Application of Metaheuristics in Scheduling Continuous/Semi-continuous Process Industries and a Case Study

by

Mohammad Moein Jalalian

A Thesis

presented to

The University of Guelph

In partial fulfilment of requirements

for the degree of

Master of Applied Science

in

Engineering

Guelph, Ontario, Canada

ABSTRACT

APPLICATION OF METAHEURISTICS IN SCHEDULING CONTINUOUS/SEMI-CONTINUOUS PROCESS INDUSTRIES AND A CASE STUDY

Mohammad Moein Jalalian
University of Guelph, 2018

Advisor:
Professor F. M. Defersha

In today's competitive industry, scheduling plays a significant role in improving the efficiency of manufacturing systems. Hence, many scholars and practitioners have been researching to enhance the quality of scheduling methods. In this research, the focus is on solving a real-world scheduling problem in the food industry which was previously dealt with a very time-consuming manual method without high-quality solutions. The problem is to find the best schedule for producing multiple products on multiple machines in a semi-continuous manufacturing system. Having a continuous section in the system makes scheduling too complicated than the manual method could deal with properly. So, similar to many scheduling problems, in this thesis, metaheuristics (GA and MPSA) are applied to the problem in order to address the defects of the manual method. Selected methods show promising results and performance against the manual method used before. Statistical analysis shows better performance of the genetic algorithm while the other method is more robust to the selected parameters.

Keywords: *Continuous Manufacturing; Food Industry; Genetic Algorithm; Multiple-Path Simulated Annealing.*

Dedicated to my mother and father.

ACKNOWLEDGEMENTS

I am wholeheartedly grateful to my wonderful advisor, Professor Fantahun Defersha for supporting and providing me with everything I needed for preparing this thesis. Without his kind support, invaluable advice, endless patience, and extensive knowledge, I would never have been able to finish this thesis. I will always remember his caring attitude for making progress in my knowledge under his supervision.

My warmest thanks go to my beloved parents for their limitless support and exemplary patience. Every success I have made throughout my life has become possible by their encouragement and affection. I would also like to express my special thank to my lovely sister Matin and her husband Hamed for their supports during my stay in Canada. I am always indebted to them for their kind helps. I am also deeply appreciated to my sister Mona for her everlasting love and support.

Above all, I am always thankful to Almighty God, the most compassionate and the most merciful, whose many blessings have made me who I am today.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

α	Significance level
γ	Cooling Coefficient
BS	Best Solution found so far
E	Energy Function
F_i	Factor of product i
F_{max}	Maximum acceptable value of factors
F_{min}	Minimum acceptable value of factors
$F\prime$	New value of F after perturbation
k	index of
m	Mutation probability index
M_i	Machine index
n	Iteration counter , n = 1,2,. . . , N maximum number of the iterations in each search path.
P_0	Index of initial population
P_i	Products index
P_S	Step Factor Probability
Q_i	Quantity needed to be produced for each product
Q	Maximum number of iteration in each temperature
rand()	Randomly generated number
r	Index of cross-over probability
S	Population size
S_{max}	Maximum step that can be taken down or up
T_0	Initial temperature

T_{min}	Minimum temperature
TRG_{imax}	Maximum target of product i
TRG_{imin}	Minimum target of product i

LIST OF ACRONYMS

ACO	Ant Colony Optimization
ANOVA	Analysis of Variance
BP	Batch Production
BCO	Bee Colony Optimization
BB	Branch and Bound
BOM	Bill of Materials
CAD	Computer-Aided Design
CP	Continuous Production
EA	Evolutionary Algorithm
FP	Flow Production
GA	Genetic Algorithm
GP	Genetic Programming
GEP	Generational Evolutionary Programming
HGA	Hybrid Genetic Algorithm
JP	Job Production
LP	Linear Programming
MP	Mass Production
MIP	Mixed Integer Programming
MILP	Mixed Integer-Linear Programming
MTO	Make-To-Order
MTS	Make-To-Stock
MPSA	Multiple-Path Simulated Annealing
NSGA-II	Nondominated Sorting Genetic Algorithm

