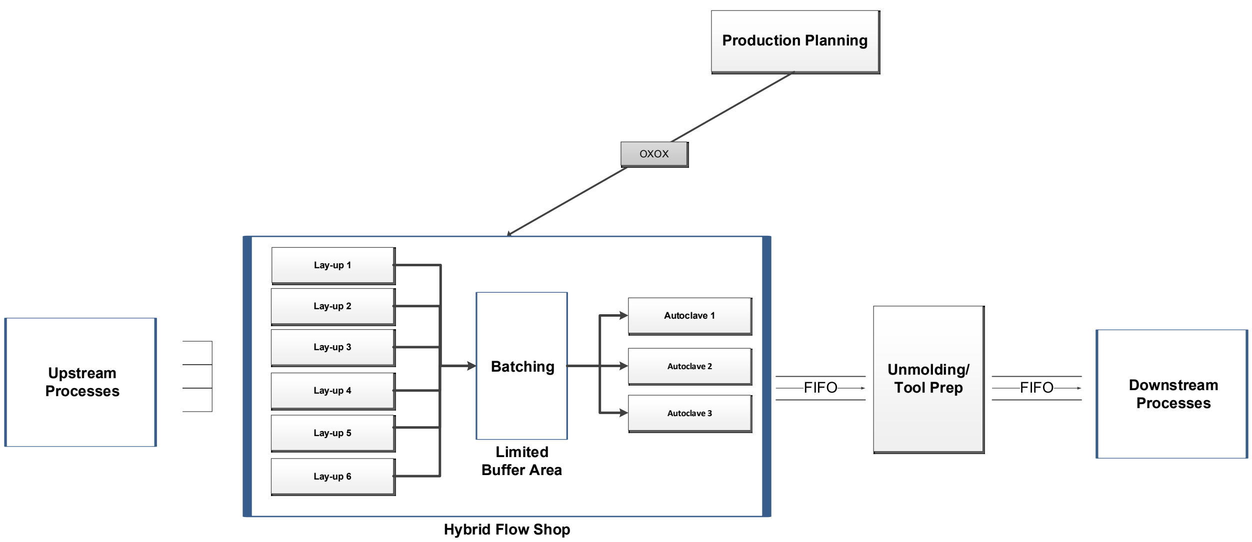

Figure 5. Forming a scheduling system composed of lay-up cells, curing process and the buffer between them

Each of the stages in Figure 5 has its own limitations. In the following, the indices, parameters and variables needed to model these stages are introduced and then the development of the MILP model for each stage is explained step by step.

3.3.1 Indices

i = 1, 2, 3, …, I	jobs that are released to be scheduled
t = 1, 2, 3, …, 480	time slots of 15 minutes during the next 5 days
lu = 1, 2, 3, …, 6	The number assigned to each lay-up cell
c = 1, 2, 3, …, 25	The number assigned to each cure operation during the next 5 days

3.3.2 Input Parameters

I (lu)	Set of jobs that need to be processed in lay-up cell lu (lu=1, 2, …, 6)
OperatorCap (lu)	The manpower capacity of each cell lu (lu=1, 2, …, 6)
LaserCap (lu)	The laser capacity of each cell lu (lu=1, 2, …, 6)
Laser (i)	1 when job i needs a laser station to be processed and 0 when it doesn't
ProcessingTime (i)	The time job (i) takes to be processed in one of the lay-up work stations
Cure (c)	The cure operation number (c) in the next five days
StartCure (c)	The time cure number (c) starts
DurationCure (c)	The time takes for Cure(c) to be completed
SizeMold (i)	The dimensions of the mold which is used for job (i), i=1, 2, 3, …, J
AutoclaveCap (c)	The physical capacity of the autoclave that is operating cure number (c)
CureProfile (i) = 1,2,….,5	Type of cure recipe that job (i) needs
CureType (c) = 1,2,….,5	Type of the cure recipe that cure number (c) provides
BufferTime	Maximum time a job can stay in the intermediate buffer
M (i,j)	1 if the job (i) needs the mold (j). Otherwise it's zero.
K (c,t)	1 when the cure number (c) is active, otherwise it's zero.
T (j)	Number of mold (j) available in the system
M	A relatively big number

3.3.3 Decision Variables

S (i,t)	Start time of job (i), it's 1 when job (i) is started at time (t), otherwise it's zero.
X (i,t)	Whether or not job (i) is being processed at time (t) in one of the lay-up cells. If X (i,t) is 1 it means that job (i) is being processed at time (t). Otherwise it's zero.
Y (i,c)	1 when job (i) is assigned to Cure (c). Otherwise it's zero.
Z (i,t)	1 when job (i) is in the buffer during time (t) and it's zero when job (i) is not in the buffer.

3.3.4 Objective Function

The objective function of this model is to maximize the utilization of the resources in the second stage (autoclaves). From a technical point of view, we try to maximize the number of products that are sent to each cure treatment. To do so, we have developed the following objective function.

$$\text{Min} \sum_{i\,\in\,I} \sum_{c=1}^{25} Y(i,c) * c$$

This objective function tries to minimize the value of *Y (i,c) * c*. *Y (i,c)* is a binary variable which is equal to 1 when job (i) is assigned to *Cure (c)* and it is 0 otherwise. This value is multiplied by c which is a number assigned to each cure operation during the next 5 days (c = 1, 2, 3, …, 25). Therefore, this objective function strives to push the jobs toward the smaller amounts of c which in turn means the earliest cure treatments.

In other words, the objective function is designed in a way that maximizes the number of jobs which can be assigned to *Cure (1)*. Then it goes to *Cure (2)* and tries to fill up the autoclave as much as possible. Then it does the same thing for *Cure (3)* and so on. This process continues until the last cure which is Cure (5) is filled. Thus, normally, the first days cures will be utilized as much as possible and the capacity of the last days cures will be reserved as much as possible which is very useful in case of facing increased demand.

3.3.5 Constraints

In this section, the constraints of each part of the problem (two main stages, the intermediate buffer and the molding constraint) will be presented in detail.

3.3.5.1 Lay-up Cells Constraints

As we are planning to develop a time-indexed mixed integer linear programming model, first we need to discretize time. We know that currently the scheduler looks 5 days ahead and decides about the production requirements. To keep it simple, we assume the planning horizon is 5 days and we divide this time to 480 to get time slots of 15 minutes. Therefore, t=1, 2, 3, …, 480 shows the time periods. According to the data we have, rounding the processing times of different products to fit them into slots of 15 minutes is a reasonable assumption.

Every day, we have a set of jobs with different due dates but we know they should be started within the next five days. So, i=1, 2, 3, …, J show the jobs that are released to be scheduled.

Each job has its own lay-up process, so, we can put the jobs into 6 different sets each one showing the available jobs for each lay-up cell.

I (1), *I (2)*, …, *I (6)* show the sets of jobs that need to be processed in lay-up cell 1, 2, …, 6 respectively. We show them as *I (lu)*,

Each lay-up cell has a number of work stations. The number of work stations (operators) shows the capacity of that lay-up cell. We define a parameter named *OperatorCap (lu)* (lu=1, 2, …, 6) to show the manpower capacity of each cell.

Each lay-up cell has a number of laser work stations. We define a parameter named *LaserCap (lu)* (lu=1, 2, …, 6) to show the laser capacity of each cell.

Another parameter which should be defined is *Laser (i)*. *Laser (i)* is 1 when job i needs a laser-equipped station to be processed and it is 0 when it needs a regular laser station.

ProcessingTime (i) is another parameter that shows the time job (i) takes to be processed in one of the lay-up work stations.

S (i,t) is a variable which represents the start time of job (i). If *S (i,t)* is 1 it means that job (i) is started on time (t). Otherwise it's zero.

X (i,t) is a variable which determines whether or not job (i) is being processed in time (t) in one of the lay-up cells. If *X (i,t)* is 1 it means that job (i) is being processed in time (t). Otherwise it's zero.

So, the first set of constraints for the lay-up cells can be defined as:

$$\sum_{i \in I(lu)} X\,(i,t) \;\leq OperatorCap(lu)$$

$$\forall\, t = 1,2,\dots,480 \;and\; lu = 1,2,\dots,6 \qquad (1)$$

$$\textstyle\sum_{i \in I(lu)} X\,(i,t) * Laser\,(i) \leq LaserCap(lu)$$

$$\forall\, t = 1,2,\dots,480 \;and\; lu = 1,2,\dots,6 \qquad (2)$$

$$ProcessingTime(i) * S(i,t) \leq \textstyle\sum_{r=t}^{t+ProcessingTime(i)-1} \; X(i,t)$$

$$\forall\, i \in I(lu) \;and\; lu = 1,2,\dots,6 \quad t = 1\,,2,\dots,480 - ProcessingTime(i) \qquad (3)$$

$$\sum_{t=1}^{480} X(i,t) = ProcessingTime(i)$$

$$\forall\, i \;\in\; I(lu) \;and\; lu = 1,2,\dots,6 \qquad (4)$$

$$\textstyle\sum_{t=1}^{480} S(i,t) = 1 \qquad \forall\, i \;\in\; I(lu) \;and\; lu = 1,2,\dots,6 \qquad (5)$$

Equation (1) ensures that we do not exceed the capacity of lay-up work stations. Equation (2) ensures that we do not exceed the laser capacity. Equation (3) and (4) ensure that when a job starts, the process will continue until its lay-up process is completed. In other words, these sets of equations prohibit pre-emption in the lay-up processes. Equation (5) states that a job can start at only one point in time. It also ensures that the job will be started within the next five days, so, we will certainly meet the due date of the job.

3.3.5.2 *Cure Process Constraints*

The next step of the modeling is focused on the curing process. We assume the schedule of different cure types is fixed. It means during the next five days we will have 25 cure operations with known start and finish times (Figure 6). To maintain the simplicity of the model for now, we just look at autoclave #1 and #2 which cover 95% of the products.

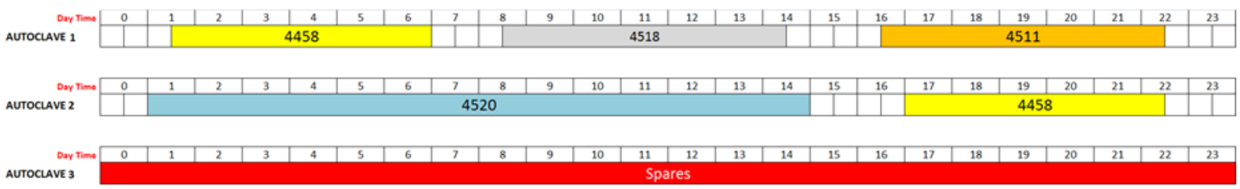

Figure 6. The fixed Schedule of Autoclaves in the Cure Process

We define the *Cure (c)* in order to assign a number to each of these cure operations. So, *Cure (1)* & *Cure (2)* & *Cure (3)* indicate the cure operations on Autoclave #1 in the first day, *Cure (4)* and *Cure (5)* indicate the cure operations on Autoclave #2 in the first day, …, *Cure (25)* indicates the last cure operation in Autoclave #2 in the fifth day.

We define a parameter named *StartCure (c)* as the time *Cure (c)* starts and another parameter named *DurationCure (c)* as the time *Cure (c)* takes to be completed. *SizeMold (i)* is another parameter that specifies the dimensions of the mold which is used for job (i). *AutoclaveCap (c)* is also defined as the capacity of the autoclave that is operating cure number (c).

We know we have 5 different types of cure. So, we define another parameter named *CureType (i)* that specifies which type of cure each job needs. *CureType (c)* also shows which type of cure each of the 25 cure operations provides. *CureType (i)* / *CureType (c)* can get the values 1, 2, 3, 4, or 5.

We define *Y (i,c)* as a variable which is equal to 1 when job (i) is assigned to *Cure (c)*. Otherwise it's zero.

$$Y(i,c) * CureProfile(i) = Y(i,c) * CureType(c)$$

$$\forall\, i = 1, 2, \dots, I \text{ and } c = 1, 2, \dots, 25 \tag{6}$$

$$\textstyle\sum_{c=1}^{25} Y(i,c) = 1 \qquad \forall\, i = 1, 2, \dots, I \tag{7}$$

$$\sum_{i=1}^{J} Y\ (i,c) * \ SizeMold(i) \ \leq AutoclaveCap(c) \qquad \forall \ c = 1, 2, \ldots, 25 \tag{8}$$

Equation (6) suggests that jobs cannot be assigned to the cure operations which are not the same type. Equation (7) states that a job can be assigned only to one of the cure operations. Equation (8) ensures that we do not exceed the capacity of the autoclaves.

3.3.5.3 The Constraints of the Buffer between the Lay-up Cells and the Cure Process

There are two types of constraints for the buffer. The first one is the time limitation; we know that we cannot keep a part in the buffer for more than two shifts (Maximum Buffer Time).

$$\mathrm{StartCure(c)} * \mathrm{Y(i,c)} - \sum_{\mathrm{t=1}}^{480} \mathrm{t} * \ \mathrm{S(i,t)} - \mathrm{ProcessingTime(i)} \geq (\mathrm{Y(i,c)} - 1) * M$$

$$\forall \ i = 1, 2, \ldots, I \qquad \text{and} \ \ c = 1, 2, \ldots, 25 \tag{9}$$

$$\mathrm{StartCure(c)} * \mathrm{Y(i,c)} - \sum_{\mathrm{t=1}}^{480} \mathrm{t} * \ \mathrm{S(i,t)} - \mathrm{ProcessingTime(i)}$$
$$\leq Maximum\ Buffer\ Time + \big(1 - \mathrm{Y(i,c)}\big) * M$$

$$\forall \ i = 1, 2, \ldots, I \qquad \text{and} \ \ c = 1, 2, \ldots, 25 \tag{10}$$

The buffer time cannot be negative (equation 9). Equation (10) ensures that the time limit of the buffer is not exceeded.

The second constraint of the buffer is its physical capacity. We define the variable *Z (i,t)* to model this part of the problem. *Z (i,t)* equals to 1 when job (i) is in the buffer during time (t) and it's zero when job (i) is not in the buffer.

We divide the whole journey of a job from beginning to the end of the tooling cycle to five periods.

i. Before the job is processed in one of the lay-up cells:

$$Z(i,t') \leq 1 - \sum_{t=t'}^{480} S(i,t) \qquad \forall \ i = 1, 2, \ldots, J \ \ and \ \ t' = 1, 2, \ldots, 480 \tag{11}$$

Equation (11) states that job (i) cannot be in the buffer before its processing in one of the lay-up cells is started.

ii. While the job is being processed in one of the lay-up cells:

$$Z(i,t) \leq 1 - X(i,t) \qquad \forall\, i = 1,2,\dots,I \;\; and \;\; t = 1,2,\dots,480 \tag{12}$$

Equation (12) states that job (i) cannot be in the buffer during the time it's being processed in one of the lay-up cells.

iii. While the job is in the buffer between lay-up cells and autoclaves:

$$\sum_{t=1}^{480} Z(i,t) = StartCure(c) * Y(i,c) - \sum_{t=1}^{480} t * S(i,t) - ProcessingTime(i) \tag{13}$$

Equation (13) states that the job is in the buffer when its lay-up process is finished until its cure process starts.

iv. While the job is being cured:

$$\sum_{t=StartCure(c)}^{StartCure(c)+DurationCure(c)} Z(i,t) \leq \big(1 - Y(i,c)\big) * DurationCure(c)$$

$$\forall\, i = 1,2,\dots,I \;\; and \;\; c = 1,2,\dots,25 \tag{14}$$

Equation (14) states that job (i) cannot be in the buffer during the time it's being cured.

v. After the job is cured:

$$\sum_{t=StartCure(c)+DurationCure(c)}^{480} Z(i,t) \leq \big(1 - Y(i,c)\big) * 480$$

$$\forall\, i = 1,2,\dots,I \;\; and \;\; c = 1,2,\dots,25 \tag{15}$$

Equation (15) states that job (i) cannot be in the buffer after it is cured.

Now, we are able to use Z(i,t) in order to define the constraint of physical capacity of the buffer:

$$\sum_{i=1}^{I} Z\,(i,t) * SizeMold(i) \leq Buffer\ Capacity \qquad \forall\;\; t = 1,2,\dots,480 \tag{16}$$

3.3.5.4 The Molding Constraint

In order to model this part of the problem, we need to define a parameter *M (i,j)* which is equal to 1 if the job (i) needs the mold (j). Otherwise it's zero. *T (j)* is another parameter that specifies the number of available molds of type (j).