PSO	Particle Swarm Optimization
PR	Pareto Ranking
SAGA	Structure Adapted Genetic Algorithm
SA	Simulated Annealing
SCP	Semi-Continuous Production
SSPC	Systematic Single Point Crossover
SSEP	Steady State Evolutionary Programming
TS	Tabu Search

Chapter 1

Introduction

Throughout the history, production systems have developed dramatically in all aspects. First manufacturing equipment was used in the middle of the eighteenth century; however, in late 1800s factories with small production systems started to be concerned about the productivity of their resources. The chart proposed by Henry Gantt in 1916 played a significant role in the rise of scheduling methods from years until the early 1950's when scheduling methods were strengthened by computers (Herrmann, 2006). Scheduling is the most typical common point between business and manufacturing systems. Several factors in production systems are evaluated by schedulers based on the requirement of the business in order to keep costs down while operations are done according to the budget (Bodington, 1995). Scheduling methodologies are to allocate limited resources of companies such as machines and workers to the tasks in such a way that the predefined objective of the plant is optimized. The objective could be minimizing the completion time of all activities, or minimizing the number of activities that are done after due dates (Pinedo, 2012). One of the factors that must be considered before making any decision about the scheduling methods is the nature of industries. Considering the products produced in manufacturing systems, two types of industries are distinguished: discrete product industries and process industries.

Several factors make these two types of industries different such as:

- In process manufacturing, products are made using various formulations and multiple recipes while in discrete product systems multilevel Bills of Materials (BOM) are used to produce final products.
- In discrete manufacturing, parts may be taken apart to be used on another product if need be, whereas in process industries it is impossible to take apart an ingredient from a finished product.
- Discrete manufacturers produce or assemble parts that are distinguishable as a distinct product (e.g., airplanes, smartphones, automobiles) in contrast with process industries that the materials are in bulk quantities, such as pharmaceuticals, beverages, and refineries.

Other than the type of products, there exist other factors which should be taken into account in order to find out the best scheduling approach for the manufacturing systems. In the following, first the nature of different production industries are classified in more details in section 1.1 and then in section 1.2 the classification of scheduling problems based on the time scale is presented.

1.1. Typology of Manufacturing Systems

There are various types of production which can be categorized into three or four larger classes based on the level of production and diversity of products. As it is depicted in figure 1.1, these four types are job production, batch production, continuous production, and mass production which is classified as a subset of continuous production (Roy, 2007).

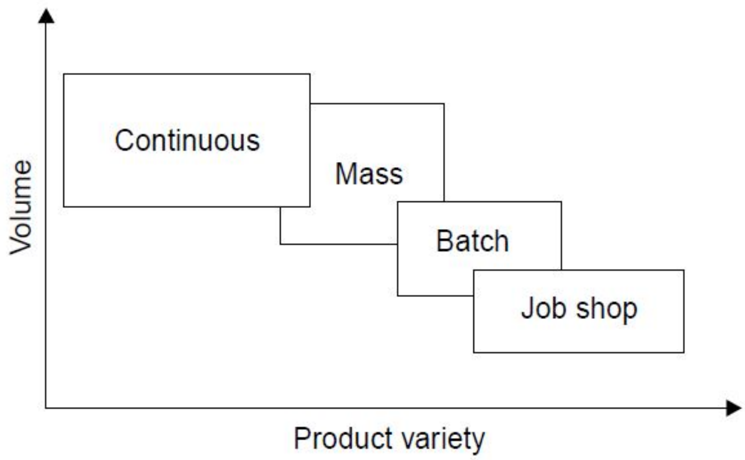


Figure 1.1: Different Types of Production Systems Roy (2007)

1.1.1. Job Production (JP)

JP is the oldest type of production which involves producing small-scale production and used for custom or individual requirements of customers. This system has a lot of flexibility of operation, and therefore needs general purpose machines. This flexibility often prevents using automated manufacturing systems; however computer-aided design(CAD) is used. In general, this system can be used for three different situations:

(a) A limited number of products manufactured only once.