We also need to define a parameter named *K (c,t)*. *K (c,t)* is equal to 1 when the cure number (c) is active, otherwise it's zero. This parameter also includes the unmolding part after each cure. For example, if the cure starts at time 100 and it takes 6 hours (24*15 minutes) and the unmolding after cure takes 4 hours (16*15 minutes), K(c,t) will be 1 for t=100 to 140 and it is 0 for t<100 and t>140.

Having *X (i,t)*, *Z (i,t)*, *Y (i,c)* and *K (c,t)*, we can easily track each product and see whether or not it is attached to its specific mold. The equation below insures that at any given time we do not exceed the number of available molds for different part types.

$$\sum_{i \in I(lu)} M(i,j) * X(i,t) + \sum_{i \in I(lu)} M(i,j) * Z(i,t) + \sum_{i \in I(lu)} M(i,j) * Y(i,c) * K(c,t) \le T(j)$$

$$\forall\, t = 1, 2, \dots, 480 \text{ and } lu = 1, 2, \dots, 6 \qquad (17)$$

3.3.6 The MILP Model

Objective Function

$$\text{Min} \sum_{i \in I} \sum_{c=1}^{25} Y(i,c)*c$$

Subject to

$$\sum_{i \in I(lu)} X(i,t) \le \text{OperatorCap}(lu) \qquad \forall\, t=1, 2, \dots, 480 \text{ and } lu=1, 2, \dots, n \qquad (1)$$

$$\sum_{i \in I(lu)} X(i,t) * \text{Laser}(i) \le \text{LaserCap}(lu) \qquad \forall\, t=1, 2, \dots, 480 \text{ and } lu=1, 2, \dots, n \qquad (2)$$

$$\text{ProcessingTime}(i)*S(i,t) \le \sum_{r=t}^{t+\text{ProcessingTime}(i)-1} X(i,t) \qquad (3)$$

$$\forall\, i \in I(lu) \text{ and } lu=1, 2, \dots, 6 \quad t=1, 2, \dots, 480-\text{ProcessingTime}(i)$$

$$\sum_{t=1}^{480} X(i,t) = \text{ProcessingTime}(i) \qquad \forall\, i \in I(lu) \text{ and } lu=1, 2, \dots, n \qquad (4)$$

$$\sum_{t=1}^{480} S(i,t) = 1 \qquad \forall\ i \in I(lu) \text{ and } lu=1, 2, \ldots, n \qquad (5)$$

$$Y(i,c)*CureProfile(i)=Y(i,c)*CureType(c) \qquad \forall\ i=1, 2, \ldots, I \text{ and } c=1, 2, \ldots, k \qquad (6)$$

$$\sum_{c=1}^{25} Y(i,c) = 1 \qquad \forall\ i=1, 2, \ldots, I \qquad (7)$$

$$\sum_{i=1}^{J} Y\ (i,c)*\ SizeMold(i) \leq AutoclaveCap(c) \qquad \forall\ c=1, 2, \ldots, k \qquad (8)$$

$$StartCure(c)*Y(i,c)-\sum_{t=1}^{480} t*\ S(i,t)-ProcessingTime(i) \geq \left(Y(i,c)-1\right)*M$$

$$\forall\ i=1, 2, \ldots, I \text{ and } c=1, 2, \ldots, k \qquad (9)$$

$$StartCure(c)*Y(i,c)-\sum_{t=1}^{480} t*\ S(i,t)-ProcessingTime(i) \leq \text{Maximum Buffer Time}+\left(1-Y(i,c)\right)*M$$

$$\forall\ i=1, 2, \ldots, I \text{ and } c=1, 2, \ldots, k \qquad (10)$$

$$Z(i,t') \leq 1-\sum_{t=t'}^{480} S(i,t) \qquad \forall\ i=1, 2, \ldots, I \text{ and } t'=1, 2, \ldots, 480 \qquad (11)$$

$$Z(i,t) \leq 1-X(i,t) \qquad \forall\ i=1, 2, \ldots, I \text{ and } t=1, 2, \ldots, 480 \qquad (12)$$

$$\sum_{t=1}^{480} Z(i,t) = StartCure(c)*Y(i,c)-\sum_{t=1}^{480} t*\ S(i,t)-ProcessingTime(i)$$

$$\forall\ i=1, 2, \ldots, I \qquad (13)$$

$$\sum_{t=StartCure(c)}^{StartCure(c)+DurationCure(c)} Z(i,t) \leq \left(1-Y(i,c)\right)*DurationCure(c)$$

$$\forall\ i=1, 2, \ldots, I \text{ and } c=1, 2, \ldots, k \qquad (14)$$

$$\sum_{t=StartCure(c)+DurationCure(c)}^{480} Z(i,t) \leq \left(1-Y(i,c)\right)*480 \qquad \forall\ i=1, 2, \ldots, I \text{ and } c=1, 2, \ldots, k \qquad (15)$$

$$\sum_{i=1}^{I} Z\ (i,t)*\ SizeMold(i) \leq \text{Buffer Capacity} \qquad \forall\ t=1, 2, \ldots, 480 \qquad (16)$$

$$\sum_{i \in I(lu)} M\ (i,j)*X(i,t)+\sum_{i \in I(lu)} M\ (i,j)*Z(i,t)+\sum_{i \in I(lu)} M\ (i,j)*Y(i,c)*K(c,t) \leq T(j)$$

$$\forall\ t=1, 2, \ldots, 480 \text{ and } lu=1, 2, \ldots, n \qquad (17)$$

3.4 Solving the MILP Model and Discussion on the Results

In order to solve the developed MILP model, we have coded it in CPLEX and linked it to a number of Microsoft Excel Worksheets which include the data we have received from the Composite Manufacturing Center of the ABC Company.

The results from running the model demonstrate that CPLEX with an underlying branch and bound algorithm is capable of solving the MILP model perfectly for small and medium-size problems. The problems with up to 50 jobs are optimally solved in less than five minutes. The problems with up to 70 jobs can be also optimally solved in less than 1 hour. After obtaining the results from CPLEX, a Gantt chart is drawn to visualize the start time and finish time of the jobs in different stages. Figure 7 shows an example of a problem with 70 jobs which is solved in CPLEX within 42 minutes.

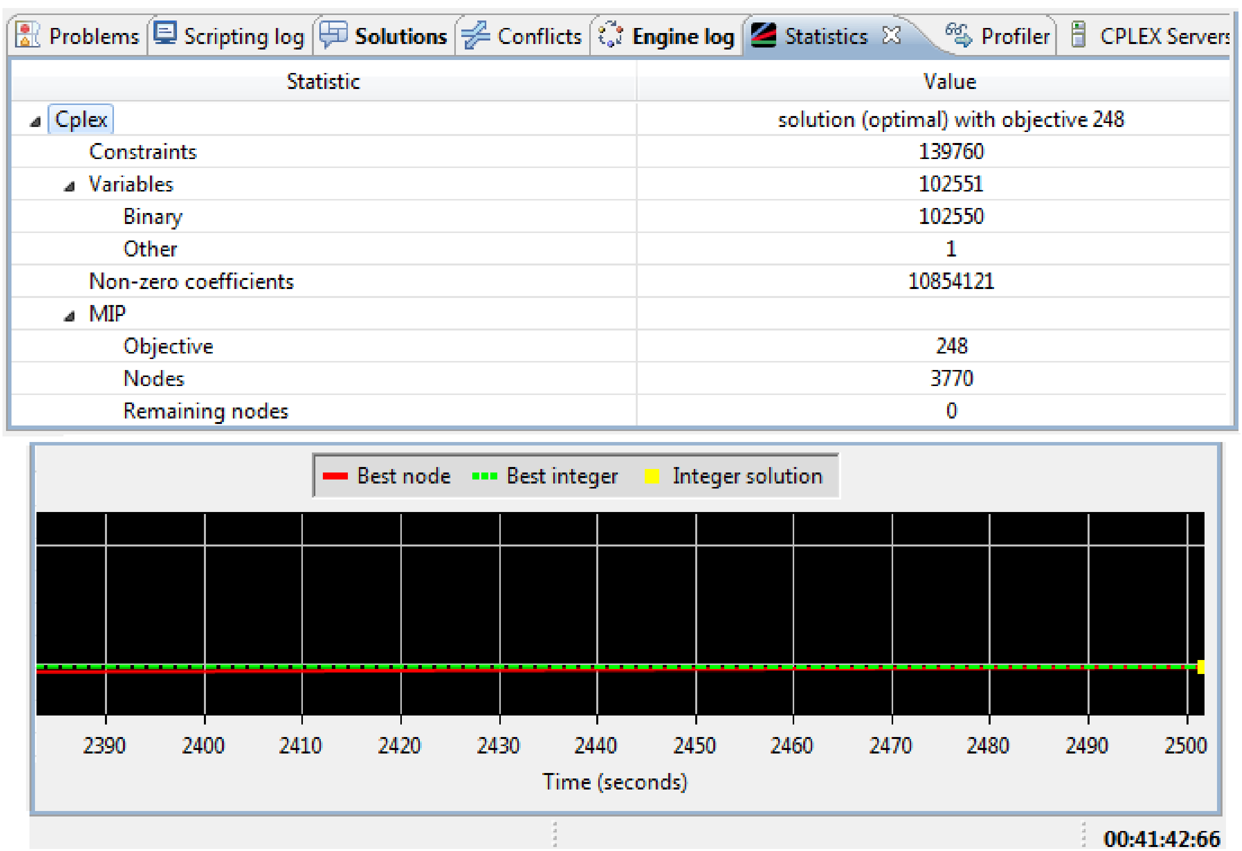

Figure 7. An example of MILP model results in CPLEX

However, for the larger instances of the problem, CPLEX achieves the optimal solutions in relatively long CPU run times. For example, a problem with a set of 100 jobs was solved optimally in 24 hours. The reason for such a long run time of CPLEX for large instances of the problem is the huge number of constraints and variables. For the problem with 100 jobs, we had 210,660 constraints and 146,501 variables. This high number of constraints and variables is partly due to the nature of the modeling approach we used in this thesis which is a discrete-time MILP model.

There is no general fact in the literature to prove that either continuous-time or discrete-time models are superior in terms of computational efficiency for real-world scheduling problems. Normally, the continuous-time models require lower number of variables and constraints compared to the discrete-time models. However, the constraints of the continuous time approach can be more complicated and may result in greater computational complexity (Stefansson et al., 2011).

In discrete-time modeling of the scheduling problems, the constraints need to be monitored only at specific and pre-determined points of time. This reduces the complexity of the model and makes it structurally simpler and easier to solve, especially when there exist constraints of inventory or capacity. However, the size of the mathematical model highly depends on the duration (or number) of the uniform time intervals postulated for the scheduling horizon.

Although this type of modelling is considered a simplification of the original problem (due to time approximations), its efficiency and adaptability to various real-world industrial scheduling problems has been proved, especially in cases that a small number of time intervals is enough to develop the model.

On the other hand, continuous-time approaches need significantly lower numbers of binary variables and lead to solutions with higher quality in shorter times. However, formulating some specific aspects of the real-world problems (e.g. limitations associated with resources and inventories) usually calls for defining more complicated constraints with many big-M terms which increase the complexity of the model (Méndez et al., 2006).

Stefansson et al. (2011) analyze the advantages and disadvantages of continuous-time versus discrete-time MIP formulation for the problem of scheduling a large real-world multi-stage and multi-product flow shop in pharmaceutical industry. In their formulations, the discrete time model has a significant larger number of variables while the continuous time model has a considerable larger number of constraints. However, as they state, the number of constraints and variables is not a very illustrative criterion to define the complexity of a model and the solution efficiency highly depends on the specific problem, the solution space and the formulation structure of the model. Nonetheless, in the case of their pharmaceutical operations problem, the continuous-time model is superior to the discrete-time formulation in terms of solution quality and computation time.

Floudas and Lin (2004) study different practices of scheduling of multi-product and multi-stage batch and continuous chemical processes based on continuous and discrete time representation. They suggest that the most crucial advantage of the discrete-time modeling approach is that it provides a reference grid of time for all shared processes. This facilitates the formulation of diverse types of scheduling constraints (especially resource constraints) in a quite simple and relatively straightforward structure. However, the concept of discretization of continuous time denotes that this type of modelling is actually an approximation of the original problem. In addition, deciding about the duration of the uniform time intervals (dividing the scheduling horizon to equal time units) is a matter of tradeoff between accuracy (quality of the solution) and the complexity of the model (required computational time).

Floudas and Lin (2005) review the models which are developed based on Mixed Integer Linear Programming (MILP) formulations in order to solve scheduling problems in chemical processing systems. They categorize the models in two groups of continuous-time and discrete-time formulations and suggest a number of advantages and disadvantages for each category. In general, while discrete-time models are more flexible in terms of defining constraints and provide a simple modeling structure, they usually have a huge number of variables and constraints which reduce the efficiency of the model. On the contrary, continuous-time models need lower number of variables but formulation of constraints is sometimes a challenging issue which results into higher levels of complexity of the model. They also study different approaches proposed for increasing the efficiency of MILP models from a computational point of view.

According to their report, the most fruitful approaches for improving the computational efficiency of the models are reformulation of the constraints, introduction of additional cut constraints (in order to opt out infeasible solutions at early stages of the branch and bound searching process), use of heuristics to simplify the problem and reduce the search space, decomposing the large MILP model to smaller sub-problems, and intervention of the branch and bound solution procedure.

The reason we decided to use the discrete-time modeling approach for this research was the fact that modeling some specific aspects of the Composite Manufacturing Center problem (e.g. limitations associated with resources and the physical capacity constraint of the buffer) can be modeled more properly by means of discretizing the scheduling horizon.

However, in order to propose a remedy to the long run times of CPLEX for large instances of the problem, we have developed a heuristic method along with a GA to obtain near-optimal solutions in far shorter times. We will introduce and explain both the novel heuristic method and the GA in chapter four in detail.

4 A Novel Heuristic Approach Combined with a Genetic Algorithm

In order to effectively reduce the CPU run times and boost the efficiency of the solution method for the problems which are solved by the MILP model described in the previous chapter, in this chapter, firstly, we aim at developing a heuristic algorithm to obtain some feasible initial solutions. This heuristic algorithm is unique to the Hybrid Flow Shop problem which is described in chapter 3. We developed this heuristic method by carefully scrutinizing the behavior of the MILP model for different problem sets and understanding how the model tries to schedule the jobs to satisfy the objective function and the constraints. Thus, the idea behind the heuristic algorithm originates from the observation of the MILP model and the Gantt charts drawn from the results of CPLEX.

Afterwards, we develop a GA with discrete-time based chromosome encoding scheme that uses the aforementioned heuristic algorithm for generating the initial population and through its stochastic iterations improves and updates the initial solutions until it obtains satisfactory results and terminates the algorithm.

In the following, we will introduce the novel heuristic method for generating the initial feasible solutions and then the GA will be presented.

4.1 The Novel Heuristic Method for Generating Initial Solutions

This algorithm is designed in a way that receives a list of data from the user as inputs, then starts processing the data and going through the algorithmic steps and finally delivers the outputs in terms of the start time of the jobs in the first stage (Lay-up Process).

4.1.1 Inputs

The algorithm starts with getting the required data about the jobs in order to create the initial population.

Table 3. The Input Data for the Heuristic Algorithm

Job Number	Lay-up Type	Laser Requirements	Cure Type	Mold Type	Processing Time	Mold Size	Due Date (Days Left)
1	1	0	4	12	8	10.6	10
2	5	1	3	56	10	14.2	15
…	…	…	…	…	…	…	…
n	3	0	2	34	6	12.3	8

The input data looks like the given example in table 3. Normally, the data can be drawn from the available data warehouses in the company's information system. Thus, only the number and type of the demanded parts (part numbers) and their due dates are enough to get all the other required information. Linking this input section of the algorithm to the MRP system is a very simple task.

4.1.2 Steps of the Algorithm

1. Make a sequence of available cure processes for each job based on their cure type.
2. Sort the jobs according to their Cure Type in ascending order.
3. Sort jobs inside each category (Cure Type) based on their ascending Processing Time (SPT).
4. Let *Lay-up (lu, c) Busy*, *Laser (lu, c) Busy* and *Cure (c) Occupancy* and *Mold (j, c) Occupancy* be equal to zero. *Lay-up (lu, c) Busy* is referred to the number of operators inside the lay-up cell (lu) which are busy before cure (c) starts. *Laser (lu, c) Busy* is referred to the number of laser-equipped operators inside the lay-up cell (lu) which are busy before cure (c) starts. *Cure (c) Occupancy* is the amount of space inside cure (c) which is already occupied. *Mold (j, c) Occupancy* is the number of type j molds which are already used for *Cure (c)*. *Mold (j, c)* is a binary variable which is equal to 1 if Mold (j) is used for a part which is going into cure (c).
5. Assign the first unassigned job on the list to the first available cure process c by starting its operation at a lay-up cell at time [Start Cure (c) – Processing Time Job (i)].