(b) A limited number of products are manufactured whenever the needed.

(c) A limited number of products are manufactured periodically at specific times.

One of the advantages of this system is that the reduction in demand cannot lead to failure in the plant since changeovers can be done easily. However, production of a variety of products with different machines requires labors with multiple skill which increases the labor cost (Roy, 2007). Plants and shops that use this type of production systems are often called job shops. In a job shop jobs often include some operations that need to be done on single or multiple machines. The route through which each job has to follow is different in another job.

Several operations of different jobs have to be scheduled such that the objective criteria such as minimization of makespan or the number of late jobs are met. A particular case of job shops are flow shops where the routing of all jobs are the same, and all jobs go through the same sequence of machines to be operated (Pinedo, 2012).

1.1.2. Batch Production (BP)

Batch production falls between mass and job production systems since various products are produced in higher amounts than job shops. This system is typically used in medium-size plants with a capacity of higher than demand. Generally, two or more types of products are intermittently manufactured in batches of same items with the size that can range from one to as many as thousands units. The number of processing operations is often large as well as routing complexity (Rippin *et al.*, 1991). So, three different systems are distinguished for batch production:

(a) A batch of products are produced only once.

(b) A batch of items are manufactured at irregular intervals when it is needed.

(c) A batch of products are produced periodically at predefined time intervals to meet the continuous demand.

One of the advantages of this system is its flexibility for changeovers between different jobs with no additional costs and at low risk of loss in the production system. However, this variability of jobs needs specially designed jigs and fixtures that could be costly (Roy, 2007).

Both discrete product manufacturing systems and process industries may adopt this type of production in different circumstances. In discrete product industries, a batch contains a limited quantity of parts which are usually processed one at a time and not all together. In contrast, in process industries, the concept of batch production means that finite amount of raw materials which are either in liquid or bulk form, go through the production line and the process is done for all of them simultaneously as a unit (Groover, 2008). Differences between discrete product and process industries in this type of production are shown in figure 1.2.

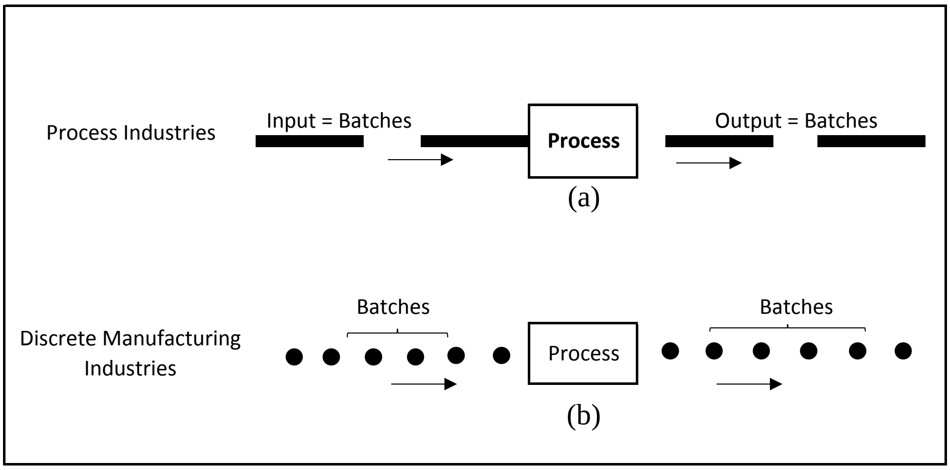


Figure 1.2: Batch production for process industries (a) and discrete manufacturing industries (b) (Groover, 2008)

In process industries two types of batch production can be distinguished: ***multiproduct*** and ***multipurpose*** batch process industries. If the routing is similar for all products, it is called multiproduct, whereas if various products follow different routings (e.g., job shops) it is called multipurpose batch process industry. Since in process industries intermediate products are unstable, they need to send to the next station to be processed with no delay. Consequently, the capacity of multipurpose batch process industries is lower than the other model (Raaymakers and Weijters, 2003).