   If the value of [Start Cure (c) – Processing Time Job (i)] is smaller than 1,

   Or if *Lay-up (lu, c)* Busy is equal to *Lay-up (lu, c) Capacity*,

   Or if *Laser (lu, c)* Busy is equal to *Laser (lu, c) Capacity*,

   Or if *Cure (c) Capacity* is not greater than *Cure (c) Occupancy*,

   Or if *Mold (j, c) Capacity* is not greater than *Mold (j, c) Occupancy*,

   Then, assign the job to the next available cure process (c′) by starting its operation at a lay-up cell at time [Start Cure (c′) – Processing Time Job (i)].
6. Update *Lay-up (lu, c) Busy* → *Lay-up (lu, c) Busy* = *Lay-up (lu, c) Busy* + 1 if the job is assigned to *Lay-up (lu, c)*
7. Update *Laser (lu, c) Busy* if Laser Requirements of the assigned job is equal to 1 → *Laser (lu, c) Busy* = *Laser (lu, c) Busy + 1*, if Laser Requirements of the assigned job is equal to 0 do not modify *Laser (lu, c) Busy*
8. Update *Cure (c) Occupancy* → *Cure (c) Occupancy* = *Cure (c) Occupancy* + *Mold Size (i)* if the job is assigned to *Cure (c)*
9. Update *Mold (j, c) Occupancy* → *Mold (j, c) Occupancy* = *Mold (j, c) Occupancy* + *Mold (j, c)*
10. Terminate if all the jobs in the list are assigned, otherwise go to number 5.

4.1.3 Outputs

Using this algorithm, we will be able to obtain one initial feasible solution to the problem. By altering the step 3, we can easily generate more initial solutions. Therefore, as another part of the algorithm, we change step 3 to one of the following steps each time and obtain a different initial feasible solution:

- Sort jobs inside each category (Cure Type) based on their descending Processing Time (LPT)
- Sort jobs inside each category (Cure Type) based on their ascending Due Dates (EDD)
- Sort jobs inside each category (Cure Type) based on their ascending Mold Size
- Sort jobs inside each category (Cure Type) based on their descending Mold Size
- Sort jobs inside each category (Cure Type) based on their ascending Lay-up Type
- Sort jobs inside each category (Cure Type) based on their descending Lay-up Type
- Sort jobs inside each category (Cure Type) based on their ascending Job Number

Therefore, we will have eight initial feasible solutions in total. With these eight initial solutions, we can create the initial population of the GA.

Each one of these solutions is a list of Job Numbers with their start times in the first stage. Having the information about the cure type of each Job Number, their start times in their respective cure process can be easily inferred.

Table 4 shows an example of the output we can get from the described heuristic algorithm for a problem with 10 jobs.

Table 4. The Output of the Heuristic Algorithm

Job Number	1	2	3	4	5	6	7	8	9	10
Start Time at the First Stage	4	61	2	88	91	53	59	95	1	49

4.2 The Proposed Meta-heuristic: A Genetic Algorithm

A meta-heuristic is a method to solve a more general class of problems by combining heuristics and user-given procedures in an efficient way. The name includes the Greek prefix "meta" which means "beyond" and refers to the higher level of heuristics. Meta-heuristics can be applied to problems which are not solvable with optimal solution methods or other problem-specific algorithm or heuristics; or when the solutions from such methods are not practical to implement or satisfactory enough (Zandieh et al., 2010).

Genetic algorithms, introduced by John Holland (1975), are iterative stochastic algorithms that use the idea of natural evolution to model the search method. A GA is a meta-heuristic algorithm that provides an algorithmic framework and uses a collection of initial solutions (initial population) which evolve through genetic operators (selection, crossover, mutation and replacement) to obtain improved solutions. GA evolves solutions for problems that have massive solution spaces and are not easily dealt with the exhaustive search methods or traditional optimization techniques (Oĝuz and Ercan, 2005).

Normally, a GA starts with an initial population of candidate solutions. In this population, solutions are encoded as a string (typically, binary or integer) and are usually referred to as *chromosomes*. Each chromosome is composed of *genes* that characterize the solution. Each chromosome is evaluated by a criterion, which is determined by the associated value of the objective function. This criterion is referred to as *fitness function*. Through *crossover* and *mutation* processes, the GA evolves the population towards an optimal solution.

Solutions drawn from the stochastic iterations and genetic process of a GA are proven to be capable of converging towards the optimal solution. Thus, the GA iterations tend to maximize the likelihood of generating such a solution. The first step is typically to evaluate the *fitness* of each candidate solution in the current population, and to select the fittest candidate solutions to act as parents of the next generation of candidate solutions. Afterwards, these selected parents are recombined by means of a crossover operator and mutated by means of a mutation operator to generate offspring. The fittest parents and offspring form a new population and the worst fitted individuals die to maintain the desired population and the described procedure is repeated again and again to create new populations until a stopping criterion is satisfied. The output of this evolution process is the best individual in the final population, which can be a highly evolved solution to the problem. The following algorithmic steps in the figure 8 demonstrate a pseudo code of the standard GA (Zandieh et al., 2010).

```
1- Initialization
        1-1- Set parameters (population_size, generation_number, percent_crossover, percent_mutation,…)
        1-2-Generate initial population (Randomly)
2- Evaluate fitness of each solution
3- Form new generation
        3-1- Select individuals for mating pool
        3-2- Apply genetic operators (crossover, mutation and reproduction) based on selection strategy
        3-3- Replace current population with new generation
4- Stop if stopping criteria is met; otherwise go to step 2
```

Figure 8. A standard GA in pseudo code (Zandieh et al., 2010)

The operations of evaluation, selection, recombination and mutation are usually performed many times in a GA. Selection, crossover, and mutation are common operations which are used in any GA and have been comprehensively investigated in the literature. However, evaluation is unique to the problem and is specifically and directly related to the structure of the solutions (i.e. the encoding of the chromosomes). Therefore, in a GA, the major challenge is to decide about the structure of solutions (chromosomes) and to select the method of evaluation (fitness function).

Other parameters include the size of the population, the portion of the population participating in the crossover operation, and the mutation rate which defines the probability with which a gene in a chromosome is mutated.

In the following, different features of the proposed GA will be presented in detail.

4.2.1 Encoding/Decoding Scheme and the Structure of Chromosomes

In Genetic Algorithms, each chromosome reflects a solution to the problem. Each chromosome is a string of genes. In this proposed algorithm, the number of genes represents the number of jobs which are ready to be scheduled. Each gene represents one of the jobs. Therefore, we assign a number to each job and gene to connect them (ex. the first job in the list is the first gene from left in the string, the second job is the second gene from left, and so on.). The value inside each gene is a natural number within the range of [1, 480 – min (processing time + cure time)] which shows the start time of the job at one of the lay-up cells. The job starts its lay-up processing at the time which is indicated by its respective gene. After being completely processed, it goes to the very first available cure process whose curing recipe suits the requirements of that particular job. For example, a chromosome for a scheduling problem with ten jobs is depicted below:

20	126	1	42	58	215	12	18	1	65

4.2.2 Initialization

An initial population of a certain number of chromosomes (solutions) is needed in each GA. Normally, these solutions are randomly generated. However, in order to make the algorithm more efficient, it's better to generate an initial population with feasible solutions which are closer to the optimal solution compared to the randomly generated solutions. To do so, we have developed a heuristic method to generate a pool of initial solutions for the GA. The initial population of the proposed GA is composed of the eight solutions we obtain from the heuristic algorithm described in section 4.1.

4.2.3 Fitness Function

The fitness function for each chromosome is defined as the utilization of the autoclaves while avoiding generating infeasible solutions. This technique incorporates the constraints into the fitness function in a dynamic way. It consists of the objective function of the original scheduling problem along with four penalty terms taken into account for the four main constraints of the problem which are namely the limited resource capacity of the lay-up cells, the limited physical capacity of the buffer, the limited physical capacity of the autoclaves and the limited number of molds. The resulting varying fitness function facilitates the GA search.

$$\sum_{i \in I} \sum_{c=1}^{25} Y(i,c) * c + \alpha * Max\left(0, \left[\left(\sum_{i \in I(lu)} M(i,j) * X(i,t) + \sum_{i \in I(lu)} M(i,j) * Z(i,t) + \sum_{i \in I(lu)} M(i,j) * Y(i,c) * K(c,t)\right) - T(j)\right]\right) + \beta * Max\left(0, \left(\sum_{i=1}^{I} Z(i,t) * SizeMold(i) - Buffer\ Capacity\right)\right) + \Omega * Max\left(0, \left(AutoclaveCap(c) - \sum_{i=1}^{J} Y(i,c) * SizeMold(i)\right)\right) + \lambda * Max\left(0, \left(\sum_{i \in I(lu)} X(i,t) - OperatorCap(lu)\right)\right) + \sigma * Max\left(0, \left(\sum_{i \in I(lu)} X(i,t) * Laser(i) - LaserCap(lu)\right)\right)$$

Where:

M (i,j)	1 if the job (i) needs the mold (j). Otherwise it's zero.
X (i,t)	Whether or not job (i) is being processed in time (t) in one of the lay-up cells. If X (i,t) is 1 it means that job (i) is being processed in time (t). Otherwise it's zero.
Y (i,c)	1 when job (i) is assigned to Cure (c). Otherwise it's zero.
Z (i,t)	1 when job (i) is in the buffer during time (t) and it's zero when job (i) is not in the buffer.
K (c,t)	1 when the cure number (c) is active, otherwise it's zero.
Laser (i)	1 when job i needs a laser station to be processed and 0 when it doesn't
T (j)	Number of mold (j) available in the system

The factors α, β, Ω, λ and σ are among the control parameters of the GA that should be tuned in the parameterization phase.

The fitness function is a basis of the selection operator. Here, we adopt the elitist principle which means the individual with the best fitness does not need to participate in the selection and reproduction operations (crossover and mutation) and goes directly to the next generation.

4.2.4 Selection

We propose the roulette wheel method as the selection technique for this GA. The aim of selection is to keep good chromosomes and eliminate bad ones. The selection is based on the fitness value of individuals. Since the elitist principle is applied, only one of the chromosomes will be reserved in the next generation, the other individuals (Population Size - 1) are selected based on the roulette wheel method and form the next generation's mating pool.

Step 1. Let 1/f(x) denote the fitness value and calculate the probability of a selection p(x) for each chromosome.

$$p(x) = \frac{\mathrm{f(x)}}{\sum_{x=1}^{\text{Population Size}-1} f(x)}$$

Step 2. Calculate the cumulative probability q(x) for each chromosome.

$$q(x) = \sum_{h=1}^{\mathrm{x}} p(h)$$

Step 3. Generate a random number r from (0,1]. If r < q(x), select the first individual; otherwise select the xth individual (2 ≤ x ≤ Population Size - 1) such that q(x-1) ≤ r ≤ q(x).

Step 4. Repeat step 2 and 3 until (Population Size – 1) individuals are selected.

4.2.5 Crossover

The purpose of designing the crossover operator is to keep the best features of each parent and randomly generate the remaining features in forming the offspring. In this algorithm, a new crossover operator will be introduced and employed.

In this method, two chromosomes are randomly selected from the mating pool. The chromosome with a larger fitness function is named C1 and the one with a smaller fitness function is named C2. Then an array of n elements (number of jobs – length of the chromosomes) consisting of random numbers between 0 and 1 is constructed. If the value of the jth member of this array is between zero and 'a', then the jth gene of the offspring chromosome is taken from the jth gene of chromosome C1. If this value is between 'a' and 'b', the jth gene of the offspring chromosome is taken from the jth gene of chromosome C2.

If this value is between ‘b’ and ‘c’, the jth gene of the offspring chromosome equals to (the jth gene of chromosome C1 - P_{max} (c)). If this value is between ‘c’ and 1, the jth gene of the offspring chromosome equals to (the jth gene of chromosome C2 - P_{max} (c)).

The values ‘a’, ‘b’, and ‘c’ are numbers between zero and one and they are among the algorithm’s parameters which should be specified. P_{max} (c) is the maximum processing time of all jobs with cure type (c).

For example, let a, b and c be 0.25, 0.5 and 0.75 respectively. One possible crossover operation is depicted below:

Chromosome C1	20	126	1	42	58	215	13	18	1	65
Chromosome C2	30	13	41	5	69	156	20	1	25	4
P_{max} (c)	10	12	15	12	10	15	12	10	15	10
Array 1	0.36	0.2	0.96	0.32	0.1	0.38	0.18	0.7	0.86	0.07
Offspring 1	30	126	41-15	5	58	156	13	18-10	25-15	65-10
Offspring 1	30	126	26	5	58	156	13	8	10	55
Array 2	0.66	0.72	0.86	0.38	0.19	0.28	0.68	0.51	0.16	0.47
Offspring 2	20-10	126-12	41-15	5	58	156	13-12	18-10	1	4
Offspring 2	10	114	26	5	58	156	1	8	1	4

4.2.6 Mutation

The aim of mutation operator is to introduce variations into solutions. Moreover, more solution space can be explored and the chance of being trapped in the local optimum is reduced. We employ swap operator for mutation which is very common in genetic algorithms designed for scheduling problems.

In this mutation method, we randomly select two genes from a chromosome (with a probability that should be specified as one of the GA parameters) and then exchange their positions. The example below, illustrates the function of this mutation operator:

Parent	20	126	**1**	42	58	215	12	**18**	1	65
Offspring	20	126	**18**	42	58	215	12	**1**	1	65

4.2.7 Termination Criteria

We can employ two different techniques as the termination criterion:

1. If the best value for chromosomes fitness functions does not improve in a number of consecutive generations (which is another parameter of the algorithm), the algorithm is terminated.
2. A certain number of iterations which is again a parameter of the algorithm.

All these parameters would be specified after running a series of test problems.

4.2.8 Parameters Setting

The performance of a GA depends greatly on the control parameters. These main parameters (population size, crossover rate, mutation rate and number of generations) can remarkably influence the efficiency of the GA. The population size is the certain number of chromosomes in each generation. The crossover rate is the probability of applying the crossover operation in the process of reproducing new individuals. The mutation rate is the probability of using the mutation operator in the chromosomes reproduction process. And number of generations indicates the number of iterations in the process of evolution until the algorithm terminates.

Besides these main parameters, in the proposed GA there are some other parameters that need to be tuned. The parameters used in the fitness function (α, β, Ω, λ and σ) and crossover operator (a, b and c) are among these parameters.

After coding the algorithms in C# and running the code with various test problems and different parameter combinations and comparing the efficiency of the algorithm with respect to the quality of the results, the following values are selected as the best combinations of the parameter values for this specific GA.

Table 5. Selected values for parameters of the GA

	Problem Size		
	Small **Up to 40 Jobs**	**Medium** **40 to 100 Jobs**	**Large** **More than 100 Jobs**
Crossover Rate	0.5	0.8	0.85
Mutation Rate	0.2	0.1	0.5
Population Size	8	8	8
Number of Generations	100	600	1000
α	10	20	50
β	10	20	50
Ω	10	20	50
$\boldsymbol{\lambda}$	10	20	50
$\boldsymbol{\sigma}$	10	20	50
a	0.25	0.25	0.25
b	0.5	0.5	0.5
c	0.75	0.75	0.75

5 Numerical Experiments, Results and Analysis

In this chapter, the MILP model as well as the proposed heuristic method and the GA are applied to different sets of data and the results are presented. Data used in these numerical experiments are gathered from the Composite Manufacturing Center of the ABC Company. For the purpose of confidentiality, the data is transformed and masked in a way that their specific realistic characteristics are preserved. Therefore, each problem is defined as a set of jobs to be scheduled and these jobs are presented with a job number and a number of other attributes assigned to them such as lay-up type, laser requirements, and cure profile and so on (i.e. Table 3 in section 4.1.1)

In the following sections, first, a computational analysis of different methods is conducted in which the MILP model and the GA are used to solve small, medium and large instances of the problem. Afterwards, a comparison of the efficiency and accuracy of these two solution approaches is drawn through which the main differences of them are revealed. The advantages and disadvantages of each method are also discussed in this section. Finally, a section for discussions about the results and their implications is prepared to give the users of this research some insight on how and when to use the different proposed solution methods for different types of problems.

5.1 Computational Analysis

In this section, the proposed solution approaches (i.e. the MILP model and the heuristic/GA) are applied to three levels of small, medium and large-size problems. Small-size problems in this research are defined as scheduling problems with up to 40 jobs. Medium-size problems are those with maximum 100 jobs and large-size problems are the ones with more than 100 jobs.

For each level, different proposed methods are used several times for various data sets and the average of the results are summarized in the table 6.

Table 6. Summary of the Results of Applying MILP and GA to Small, Medium and Large-size Problems

		Problem Size		
		Small **Up to 40 Jobs**	**Medium** **40 to 100 Jobs**	**Large** **More than 100 Jobs**
Average Value of the Objective Function for the Initial Solutions (rounded to the nearest integer)	SPT	81	285	500 <
	LPT	89	293	500 <
	EDD	85	285	500 <
	Descending Mold Size	87	287	500 <
	Ascending Mold Size	82	295	500 <
	Descending Lay-up Type	90	293	500 <
	Ascending Lay-up Type	89	295	500 <
	Ascending Job Number	94	298	500 <
Average Value of the Objective Function	MILP	65	236	-
	GA	78	254	470 <
Average Run Time (minutes)	MILP	2	40	-
	GA	1	2	9

In the following, the results for each level of problems will be discussed and the performance of the proposed methods will be analyzed.

5.1.1 Small-size Problems

For problems with up to 40 jobs, the MILP model works perfectly. The optimal schedules for composite manufacturing systems, which are modeled as described in chapter 3, can be obtained within 2-3 minutes.

In the next page, an example of a problem with 40 jobs is illustrated by means of Gantt Charts. As it can be observed in figure 9, first, the start time of the jobs in their respective lay-up cell is specified. Then, the jobs are grouped in batches and are sent to an autoclave whose operating cure process is compatible with their cure profile. The vertical axis of the figure 9 shows the resources (lay-up cells and autoclaves) while the horizontal axis displays the time horizon.

As it was mentioned before in Section 3, it should be noted that the scheduling time horizon is discretized to 15 minutes intervals. Therefore, each scheduling day (three shifts of 8 hours) is shown as 96 time intervals (96 * 15 minutes = 24 hours). Thus, figure 9 displays one day of the scheduling horizon. In other words, all the 40 jobs are scheduled for the first day lay-up and cure processes.

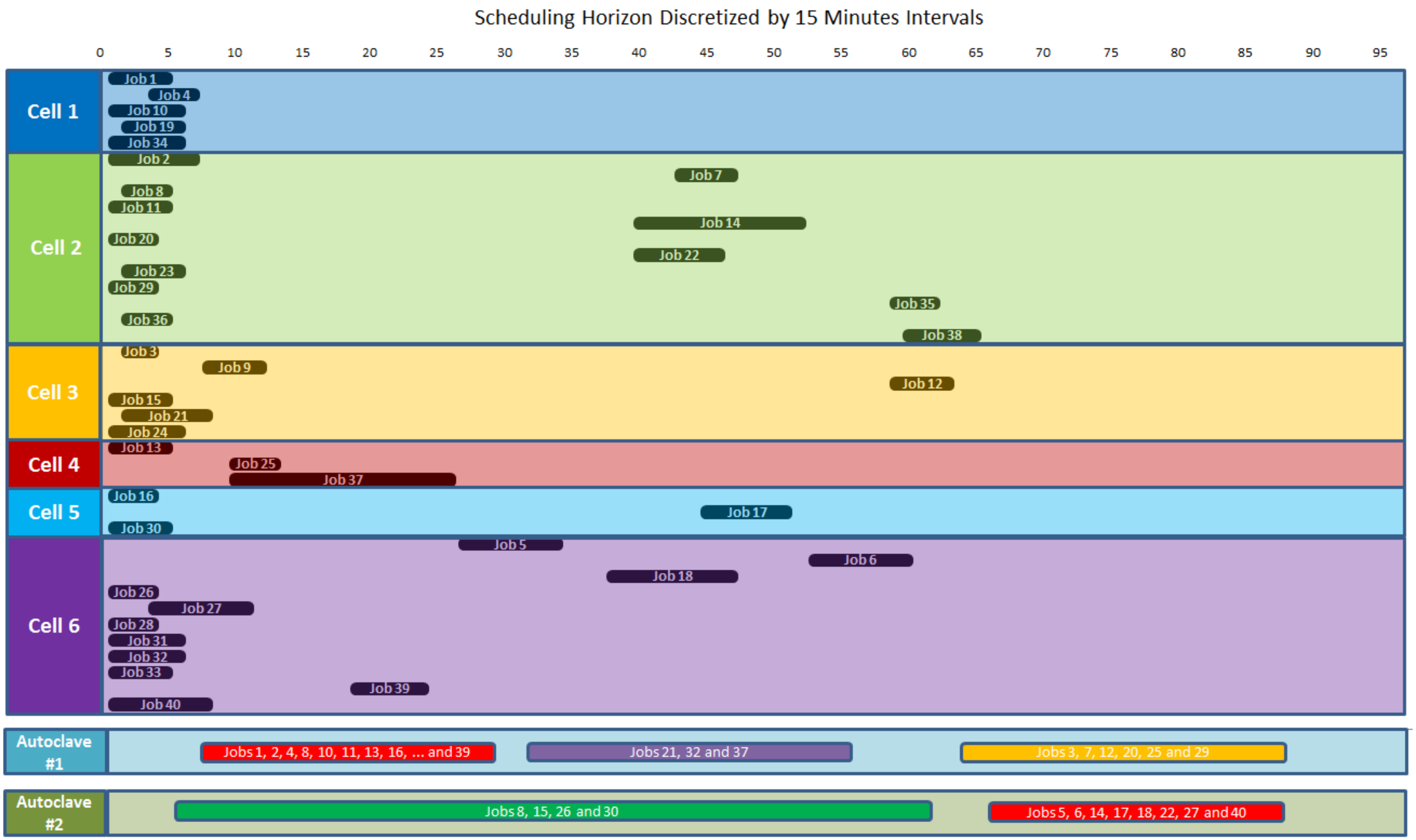

Figure 9. Optimal schedule obtained from the MILP model when applied to a problem with 40 jobs

The proposed MILP model strives for pushing the jobs toward the earlier cure treatments. Therefore, the earlier batches are utilized as much as possible first and then, if there is any job left, the next cure operations will be utilized. Table 7 shows the utilization of the autoclaves for the five cure treatments for the described example. As it can be observed, for cure number 1 and 5 which provide the same cure treatment (red), the earlier cure process (cure number 1) is utilized as much as possible and then, the remained jobs are assigned to cure number 5.

Table 7. Utilization of the Autoclaves

Cure Type	Cure Number	Autoclave Number	Utilization	Number of Jobs Assigned
Red	1	1	92%	21
Red	5	2	32%	6
Purple	2	1	11%	3
Yellow	3	1	22%	6
Green	4	2	29%	4

The other important point about the solutions obtained from the MILP model is that most of the jobs are assigned to the lay-up cells just before their cure process starts. The reason behind this manner of job assignment is the buffer capacity constraint. The model tries to minimize the waiting time of the jobs in the buffer area. Thus, it schedules them in a way that they can directly move to the cure operation (without waiting in the buffer) when their lay-up process in finished.

The heuristic algorithm is also used to solve the problems with up to 40 jobs. The eight initial solutions which are the outputs of the heuristic algorithm are generated within 5 seconds. The solutions with the underlying SPT, LPT and EDD dispatching rules are presented in the next pages as examples of the heuristic algorithm. Figures 10, 11 and 12 show the results of the heuristic method which are obtained based on SPT, LPT and EDD rules, respectively.

The GA is also used to improve these initial solutions and achieve the near-optimal solutions. The results of applying the GA to the same problems, that were solved by the MILP model before, show that for problems with small numbers of jobs, the solutions obtained from the GA does not differ significantly from those which are generated by the heuristic algorithm with an underlying SPT rule because these solutions are quite close to the optimal solutions. However, the solutions from the heuristic algorithm with some of the other dispatching rules will be improved significantly through stochastic iterations of the GA. Since the elitist principle is adopted in the GA, the chromosome related to the solutions obtained from SPT rule will remain in all iterations and will be slightly improved.

Figure 13 shows the solution which is obtained by the GA for 40 jobs. A simple comparison of this solution with the three previous figures (which are examples of the solutions obtained from the heuristic algorithm) shows that the GA does not change the initial solutions a lot for small-size problems. The reason is the fact that there is almost no room for improvement for small instances of the problem and the heuristic method itself generates very good solutions. These solutions are already very close to the optimal solutions obtained from the MILP model and even in some cases they are optimal. Thus, it is clear that the GA terminates quite fast (in less than one minute).

The values of the objective function both for the initial solutions obtained from the heuristic method and the final solutions obtained from the GA are very close to the values of the objective function for the optimal solutions derived from the MILP model.

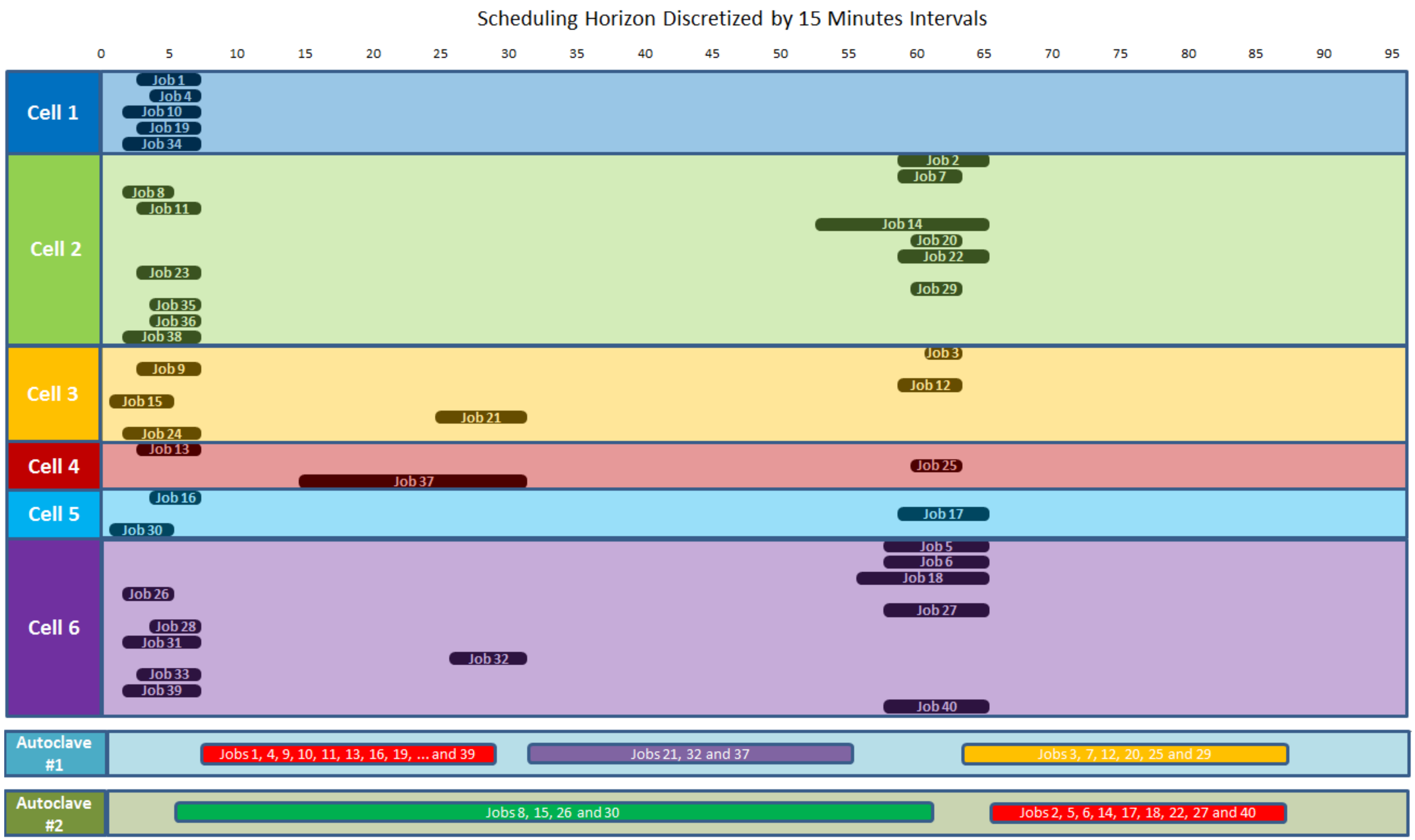

Figure 10. Initial schedule obtained from the heuristic algorithm with an underlying SPT rule

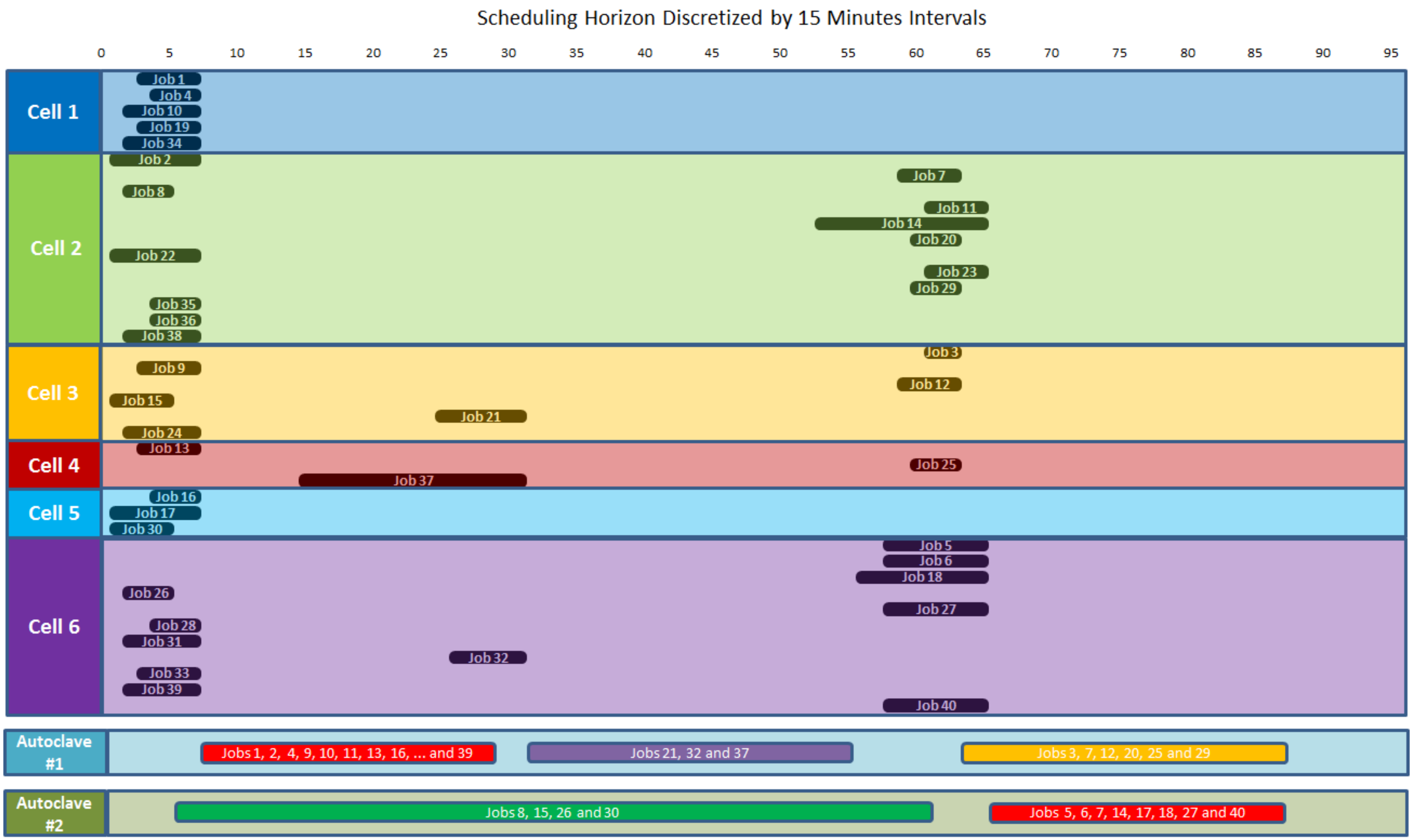

Figure 11. Initial schedule obtained from the heuristic algorithm with an underlying LPT rule