1.1.3. Continuous Production (CP)

In this type, production lines often run for 24 hours on a three shifts basis, and maintenance shutdowns happen at irregular intervals due to economic reasons. Unlike the job production, this type is highly automated such that items transfer from one stage to another automatically and continuously (Roy, 2007). Also in contrast with two previous types, this type shows very low or even no flexibility. So, the layout and equipment are designed to be used for producing only one type of product or with minor changes (Cooke and Rohleder, 2006).

Mass Production (MP) and ***Flow Production (FP)*** are the two types of continuous production systems. In both of them a large number of identical products is produced automatically; however, in the former, the flexibility of layout and tooling for producing another item in same production processes is more than the latter. In other words in flow production layout and equipment are designed for only minor modifications, so, if the decision is made to switch over to a different type of product, extensive changes in layout and equipment are needed. There also exists another type of production which combines both continuous and batch production systems called ***Semi-Continuous Production (SCP)***.

Since a large number of products are produced in this production type, a higher amount of discount is probable, at the time of purchasing raw materials in contrast with job and batch production industries. Lower labor cost is another advantage of this type, as the production system is highly automated, and only a few skilled workers are needed. On the other hand, the risk of failure could be high, due to the change in demand and low flexibility of production line (Roy, 2007).

This type of production may be exploited in both discrete product and process industries. In discrete product industries, continuous production means that all the production equipment are dedicated to manufacturing a product without any breaks for changeovers. By contrast, in continuous production process industries, the process is done on the materials while they are passing through the equipment continuously and the output flow has no interruptions. Materials are usually in the forms of liquid, gas, powder or similar state (Groover, 2008). In continuous flow process industries, the focus of scheduling approaches is mostly on bottleneck stages with only a single processing unit. This is due to the lack of flexibility in this type of production that often reduces the chance of using multiple machines and consequently causes bottleneck production (Fransoo, 1993). As a result, single-machine, multi-product, and lot-sizing problems are mostly followed for scheduling of continuous process industries (Kılıç, 2011). Figure 1.3 shows the differences between continuous production in discrete product and process industries.

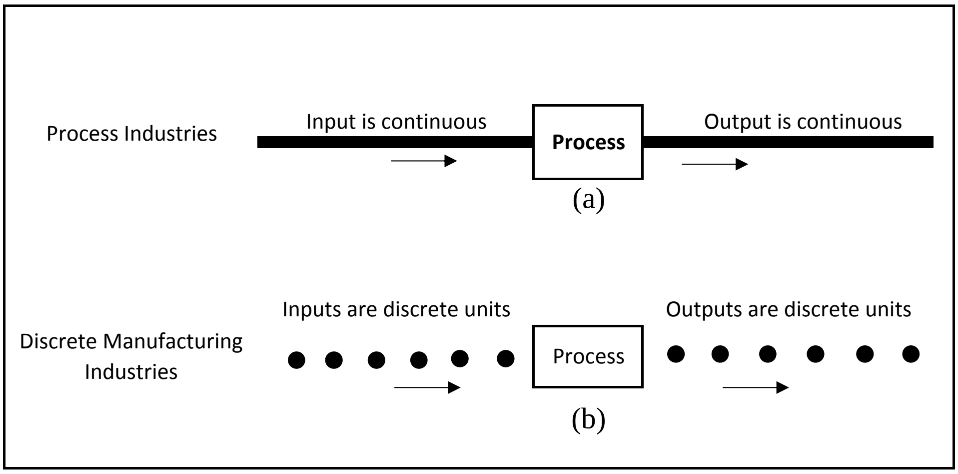


Figure 1.3: Continuous production for process industries (c) and discrete manufacturing industries (d) Groover (2008)

1.2. Classification of Scheduling Problems

The representation of time and processing events play a significant role in modeling scheduling methodologies. Understanding the differences in various modeling approaches, the problems they can be applied to, and their advantages and disadvantages are crucial to finding the best scheduling model (Hazaras, 2012). Scheduling models can be classified according to three different time representations: discrete-time, continuous-time, and mixed-time. Each of these three representations has its application considering the nature of industries. Sections 1.2.1, 1.2.2, and 1.2.3 provide more detail regarding these three models.