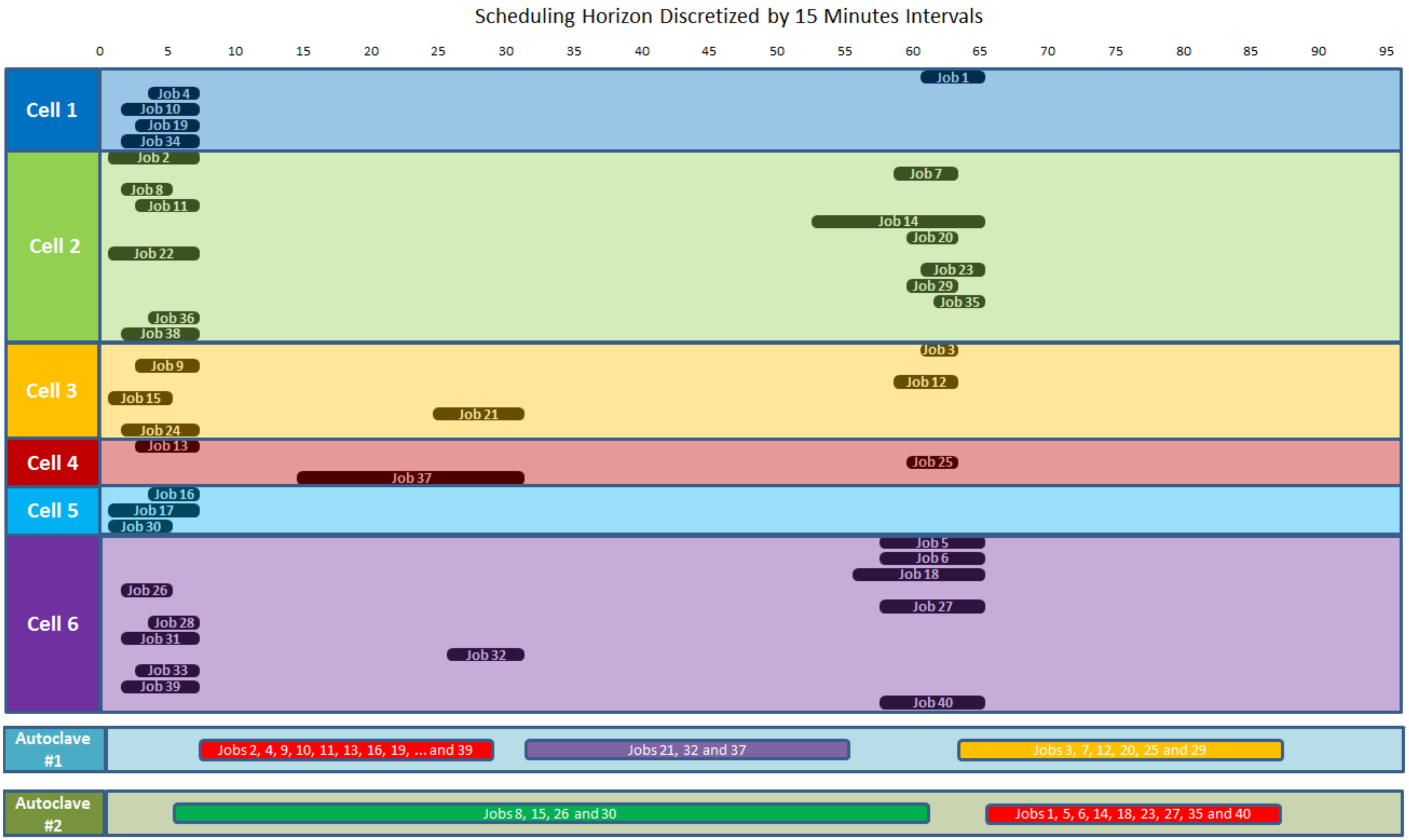

Figure 12. Initial schedule obtained from the heuristic algorithm with an underlying EDD rule

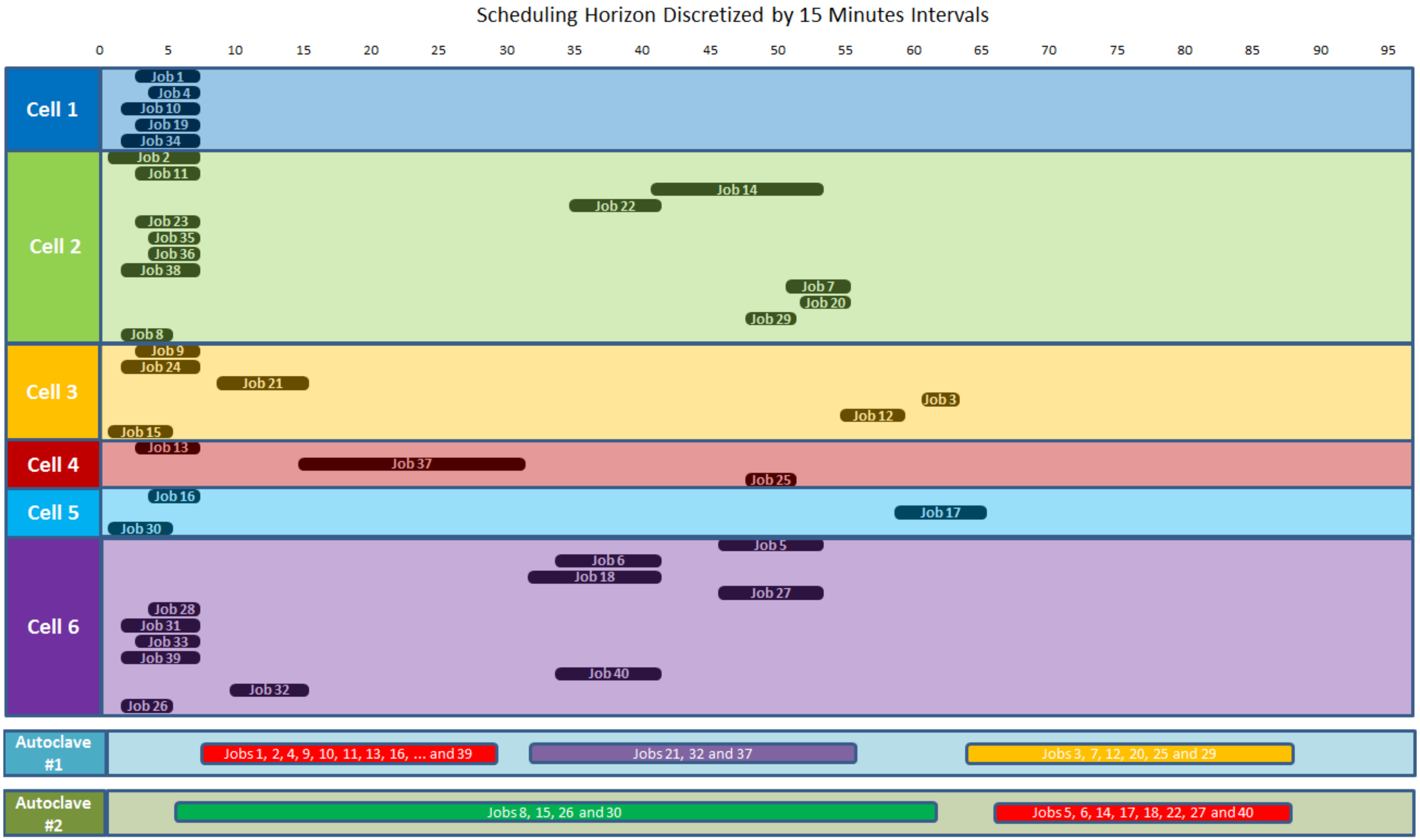

Figure 13. Near-optimal schedule obtained from the GA when applied to a problem with 40 jobs

5.1.2 Medium-size Problems

For problems with 40 to 100 jobs, the MILP model can obtain the optimal solutions. However, the run times increase significantly by increasing the number of jobs. For problems with up to 70 jobs, CPLEX can achieve the optimal solutions within 45 minutes. When it goes beyond 70, the run times start to increase dramatically. The average run times for problems with 80, 90 and 100 jobs are 112, 635 and 1464 minutes, respectively.

The reason for such a significant increase in the run times is the enormous number of constraints and variables of the MILP model when the number of jobs increases. Hybrid Flow Shop problems are NP-hard problems and it is quite normal that for large instances of the problem, exact solution methodologies (i.e. MIP) have less computational efficiency and sometimes they are even unable to solve the problems.

On the other hand, the heuristic algorithm generates the initial solutions for these problems within a few seconds. Although the quality of these solutions is not as high as the results from the MILP model, the stochastic iterations of the proposed GA enables them to evolve and finally transform into high quality solutions which are comparable to those of the exact solution methods.

GA obtains the near-optimal solutions for medium-size problems within 4-5 minutes. A comparison of the results reveal that the objective function of the solutions obtained from the GA is very close to that of the MILP model when applied to the same problem sets. However, the run times of the GA are far shorter than those of the MILP model.

In the next pages, the results of the MILP model, the heuristic method with an underlying SPT rule and finally the GA are shown through figures 14, 15 and 16. It is worth mentioning that the other initial solutions generated by the heuristic method (based on the other dispatching rules) have the similar quality as the heuristic method with SPT rule. However, only the results from the SPT rule are shown as a sample of the outputs of the heuristic algorithm when applied to medium-size problems.

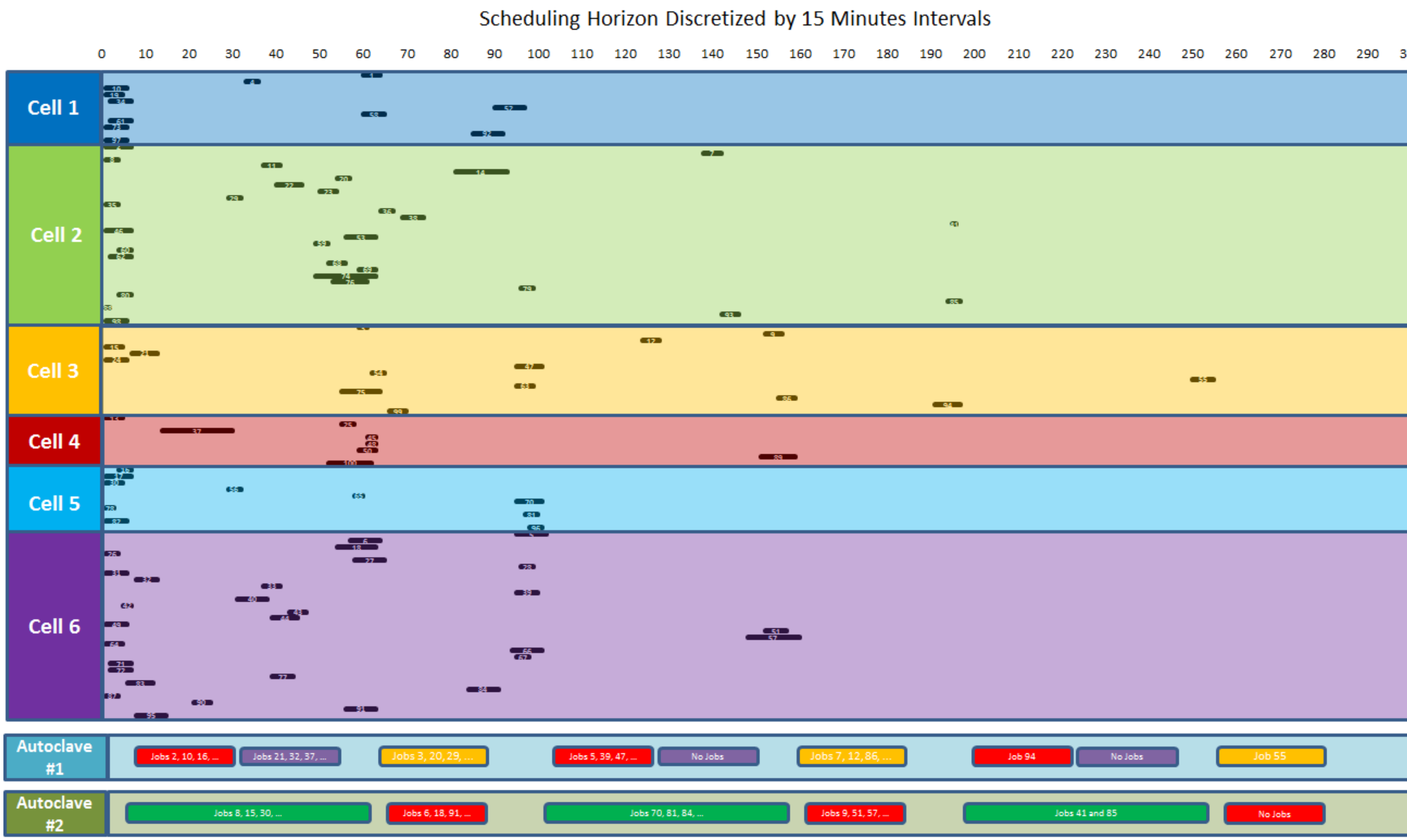

Figure 14. Optimal schedule obtained from the MILP model when applied to a problem with 100 jobs

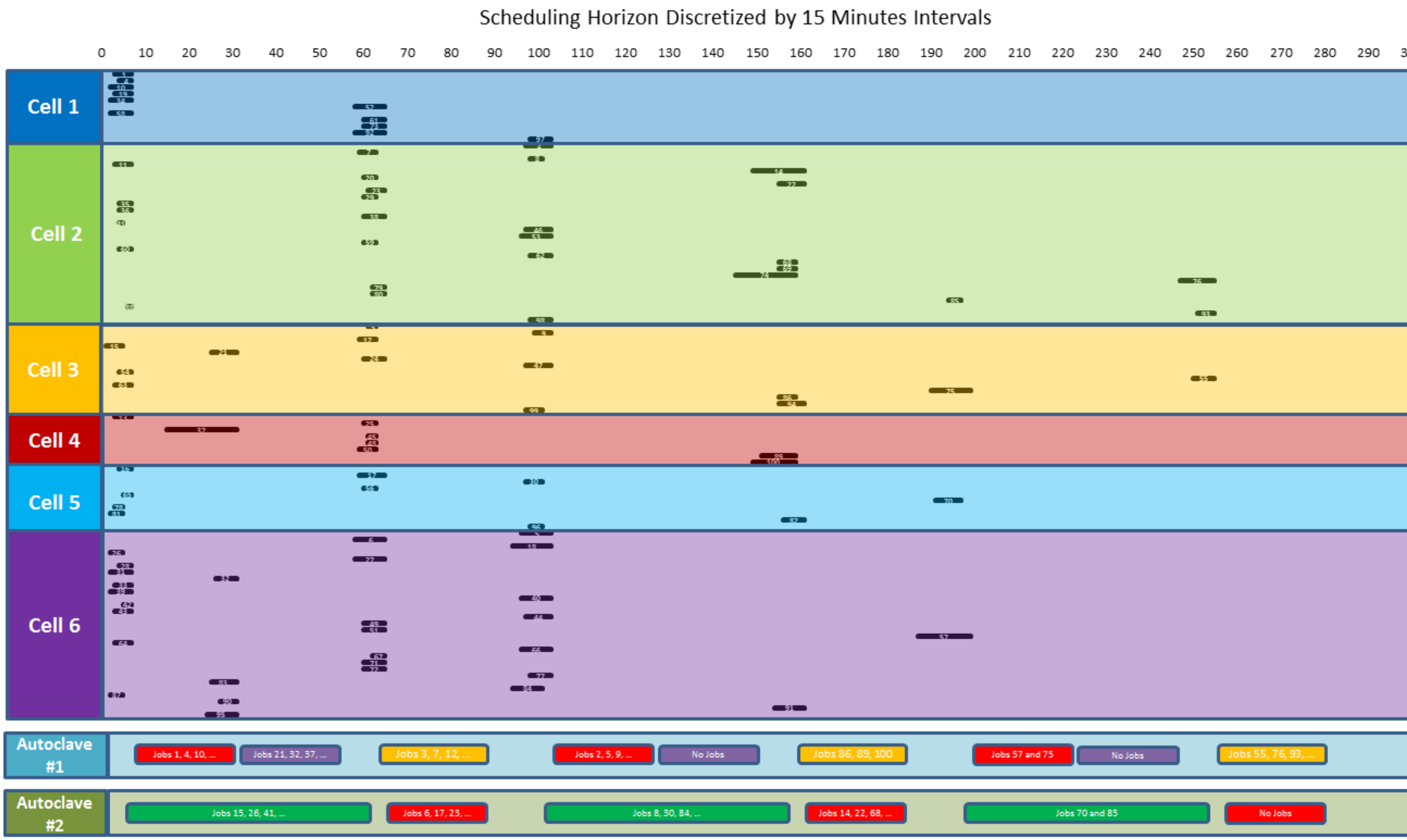

Figure 15. Initial schedule obtained from the heuristic method with an underlying SPT rule when applied to a problem with 100 jobs