1.2.1. Discrete-time Models

Modeling the process scheduling was firstly based on the discrete-time representation in which the scheduling horizon is divided into some intervals with equal durations such that the starting and ending of tasks are within predetermined boundaries. Figure 1.4 illustrates the discrete time approach.

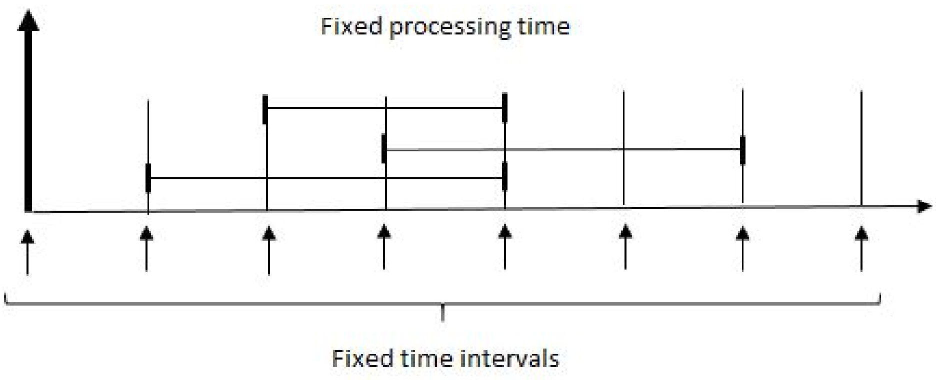


Figure 1.4: Discrete-time representation (Maravelias, 2005)

Making a reference network of time for all tasks is the advantage of this approach which is useful in terms of monitoring tasks at predefined intervals.

This feature reduces the problem complexity and makes it easier to be formulated. However, since for an appropriate estimation of the initial problem the time intervals need to be as small as the time of the shortest task, many variables are needed for the formulation and the size of the scheduling problem will increase exponentially. So the problem transforms into a combinatorial problem. Furthermore, for the problems that the processing time depends on the batch size, operations are difficult to be accounted for discrete-time models (Hazaras, 2012).

1.2.2. Continuous-time Models

Since the nature of time is continuous, formulating the actual problems in discrete-time models is only an approximation with less accuracy which might lead to a suboptimal solution. Moreover, for operating systems with variable processing times such as continuous flow systems, which runs continuously and are capable of doing tasks for time periods with any durations, this approximate modeling may cause a considerable deviation from the true solutions. Limitations of discrete-time approach have researchers to develop the continuous-time model over the last two decades. In this model, processes can be started and finished at any time within the time horizon. In fact, in this model, the duration of operations are decision variables and to be found. Since in these models there is no data dependency like in discrete-time models, and the timing of events can be changed during the horizon, the size of the scheduling problem in terms of mathematical programming is much smaller, and needless calculation. However, this variable timing of events makes scheduling processes more complicated to be modeled compared to discrete-time models (Floudas and Lin, 2004). The basic illustration of continuous-time representation can be seen in the figure 1.5.

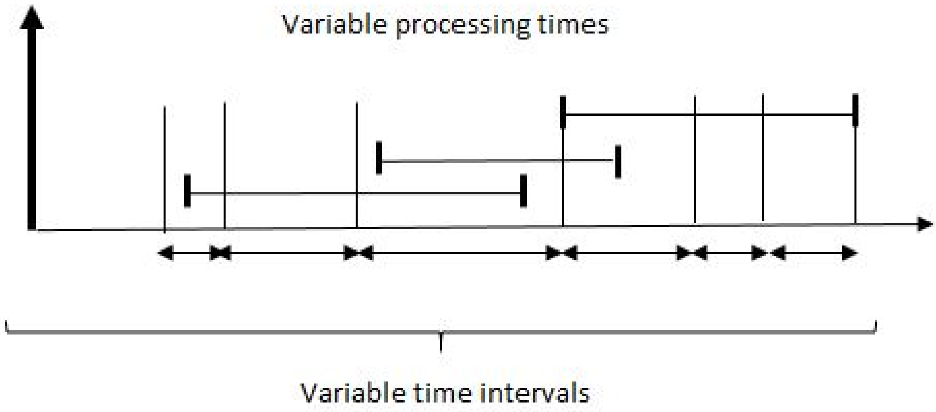


Figure 1.5: Continuous-time representation (Maravelias, 2005)

1.2.3. Mixed-time Models

Since there are some limitations for both discrete time and continuous time models, a mixed-time approach was presented by Maravelias (2005) to address these limitations. In this approach the time is divided into intervals with equal duration while the starting time of tasks are fixed just like the discretetime representation, but there is no exact finishing time for production operations, so the number of time intervals that operations can cover is not pre-specified and they can take as many intervals as desired. However, they should be finished at or before the ending of the last time slot (Maravelias, 2005).

One of the advantages of this time representation over the continuous time models is to linearly handle back-order costs and carrying inventory costs which is impossible in continuous models. Furthermore, this approach can model the due dates and material delivery without any increase in computational costs. With this model, semi-continuous production systems can be scheduled more precisely than the pure discrete-time approaches (Hazaras, 2012). The basic illustration of this model is shown in figure 1.6. The focus in this thesis is restricted to scheduling problems of continuous production process industries in continuous-time scale. Some of the basic characteristics of process industries and their importance are

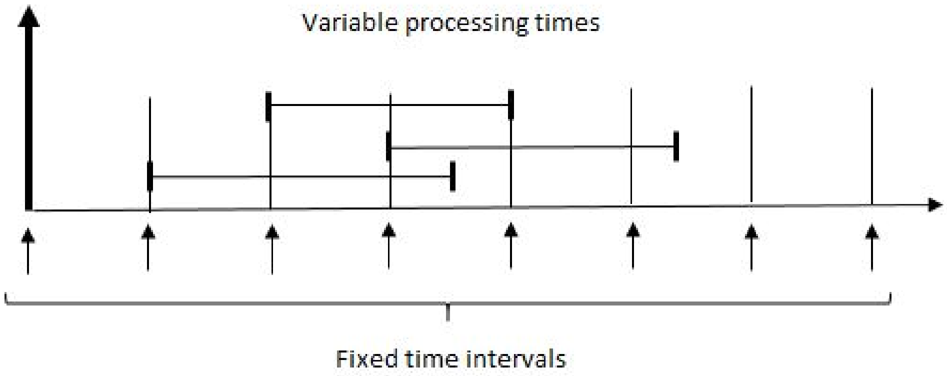


Figure 1.6: Mixed-time representation (Maravelias, 2005)

discussed in the following section.

1.3. Characteristics of process industries

A vast amount of research has been conducted regarding the characteristics of process industries. Many of them are particularly focused on different situations of production processes, demand management, and quality in both discrete product and process industries (Kılıç, 2011). However, not all of those characteristics are directly helpful in term of realizing the scheduling problem in the process industry. The following characteristics are the ones with the highest importance in terms of defining scheduling problems in process industries.

Raw materials

Raw materials play an important role in scheduling process industries. This is mainly due to the fact that most of them are directly sent from mining or agriculture industries and the quality of them may vary at different times. This puts an uncertainty in the process of scheduling and makes it more difficult, and the quality and availability of raw materials must be taken into account when scheduling the manufacturing system (Gunasekaran, 1998).