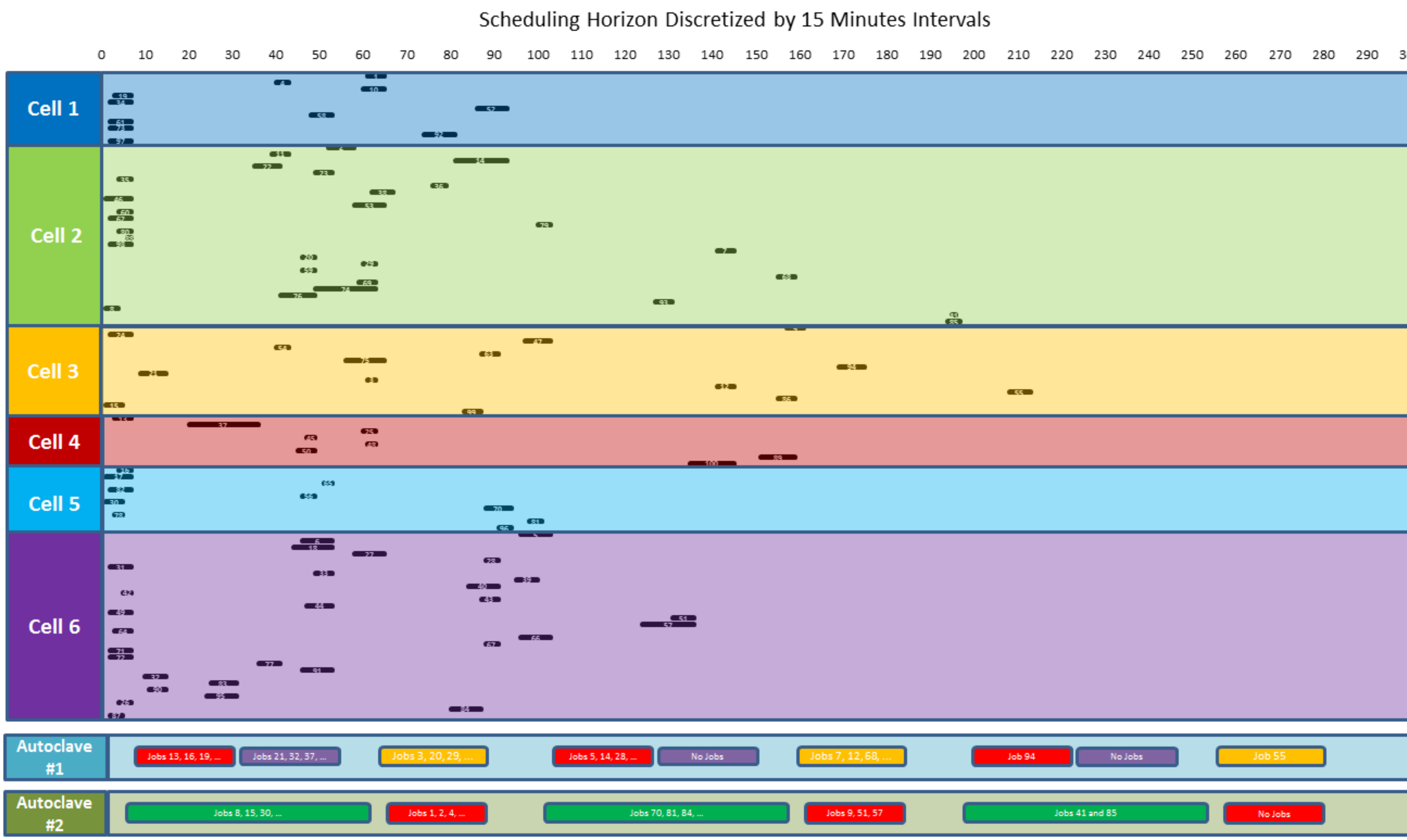

Figure 16. Near-optimal schedule obtained from the GA when applied to a problem with 100 jobs

5.1.3 Large-size Problems

For problems with more than 100 jobs, CPLEX could not achieve the optimal solutions in reasonable CPU times. While the MILP model is useful for small and medium-size problems, for the larger numbers of jobs, the number of constraints and variables grows dramatically and the computation time grows exponentially. This is a very common barrier in the scheduling research area; complex scheduling systems which are modeled with exact solution methodologies usually fail to address the large instances of the problems within practical run times.

However, near-optimal solution methodologies (i.e. metaheuristics) are proven to be very useful and efficient in these cases. The heuristic algorithm generates the initial solutions for large-size problems (up to 300 jobs) in less than 10 seconds. Then, the GA obtains the near-optimal solutions for these problems within 8-12 minutes, depending on the number of jobs. Since the comparison of the results for small and medium-size problems with MILP model results (optimal solutions) show the high quality of the results from the GA, its results for the large-size problems can be considered strongly reliable.

The results from the GA compared to those of the heuristic method shows the high levels of improvement in solution quality which is achieved by the stochastic iterations of the proposed GA. The main reason for such improved solutions is that there is a quite large room for modification of the initial solutions when the size of the problem is large. Therefore, the unique nature of the crossover operator in the proposed GA makes the algorithm very efficient for large instances of the problem.

The larger the problem is, the more time will be consumed by the GA to reach the termination criterion. However, the difference between the initial solutions and the near-optimal solutions (final results of the GA) becomes much more significant when the size of the problem grows.

In the following section, a comparison between the proposed MILP model and the GA is presented which explains the special characteristics of each of these solution approaches in detail.

5.2 Comparison between the Two Proposed Solution Methods

By comparing the results obtained from the MILP model and those which are achieved using the heuristic method and the GA, it is possible to compare these solution methodologies both from the viewpoint of solution quality and computational efficiency.

A careful observation of the results shows that for small-size problems both methods can work perfectly. The MILP model delivers optimal solutions within really short run times (about 2 minutes) and the GA also generates solutions which are very close to optimal in less than one minute. However, when the size of the problem grows, while the solution is optimal the computational efficiency of the MILP model declines dramatically.

However, the GA preserves both the high solution quality and computational efficiency when the sizes of the problems grow. A comparison of the results for medium-size problems prove that the values of the objective function for GA solutions are very close to the results of CPLEX that generates optimal solutions for the MILP model. Therefore, the proposed GA maintains the high computational efficiency while the quality of the solutions is still satisfactory.

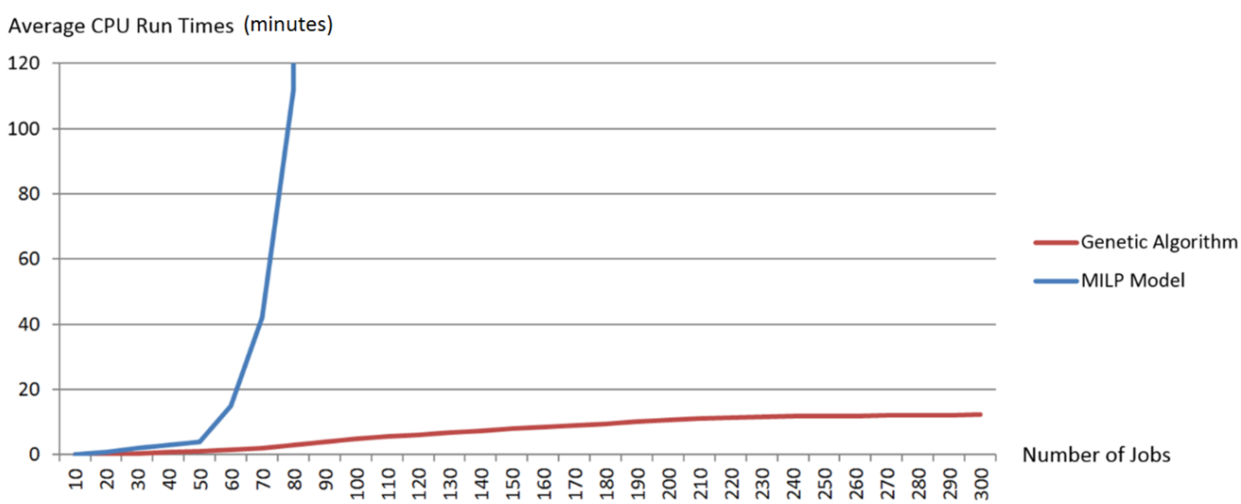

Figure 17. Comparing the MILP model and the GA from the viewpoint of CPU run times

Figure 17 shows the differences in CPU run times for the two proposed methods when they are applied to different sets of problems. The horizontal axis shows the number of jobs while the vertical axis displays the average run times in minutes. The MILP model's efficiency declines dramatically when the number of jobs is more than 70 and it is not useful when the number of jobs goes beyond 100. On the contrary, the GA maintains the high computational efficiency even for problems with up to 300 jobs.

The number of iterations and consequently, the run time of the GA increase when the sizes of the problems grow but this increase is very gradual and mild compared to the severe increase in the run times of the MILP model.

The solution qualities are also compared in Figure 18. The horizontal axis shows the number of jobs and the vertical axis shows the average value of the objective function. As it can be easily observed, the values of the objective functions of the two proposed methods are very close to each other for small and medium-size problems. For large-size problems, only the GA can solve the model. Therefore, there is no basis for comparing the results with optimal solutions. However, observation of the differences between the two methods for small and medium-size problems can somehow imply the high quality of the solutions obtained from the GA even for very large-size problems.

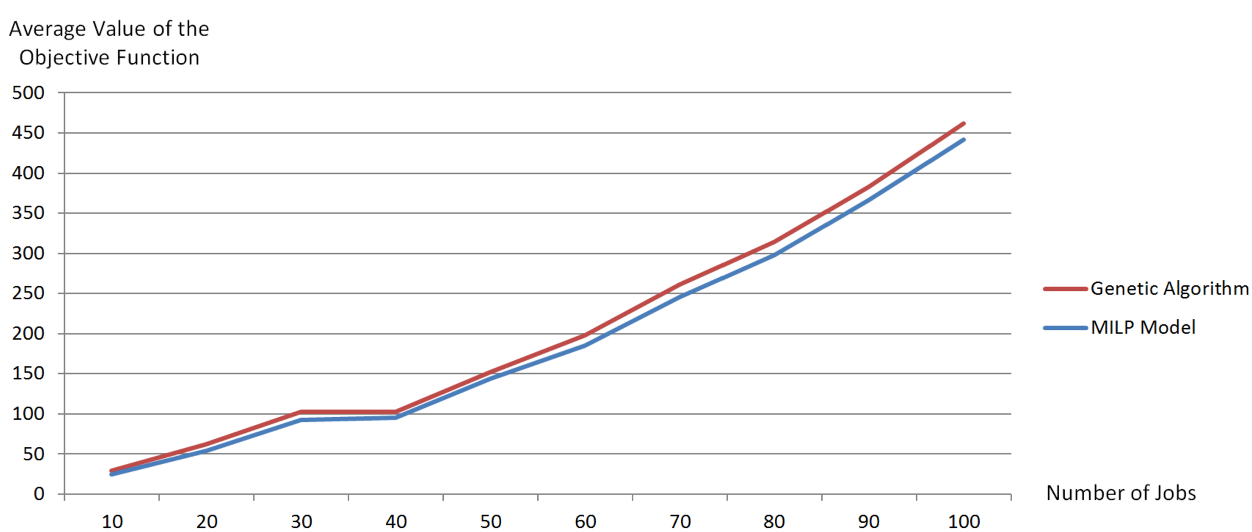

Figure 18. Comparing the average objective function of the MILP model and the GA for different problem sizes

Figure 19 shows the optimality gap of the results obtained from the GA for small and medium-size problems. The horizontal axis shows the number of jobs while the vertical axis demonstrates the gap between the solutions achieved by the GA and the optimal solutions. As it can be observed, the optimality gap reduces when the number of jobs increases.

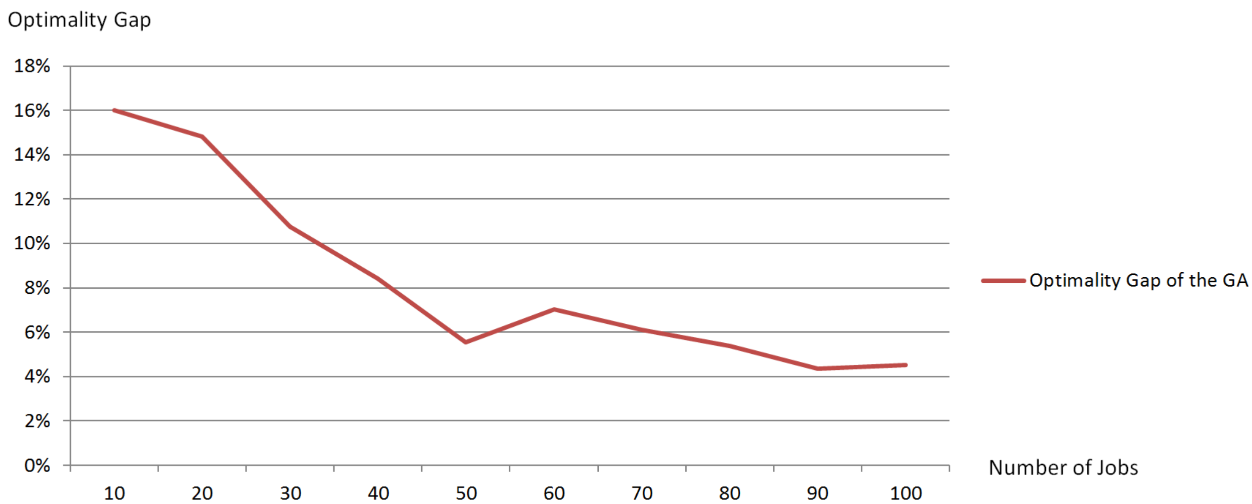

Figure 19. Optimality Gap of the Proposed GA

It can be concluded from Figure 19 that for small-size problems, it's better to use the MILP model because while the computational efficiency is high, the solutions are optimal. Using the GA for small-size problems is not recommended because even though the algorithm delivers the solutions in shorter times, the optimality gap is relatively high (average 12%). In addition, the time difference is not that significant to compromise the solution quality. However, for medium and large-size problems, the average optimality gap is under 5% and the GA outperforms the MILP model from the viewpoint of computational efficiency. Therefore, the GA is recommended for larger instances of the problem.

The SPT-based heuristic algorithm obtains very good solutions which are usually close to the optimal solutions. However, the optimality gap between the SPT-based heuristic and the MILP model depends on the correlation between the processing times and the sizes of the molds. Due to the nature of the objective function of the MILP model, the model tries to maximize the number of jobs in the earlier cure cycles.

$$\text{Min} \sum_{i \in I} \sum_{c=1}^{25} Y(i,c) * c$$

This function is minimized by matching small *c* with the largest $\sum Y(i,c)$ possible, and this is possible by processing as many small jobs (SPT) with small mold sizes first as possible. If small jobs do not have small mold sizes then the gap between SPT and MILP solutions is expected to widen. Therefore, in data sets where processing time and mold size have high positive correlation, SPT is expected to provide close to optimal solutions. In other words, lower the positive correlation wider the gap is expected to be.

Figure 20 shows the optimality gap of the SPT-based heuristic with respect to the correlation coefficient between the processing times of the jobs and the sizes of their molds. The equation for the correlation coefficient is: $\text{Correlation (X, Y)} = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$

Fourteen sets of data with different correlation coefficients are used to solve a medium-size problem (50 jobs).

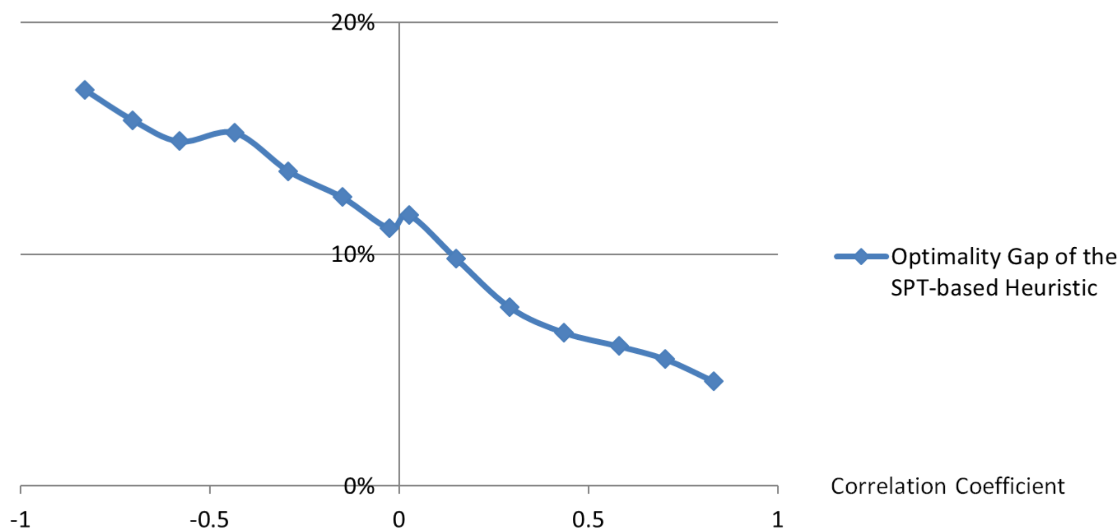

Figure 20. Optimality Gap of the SPT-based Heuristic

As it can be observed in Figure 20, when the correlation between these two parameters is negative the optimality gap is more than 10%. Therefore, for problems in which there is no correlation between the processing times and the sizes of the molds or the correlation is negative, SPT will no longer provide solutions close to the optimal solutions provided by the proposed MILP model.

5.3 Discussion and Implications

Analyzing and comparing the results obtained from the proposed solution methods can result in useful insights into the usage of each of these methods under different circumstances. As discussed in the previous sections, while the MILP model is very useful for solving small-size problems and medium-size problems with up to 70 jobs, its efficiency declines when it's applied to large-size problems.

In the Composite Manufacturing Center of the ABC Company, the number of parts that are manufactured every day is normally more than one hundred. As a matter of fact, the ABC Company is a large plant with a relatively large composite manufacturing system. Therefore, even though the MILP model can be useful for smaller composite manufacturing systems, the ABC Company needs to use a scheduling tool that can address the needs of a higher production rate. For smaller companies, the MILP model delivers optimal solutions in quite short run times.

Larger facilities such as the ABC Company can employ the proposed GA for their daily usage. In the Composite Manufacturing Center of the ABC Company, the scheduling horizon is always the next five days. Thus, it is a rolling horizon in which jobs should be scheduled in a way that all due days are met and the objective function (maximizing the utilization of the autoclaves) is satisfied. This is possible by linking the proposed GA to the MRP system and using it every day.

Suppose the production planner of the Composite Manufacturing Center starts using the GA today. They give a list of jobs to the algorithm as inputs and receive the near-optimal schedule of the jobs for the next five days (day 1 – 5). The next day, the production planner can run the algorithm again but this time the input list excludes the jobs that were scheduled and produced the previous day and includes the jobs for the next five days (day 2 – 6).

Every day, the production planner can run the algorithm with the updated input list and obtain the results. This can guarantee that all the jobs will be scheduled in time and all due dates will be met. This makes the GA a very useful and efficient tool for operational level and daily usage in the shop floor. The costs of implementing such a tool and connecting it to the MRP system is negligible when compared to the contributions it can make to the system.

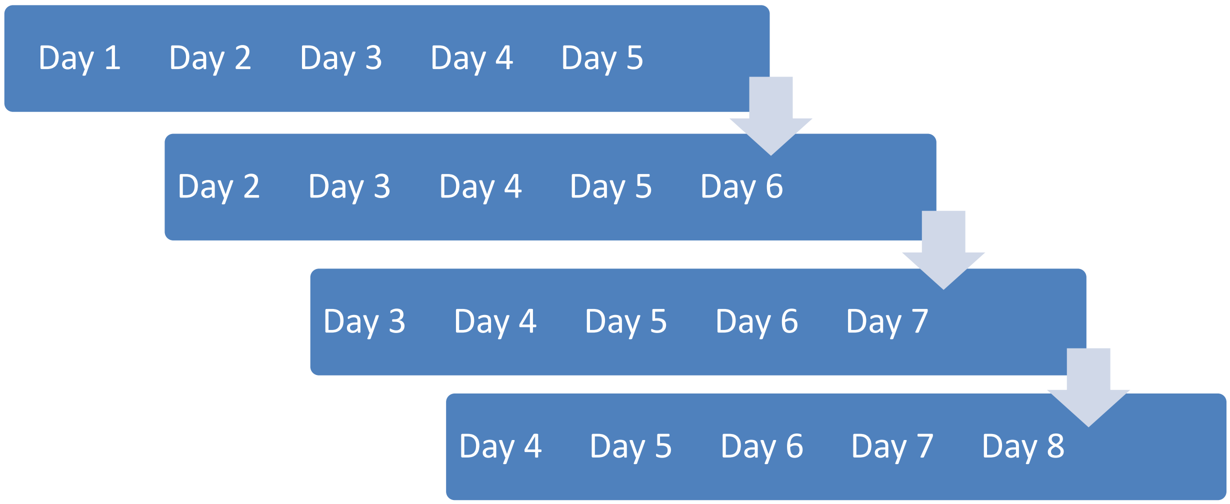

Figure 21. Rolling horizon of the scheduling system in the Composite Manufacturing Center

Both proposed methods, the MILP model and the GA, can be modified and used for composite manufacturing systems with different structures and settings as long as the main elements of a composite manufacturing system (the discrete-processing lay-up cells in the first stage and the parallel batch-processing curing machines in the second stage) exist. The number and variety of lay-up cells (and/or number of work stations in each cell), number of autoclaves and the variety of cure recipes they provide, number and types of molds/tools, and the number and length of working shifts can vary for different cases. In order to adjust the model to new composite manufacturing system settings, it is only required to define the number and type of the resources in each stage.

The models can be extended to consider the characteristics of more complicated structures. The discrete-time nature of the MILP model enables it to be quite flexible where defining new constraints is necessary. The GA also is designed in a way that by modifying the operators and the fitness function, it is possible to use it for other different types of flow shops. Therefore, the proposed methods are applicable not only to the described hybrid flow shop structure but also to other flow shops and even industries other than aerospace and composite manufacturing. In the next section, the possibility of extending the models and defining the proposed methods for various manufacturing structures will be discussed in more details.

6 Conclusions and Future Work

In this chapter, the conclusions of the research carried out in this thesis are presented. Limitations of this work, opportunities for improving it and directions for future research in the area of hybrid flow shops and composite manufacturing systems are also discussed.

6.1 Conclusions

In this research, scheduling a two-stage Hybrid Flow Shop (HFS) with identical, non-identical and unrelated parallel discrete-processing machines in the first stage and non-identical parallel batch-processing machines in the second stage, is studied for the first time.

The special characteristics of this particular type of Hybrid Flow Shop are studied through careful observation of a real case study of the Composite Manufacturing Center at the ABC Company. The required information about the current conditions of this system is gathered during several meetings with industry people responsible for managing this Composite Manufacturing Center. There are various types of constraints within aerospace composite manufacturing systems because of their certain characteristics.

A Time-indexed Mixed Integer Linear Programming model is developed to take all these characteristics into consideration in order to maximize the utilization of the resources in the second stage of this Hybrid Flow Shop. The second stage is composed of the curing autoclaves which are considered enormous capital investment. Therefore, synchronizing the sequence of work done in the lay-up process (first stage) to the autoclaves' cure sequence (second stage) in a way that maximizes the utilization of the autoclaves is of uttermost importance. This is put as the objective function of the proposed MILP model. CPLEX with an underlying branch and bound algorithm is employed to code and solve the proposed model. The results show the high level of flexibility and computational efficiency of the proposed MILP model when applied to small and medium-size problems.

This optimized schedule of the jobs obtained from the proposed MILP model also affects the tooling cycle time which is one of the most serious issues addressed in this research. The tooling cycle starts from the tools storage area and includes the lay-up cells, the cure process, the unmolding and tool preparation processes. While a part is within the tooling cycle, its mold (tool) cannot be used for other parts due to the limited number of molds for each part type.

Reducing the time that each part stays in this tooling cycle can open up some extra capacity for other parts to be produced in this manufacturing system.

So as to be able to solve large problems as well, a heuristic algorithm is designed based on a careful scrutiny of the behavior of the MILP model. In order to enhance the capabilities of this heuristic, a number of dispatching rules are also included in it. This novel heuristic generates eight feasible initial solutions. A GA with a novel crossover operation is also introduced that uses the outputs of the aforementioned heuristic method as its initial population and improves them through its stochastic iterations to obtain near-optimal solutions.

An analysis of the results show the efficiency of the proposed GA as well as the high quality of the solutions it generates compared to the optimal solutions obtained from the MILP model. It has been shown in the thesis that optimizing the schedule for this specific type of hybrid flow shop can reveal the level of utilization of the capacity of the resources in both stages of the studied hybrid flow shop. In fact, sensitivity analysis of the MILP model can show how well the resources of this system are utilized and where the possible opportunities for improvement lie.

The physical capacity constraint of the buffer in this hybrid flow shop is one aspect of this research which has not been studied enough in the literature of scheduling problems. Therefore, this very realistic constraint is taken into consideration in the proposed MILP model and GA. The results show that in order not to violate these constraints, jobs should start being processed in a lay-up cell as close as possible to the start time of their respective cure process. This way, the jobs will spend the minimum amount of time in the buffer and the limited physical capacity will be preserved for other parts.

It has also been shown in this research that GA can be used as a very strong and efficient tool instead of exact solution methodologies, in cases where exact methods cannot achieve solutions in reasonable CPU times. The comparison of the results shows small differences between the objective function of the proposed GA and the MILP model. It also shows that applying a proper method to generate initial feasible solutions with good qualities (i.e. the ones generated by the heuristic algorithm) instead of generating random initial population, can strengthen the GA significantly and lead to near-optimal solutions in shorter times.

Another important conclusion of this research is that the length of the scheduling horizon directly affects the lead times. The longer the scheduling horizon is supposed, the longer the lead times and the larger the amount of WIP and finished goods inventories will be. When the production planner looks ahead to the needs of the customers in the near future based on the data from an MRP system, they may include some products in today's schedule which are not needed today. In fact, if the scheduling horizon is 10 days, for example, some of the parts which are decided to be produced today may be needed after a week (not today). This creates a lot of WIP and finished goods inventories and consequently lengthens the lead times. On the other hand, reducing the scheduling horizon makes the production planner unable to have enough input jobs to form the optimal batches for the second stage of the hybrid flow shop. Therefore, selecting a proper horizon length is a trade-off that should be carefully made in order to maximize the efficiency of this scheduling system.

The first contribution of this research is developing a linear programming model for this complex type of hybrid flow shop which is unique to composite manufacturing systems. The proposed model can be used for other hybrid flow shops with similar structure or for other industries provided that the main elements of the HFS remain the same.

Another contribution of this research is taking into account the tools constraint and the tooling cycle. Even though resources constraints have always been regarded as integral parts of the scheduling problems, these resources are typically defined as fixed machines in different stages. In this research, however, in addition to those resources, there are another type of portable resources (tools) that move through the stages but still need to be planned and allocated to the tasks. The scheduling models proposed in this research take into account both these resource types.

The physical capacity limitation of the buffer is another constraint that has not been tackled by the researchers in the area of Hybrid Flow Shop scheduling. This research addresses this gap in the literature as well and provides a flexible MILP model which is open to further modifications in case of the need for adding extra constraints in the future.

The heuristic algorithm which is designed for generating initial solutions is a totally novel and unique algorithm which is considered a significant contribution of this thesis.

Last but not least, the GA proposed in this research is new from various points of view. The fitness function is designed in a special way that considers different constraints of the problem and minimizes the chance of survival for infeasible solutions in each generation. The crossover operator of the proposed GA is a novel crossover which is designed specifically for the case of this particular hybrid flow shop and is proven to be very effective and unique to the case of composite manufacturing systems.

6.2 Limitations and Future Research

There are a number of ways for improving the research done in this thesis. First of all, the MILP model can be extended and improved in several ways. The objective function of the model can be expanded to consider other factors such as minimization of make-span or optimization of the resource utilization in the first stage as well as the second stage. The constraints regarding the stacking of the products inside the autoclaves can be identified and added to the model.

In order to solve the large instances of the problem using the MILP model, other solution methodologies such as Column Generation or Lagrangian Relaxation can be used. The use of relaxation methods or other algorithms for solving large MILP models are promising research avenues in the area of hybrid flow shop scheduling. Another approach could be trying to develop a new continuous-time MILP model considering all the new constraints in the problem (i.e. tooling constraint, limited buffer, etc.). While the flexibility of the model may be reduced, its efficiency can be boosted due to the reduced number of variables and smaller size of the model.

An extensive sensitivity analysis of the optimization model can be conducted in order to determine the optimal number of work stations in each lay-up cell in the first stage as well as the optimal number of required autoclaves in the second stage. Similarly, a simulation model can be developed to provide some visibility on the resources capacity usage within the studied composite manufacturing systems. One of the future objectives could be leveling the load of the work throughout the working shifts. To do so, the mentioned methods to determine the optimal number of resources in each stage could be of valuable use. Making a smooth flow of products within the manufacturing system will be easier by adjusting the required number of work stations in the first stage and optimal number of autoclaves in the second stage.

In the current research, it is assumed that the schedule of the autoclaves in the second stage of the HFS is fixed. In the future research, one can relax this assumption and try to look at the schedule of the autoclaves (operation of the various cure recipes) as another variable of the problem. In addition, the scheduling horizon is assumed to be fixed (five days) which is subject to change in future models.

Future research can also be focused on improving the proposed solution methodologies. One possible way is to elaborate the proposed heuristic method and re-design it in a way that could generate better solutions. Developing a new GA by defining improved encoding/decoding schemes for chromosomes or introducing other novel operators and/or fitness function are other ways to advance the research in this area. Furthermore, other meta-heuristic methods (e.g. PSO, Ant Colony Optimization, Neural Networks, etc.) can be used to propose new solution methods. Comparing the efficiency of these different methods with regard to quality of the solutions they generate could be a very interesting topic for future research.

The current GA can also be improved from different points of view. In order to set the control parameters of the GA more effectively, advanced tools and methods (e.g. Taguchi Experimental Design, Response Surface Methodology, etc.) can be used. This will probably boost the performance of the GA and lead to higher quality of the solutions in shorter run times.

Another interesting avenue of research for improving the proposed GA is to develop similar GAs with different crossover operators and compare their efficiency. This way the proposed crossover operator can be compared to the conventional crossover operators available in the literature of GAs and its advantages and disadvantages will be revealed. Also, these kinds of studies can potentially prepare the ground for introducing other new crossover operators that can, in turn, enhance the performance of the solution method.

Finally, other types of composite manufacturing systems (in aerospace or any other industry) with different structures can be studied and compared to the one which is studied in this research.

Bibliography

Agnetis, A., Pacciarelli, D., & Rossi, F. (1997). Batch scheduling in a two-machine flow shop with limited buffer. *Discrete Applied Mathematics*, *72*(3), 243-260.

Ahmadi, J. H., Ahmadi, R. H., Dasu, S., & Tang, C. S. (1992). Batching and scheduling jobs on batch and discrete processors. *Operations Research*, 40(4), 750-763.

Amin-Naseri, M. R., & Beheshti-Nia, M. A. (2009). Hybrid flow shop scheduling with parallel batching. *International Journal of Production Economics*, 117(1), 185-196.

Baker, K. R., & Schrage, L. E. (1978). Finding an optimal sequence by dynamic programming: an extension to precedence-related tasks. *Operations Research*, *26*(1), 111-120.

Balasubramanian, H., Mönch, L., Fowler, J., & Pfund, M. (2004). Genetic algorithm based scheduling of parallel batch machines with incompatible job families to minimize total weighted tardiness. *International Journal of Production Research*, 42(8), 1621-1638.

Behnamian, J., & Ghomi, S. F. (2011). Hybrid flow-shop scheduling with machine and resource-dependent processing times. *Applied Mathematical Modelling*, 35(3), 1107-1123.

Bellanger, A., & Oulamara, A. (2009). Scheduling hybrid flow-shop with parallel batching machines and compatibilities. *Computers & Operations Research*, 36(6), 1982-1992.

Bhatnagar, R., Chandra, P., Loulou, R., & Qiu, J. (1999). Order release and product mix coordination in a complex PCB manufacturing line with batch processors. *International Journal of Flexible Manufacturing Systems*, 11(4), 327-351.

Blomer, F., & Gunther, H. O. (2000). LP-based heuristics for scheduling chemical batch processes. *International Journal of Production Research*, 38(5), 1029-1051.

Brucker, P., Gladky, A., Hoogeveen, H., Kovalyov, M. Y., Potts, C., Tautenhahn, T., & Van De Velde, S. (1998). Scheduling a batching machine. *Journal of Scheduling*, 1: 31-54.

Campbell, H. G., Dudek, R. A., & Smith, M. L. (1970). A heuristic algorithm for the n job, m machine sequencing problem. *Management science*, *16*(10), B-630.