Recipes

Recipes show the ingredients of a product and the steps which they should take to be mixed in order to produce the intermediate and final products. Since several types of intermediate and final products are produced by only a few raw materials, different recipes could potentially be used in case of facing different limitations such as seasonal considerations and the scarcity of raw materials. This can directly affect the decision made by the schedulers (Kılıç, 2011).

Perishability

In process industries, materials are perishable and could decay in each step of the production. Hence, special attention must be paid for the handling of inventories during all the production processes and shipping to the customer. This feature puts another constraint on the process industries scheduling problem in terms of considering the storage units and the corresponding time. In addition, perishability of material is one of the factors that must be considered when selecting the batches size in order to avoid any wastage and back-order costs in process industries (Akkerman and van Donk, 2009).

Traceability

In process industries, it is often needed to have a system in which the origin of materials and products can be tracked. Especially in food processing industries where it leads to food safety. The advantage of traceability is mostly for when products are recalled in case of any quality problem (Rong and Grunow, 2010). Keeping track of materials can be a very challenging job since batches of materials are either merged or separated in consecutive processing and storage operations.

Storage

Obviously, storage limitations are common between both types of production systems. However, due to the nature of materials in process industries, storage limitations are more challenging. In spite of discrete manufacturing systems that use buffer or warehouses for storing intermediate products or raw materials, in process industries storage operations are done by discrete storage units (e.g. tanks, vessels, silos) that can store only one type of material at a time. This causes a capacity constraints for every single product rather than the total capacity of storage. Furthermore, in case of the flexibility of the storage units that can be used for different materials, they need to be allocated properly to maintain traceability and avoid perishability of materials. All these constraints, put complex restrictions into scheduling problem (Kılıç, 2011).

Setups

Production setups is another issue which is required to be considered in scheduling problems. In process industries, production setups are more time-consuming and need more effort than discrete part productions. This is due to the fact that in addition to the time for configuration of machines in case of any changeovers, there is a time needed for cleaning them as well. Consequently, in process industries, it is common to have sequence dependent setups to minimize the cleaning time (Van Wezel *et al.*, 2006).

Each of the characteristics mentioned above can have a significant effect on scheduling problems. However, all these limitations should be dealt with simultaneously in process industries scheduling which highly increases the complexity of the problem.

1.4. Organization of the thesis

The remainder of this study proceeds as follows. In chapter 2, the literature of scheduling problems on continuous and semi-continuous process industries is reviewed. Both direct and metaheuristic methods are considered. In chapter 3 the problem and its goal are described in detail, and a numerical example is given to clear the objectives of the problem and terms to be optimized. In chapter 4, solution procedure for selected metaheuristics is depicted in detail. Chapter 5 compares the performance of considered methods to find the best among them. Also, the design of experiments on the proposed algorithms has been done to analyze the performance of each method individually. And finally, conclusions and future research are presented in chapter 6.

Chapter 2

Literature Review

2.1. Introduction

Scheduling of both continuous and batch process industries is mainly concerned with the efficiency of processing operations in manufacturing systems. There exist several factors and constraints that make process industries scheduling complicated such as equipment networks, complex product recipes, storage limitations, material delivery, and utility restrictions. Due to these constraints, a vast amount of research has been conducted to explore and analyze various methods for planning and scheduling manufacturing systems in process industries over the last three decades (Hazaras, 2012). These methods are classified into two main categories: mathematical programming such as Mixed Integer Programming(MIP) and Branch and Bound (BB) method and metaheuristics such as Genetic Algorithms (GA) and Simulated Annealing (SA). Figure 2.1 shows the methods reviewed in this study.