Chandra, P., & Gupta, S. (1997). Managing batch processors to reduce lead time in a semiconductor packaging line. *International Journal of Production Research*, 35(3), 611-633.

Damodaran, P., & Srihari, K. (2004). Mixed integer formulation to minimize make-span in a flow shop with batch processing machines. *Mathematical and Computer Modelling*, 40(13), 1465-1472.

Damodaran, P., Manjeshwar, P. K., & Srihari, K. (2006). Minimizing make-span on a batch-processing machine with non-identical job sizes using genetic algorithms. *International journal of production economics*, *103*(2), 882-891.

Damodaran, P., Vélez-Gallego, M. C., & Maya, J. (2011). A GRASP approach for make-span minimization on parallel batch processing machines. *Journal of Intelligent Manufacturing*, 22(5), 767-777.

Danneberg, D., Tautenhahn, T., & Werner, F. (1999). A comparison of heuristic algorithms for flow shop scheduling problems with setup times and limited batch size. *Mathematical and Computer Modelling*, 29(9), 101-126.

Floudas, C. A., & Lin, X. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers & Chemical Engineering*, 28(11), 2109-2129.

Floudas, C. A., & Lin, X. (2005). Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Annals of Operations Research*, 139(1), 131-162.

Gicquel, C., Hege, L., Minoux, M., & Van Canneyt, W. (2012). A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints. *Computers & Operations Research*, 39(3), 629-636.

Gong, H., Tang, L., & Duin, C. W. (2010). A two-stage flow shop scheduling problem on a batching machine and a discrete machine with blocking and shared setup times. *Computers & Operations Research*, 37(5), 960-969.

Guinet, A. G. P., & Solomon, M. M. (1996). Scheduling hybrid flow-shops to minimize maximum tardiness or maximum completion time. *International Journal of Production Research*, 34(6), 1643-1654.

Heizer, J. H., Griffin, P., & Render, B. (2013). Operations management. *Supply-chain management*, *434*(436), 438-450.

Hoogeveen, H., & van de Velde, S. (1998). Scheduling by positional completion times: Analysis of a two-stage flow shop problem with a batching machine. *Mathematical Programming*, 82(1-2), 273-289.

Jenabi, M., Ghomi, S. F., Torabi, S. A., & Karimi, B. (2007). Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines. *Applied Mathematics and Computation*, 186(1), 230-245.

Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. (2009). A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Computers & Operations Research*, *36*(2), 358-378.

Kim, B., & Kim, S. (2002). Application of genetic algorithms for scheduling batch-discrete production system. *Production Planning & Control*, 13(2), 155-165.

Klemmt, A., Horn, S., Weigert, G., & Hielscher, T. (2008, December). Simulations-based and solver-based optimization approaches for batch processes in semiconductor manufacturing. In Simulation Conference, 2008. WSC 2008. Winter (pp. 2041-2049). IEEE.

Klemmt, A., & Mönch, L. (2012, December). Scheduling jobs with time constraints between consecutive process steps in semiconductor manufacturing. In Proceedings of the Winter Simulation Conference (p. 194). Winter Simulation Conference.

Klemmt, A., Weigert, G., Almeder, C., & Mönch, L. (2009, December). A comparison of MIP-based decomposition techniques and VNS approaches for batch scheduling problems. In Simulation Conference (WSC), Proceedings of the 2009 Winter (pp. 1686-1694). IEEE.

Kovalyov, M. Y., Potts, C. N., & van Wassenhove, L. N. (1993). *Single machine scheduling with set-ups to minimize the number of late items: algorithms, complexity and approximation*. INSEAD.

Leisten, R. (1990). Flow-shop sequencing problems with limited buffer storage. *The International Journal of Production Research*, *28*(11), 2085-2100.

Li, L., Qiao, F., & Wu, Q. (2008). ACO-based scheduling of parallel batch processing machines with incompatible job families to minimize total weighted tardiness. In Ant Colony Optimization and Swarm Intelligence (pp. 219-226). Springer Berlin Heidelberg.

Liker, J. K. (2005). *The Toyota Way*. Esensi.

Lin, B. M., & Cheng, T. E. (2005). Two-machine flow-shop batching and scheduling. *Annals of Operations Research*, 133(1-4), 149-161.

Liu, B., Wang, L., & Jin, Y. H. (2008). An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, *35*(9), 2791-2806.

Liu, B., Wang, L., Liu, Y., Qian, B., & Jin, Y. H. (2010). An effective hybrid particle swarm optimization for batch scheduling of polypropylene processes. *Computers & chemical engineering*, 34(4), 518-528.

Low, C. (2005). Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. *Computers & Operations Research*, 32(8), 2013-2025.

Lubin, G. (2013). *Handbook of composites*. Springer Science & Business Media.

Luo, H., Huang, G. Q., Feng Zhang, Y., & Yun Dai, Q. (2011). Hybrid flow-shop scheduling with batch-discrete processors and machine maintenance in time windows. *International Journal of Production Research*, 49(6), 1575-1603.

Luo, H., Huang, G. Q., Zhang, Y., Dai, Q., & Chen, X. (2009). Two-stage hybrid batching flow-shop scheduling with blocking and machine availability constraints using genetic algorithm. *Robotics and Computer-Integrated Manufacturing*, 25(6), 962-971.

MacCarthy, B. L., & Liu, J. (1993). Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *International Journal of Production Research*, *31*(1), 59–79.

Mathirajan, M., & Sivakumar, A. I. (2006). A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. *The International Journal of Advanced Manufacturing Technology*, *29*(9-10), 990-1001.

Mathirajan, M., Sivakumar, A. I., & Kalaivani, P. (2004). An efficient simulated annealing algorithm for scheduling burn-in oven with non-identical job sizes. *International Journal of Applied Management and Technology*, 2(2), 117-138.

Méndez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, 30(6), 913-946.

Méndez, C. A., Henning, G. P., & Cerdá, J. (2001). An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flow-shop batch facilities. *Computers & Chemical Engineering*, 25(4), 701-711.

Mönch, L., & Almeder, C. (2009). Ant colony optimization approaches for scheduling jobs with incompatible families on parallel batch machines. Proceedings MISTA, 106-114.

Mönch, L., Fowler, J. W., Dauzère-Pérès, S., Mason, S. J., & Rose, O. (2011). A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *Journal of Scheduling*, 14(6), 583-599.

Mönch, L., Zimmermann, J., & Otto, P. (2006). Machine learning techniques for scheduling jobs with incompatible families and unequal ready times on parallel batch machines. *Engineering Applications of Artificial Intelligence*, 19(3), 235-245.

Monma, C. L., & Potts, C. N. (1989). On the complexity of scheduling with batch setup times. *Operations Research*, *37*(5), 798-804.

Monma, C. L., & Potts, C. N. (1993). Analysis of heuristics for preemptive parallel machine scheduling with batch setup times. *Operations Research*, *41*(5), 981-993.

Myerson, P. (2012). *Lean supply chain and logistics management*. McGraw Hill Professional.

Nawaz, M., Enscore, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91-95.

Ng, C. T., Cheng, T. E., Yuan, J. J., & Liu, Z. H. (2003). On the single machine serial batching scheduling problem to minimize total completion time with precedence constraints, release dates and identical processing times. *Operations Research Letters*, *31*(4), 323-326.

Oĝuz, C., & Ercan, M. F. (2005). A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *Journal of Scheduling*, *8*(4), 323-351.

Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, *12*(4), 417-431.

Oulamara, A., Finke, G., & Kuiteing, A. K. (2009). Flow-shop scheduling problem with a batching machine and task compatibilities. *Computers & Operations Research*, 36(2), 391-401.

Ovacik, I. M., & Uzsoy, R. (1995). Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times. *International journal of production research*, 33(11), 3173-3192.

Petridis, V., Kazarlis, S., & Bakirtzis, A. (1998). Varying fitness functions in genetic algorithm constrained optimization: the cutting stock and unit commitment problems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *28*(5), 629-640.

Pinedo, M. L. (2012). *Scheduling: theory, algorithms, and systems*. Springer Science & Business Media.

Pinto, J. M., & Grossmann, I. E. (1995). A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Industrial & Engineering Chemistry Research*, 34(9), 3037-3051.

Potts, C. N., & Kovalyov, M. Y. (2000). Scheduling with batching: a review. *European journal of operational research*, 120(2), 228-249.

Potts, C. N., Strusevich, V. A., & Tautenhahn, T. (2001). Scheduling batches with simultaneous job processing for two-machine shop problems. *Journal of Scheduling*, 4(1), 25-51.

Pranzo, M. (2004). Batch scheduling in a two-machine flow shop with limited buffer and sequence independent setup times and removal times. *European Journal of Operational Research*, *153*(3), 581-592.

Raghavan, N. S., & Venkataramana, M. (2006, October). Scheduling parallel batch processors with incompatible job families using ant colony optimization. In Automation Science and Engineering, 2006. CASE'06. IEEE International Conference on (pp. 507-512). IEEE.

Reichelt, D., & Mönch, L. (2006). Multiobjective scheduling of jobs with incompatible families on parallel batch machines. In Evolutionary computation in combinatorial optimization (pp. 209-221). Springer Berlin Heidelberg.

Reisman, A., Kumar, A., & Motwani, J. (1997). Flow-shop scheduling/sequencing research: A statistical review of the literature, 1952-1994. *Engineering Management, IEEE Transactions on*, *44*(3), 316-329.

Rose-Anderssen, C., Baldwin, J. S., Ridgway, K., Allen, P. M., & Varga, L. (2008). Aerospace supply chains as evolutionary networks of activities: innovation via risk-sharing partnerships. *Creativity and innovation management*, *17*(4), 304-318.

Rossi, A., Pandolfi, A., & Lanzetta, M. (2014). Dynamic set-up rules for hybrid flow shop scheduling with parallel batching machines. *International Journal of Production Research*, 52(13), 3842-3857.

Rossi, A., Puppato, A., & Lanzetta, M. (2013). Heuristics for scheduling a two-stage hybrid flow shop with parallel batching machines: application at a hospital sterilization plant. *International Journal of Production Research*, 51(8), 2363-2376.

Ruiz, R., Şerifoğlu, F. S., & Urlings, T. (2008). Modeling realistic hybrid flexible flow-shop scheduling problems. Computers & Operations Research, 35(4), 1151-1175.

Sawik, T. J. (1995). Scheduling flexible flow lines with no in-process buffers. *The International Journal of Production Research*, 33(5), 1357-1367.

Sawik, T. (2000). MIP for scheduling flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modelling*, 31(13), 39-52.

Sawik, T. (2002). An exact approach for batch scheduling in flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modelling*, 36(4), 461-471.

Sawik, T. (2005). Integer programming approach to production scheduling for make-to-order manufacturing. *Mathematical and Computer Modelling*, 41(1), 99-118.

Sawik, T. (2006). Hierarchical approach to production scheduling in make-to-order assembly. *International Journal of Production Research*, 44(4), 801-830.

Shen, L., & Buscher, U. (2012). Solving the serial batching problem in job shop manufacturing systems. *European Journal of Operational Research*, *221*(1), 14-26.

Shiau, D. F., & Huang, Y. M. (2012). A hybrid two-phase encoding particle swarm optimization for total weighted completion time minimization in proportionate flexible flow shop scheduling. *The International Journal of Advanced Manufacturing Technology*, *58*(1-4), 339-357.

Sotskov, Y. N., Tautenhahn, T., & Werner, F. (1996). Heuristics for permutation flow shop scheduling with batch setup times. *Operations-Research-Spektrum*, 18(2), 67-80.

Stefansson, H., Sigmarsdottir, S., Jensson, P., & Shah, N. (2011). Discrete and continuous time representations and mathematical models for large production scheduling problems: A case study from the pharmaceutical industry. *European Journal of Operational Research*, 215(2), 383-392.

Su, L. H. (2003). A hybrid two-stage flow-shop with limited waiting time constraints. *Computers & Industrial Engineering*, 44(3), 409-424.

Sung, C. S., & Kim, Y. H. (2003). Minimizing due date related performance measures on two batch processing machines. *European Journal of Operational Research*, 147(3), 644-656.

Sung, C. S., Kim, Y. H., & Yoon, S. H. (2000). A problem reduction and decomposition approach for scheduling for a flow-shop of batch processing machines. *European Journal of Operational Research*, 121(1), 179-192.

Sung, C. S., & Min, J. I. (2001). Scheduling in a two-machine flow-shop with batch processing machine (s) for earliness/tardiness measure under a common due date. *European Journal of Operational Research*, 131(1), 95-106.

Sung, C. S., & Yoon, S. H. (1997). Minimizing maximum completion time in a two-batch-processing-machine flow-shop with dynamic arrivals allowed. *Engineering Optimization*, A35, 28(3), 231-243.

Tang, L., & Liu, P. (2009). Minimizing make-span in a two-machine flow-shop scheduling with batching and release time. *Mathematical and Computer Modelling*, *49*(5), 1071-1077.

Tang, L. X., & Xuan, H. (2006). Lagrangian relaxation algorithms for real-time hybrid flow-shop scheduling with finite intermediate buffers. *Journal of the Operational Research Society*, *57*(3), 316-324.

Tavakkoli-Moghaddam, R., Heydar, M., & Mousavi, S. M. (2010). A hybrid genetic algorithm for a bi-objective scheduling problem in a flexible manufacturing cell. *International Journal of Engineering-Transactions A: Basics*, *23*(3&4), 235.

Tasgetiren, M. F., Sevkli, M., Liang, Y. C., & Gencyilmaz, G. (2004). Particle swarm optimization algorithm for permutation flow-shop sequencing problem. In *Ant Colony Optimization and Swarm Intelligence* (pp. 382-389). Springer Berlin Heidelberg.

Townsend, W. (1977). Note-Sequencing n Jobs on m Machines to Minimise Maximum Tardiness: A Branch-and-Bound Solution. *Management Science*, *23*(9), 1016-1019.

Vepsalainen, A. P., & Morton, T. E. (1987). Priority rules for job shops with weighted tardiness costs. *Management science*, *33*(8), 1035-1047.

Voss, S., & Witt, A. (2007). Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application. *International journal of production economics*, 105(2), 445-458.

Wang, L., & Li, D. (2002). A scheduling algorithm for flexible flow shop problem. In Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on (Vol. 4, pp. 3106-3108). IEEE.

Wang, X., & Tang, L. (2009). A tabu search heuristic for the hybrid flow-shop scheduling with finite intermediate buffers. *Computers & Operations Research*, 36(3), 907-918.

Wang, J. T., Chern, M. S., & Yang, D. L. (2001). A two-machine multi-family flow-shop scheduling problem with two batch processors. *Journal of the Chinese Institute of Industrial Engineers*, 18(3), 77-85.

Wang, I. L., Yang, T., & Chang, Y. B. (2012). Scheduling two-stage hybrid flow shops with parallel batch, release time, and machine eligibility constraints. *Journal of Intelligent Manufacturing*, 23(6), 2271-2280.

Wang, L., Zhang, L., & Zheng, D. Z. (2006). An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, *33*(10), 2960-2971.

Wisner, J., Tan, K. C., & Leong, G. (2015). *Principles of supply chain management: a balanced approach*. Cengage Learning.

Womack, J. P., & Jones, D. T. (2010). *Lean thinking: banish waste and create wealth in your corporation*. Simon and Schuster.

Xuan, H., & Tang, L. (2007). Scheduling a hybrid flow-shop with batch production at the last stage. *Computers & Operations Research*, 34(9), 2718-2733.

Yimer, A. D., & Demirli, K. (2009). Fuzzy scheduling of job orders in a two-stage flowshop with batch-processing machines. International journal of approximate reasoning, 50(1), 117-137.

Yuan, J. J., Lin, Y. X., Cheng, T. C. E., & Ng, C. T. (2007). Single machine serial-batching scheduling problem with a common batch size to minimize total weighted completion time. *International Journal of Production Economics*, *105*(2), 402-406.

Yurtsever, T., Kutanoglu, E., & Johns, J. (2009, December). Heuristic based scheduling system for diffusion in semiconductor manufacturing. In Winter Simulation Conference (pp. 1677-1685). Winter Simulation Conference.

Zandieh, M., Mozaffari, E., & Gholami, M. (2010). A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. *Journal of Intelligent Manufacturing*, *21*(6), 731-743.

Ziaeifar, A., Tavakkoli-Moghaddam, R., & Pichka, K. (2012). Solving a new mathematical model for a hybrid flow shop scheduling problem with a processor assignment by a genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 61(1-4), 339-349.

Zdrzałka, S. (1991). Approximation algorithms for single-machine sequencing with delivery times and unit batch set-up times. *European Journal of Operational Research*, *51*(2), 199-209.