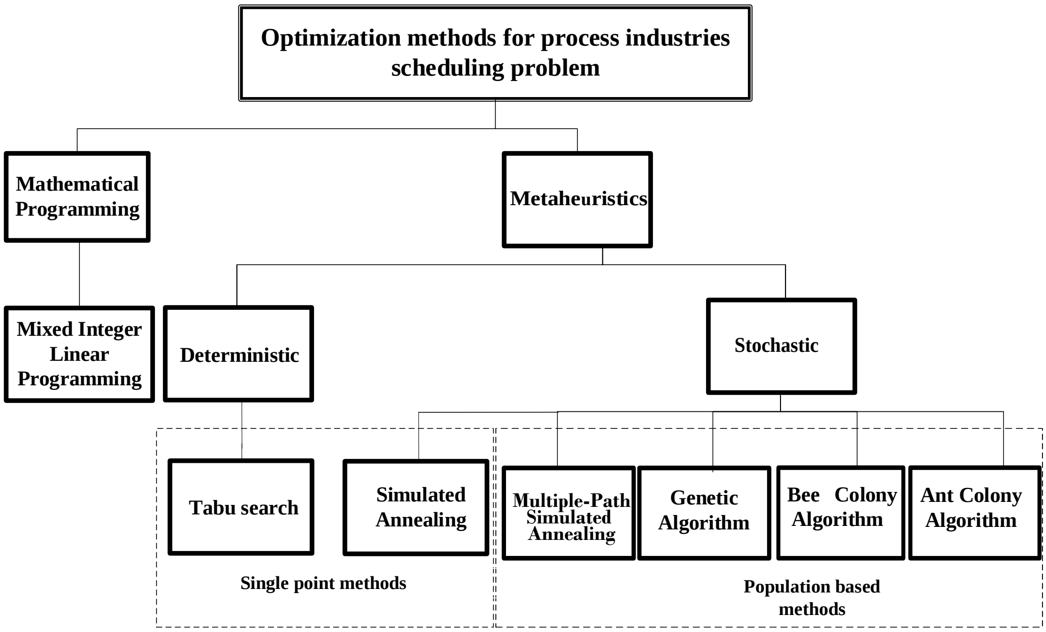


Figure 2.1: Taxonomy on the reviewed scheduling methods in process industries

2.2. Mathematical Programming

Mathematical programming is an approach on which strong and effective optimization methods for production planning or scheduling can be based. This approach has received much attention and interest from researchers in recent years. This is mostly due to the recent progress in information technology which makes organizing production data easier to be based on the time (Shapiro, 1993). Many search methods have been explored in the literature; however, mathematical programming especially Mixed Integer Linear Programming (MILP) is the most widely used method for scheduling as the solutions achieved by them are highly acceptable (Floudas *et al.*, 2005). Mendez (2006) proposed a method based on MILP for the optimization of the off-line blending and scheduling problem of an oil refinery simultaneously. The method can be used for problems with both discrete and continuous time representations. They presented an iterative procedure to address the nonlinearity of the recipes variability for different gasoline grades. Also, a set of variables and equations were considered to prevent infeasible solutions by defining penalties in the objective function. Jia *et al.* (2003) proposed a model in continuous-time scale which results in a MILP problem to address the scheduling problem in a crude oil refinery. They only considered the first section of the refinery operations which includes crude oil unloading, mixing, and inventory control. The proposed mathematical formulation was applied to four case studies to show that continuous time representation can achieve solutions more efficiently than discrete time models. The model for scheduling the last part of the refinery was developed by Jia and Ierapetritou (2003) which includes gasoline blending and distribution system. They also exploited continuous time formulation to avoid big size problems in terms of variables and constraints. Perfect mixing, negligibility of the time between changeovers, and most importantly the fixed recipe for each product are the assumptions made in this model. A mixed integer optimization model was proposed by Lee *et al.* (1996). A branch and bound method which is based on Linear Programming (LP) was also used to solve the model. Also for reducing the computation time, they used priority branching and bounding. The objective was to minimize the transition time and inventory costs. Shah and Ierapetritou (2011) proposed a mathematical model based on continuous-time representation for short-term scheduling in large-scale refineries. The model integrates quality, quantity, and logistic decisions in order to schedule real-life refinery operations. A set of valid inequalities corresponding to tanks scheduling and loading/unloading events was included into the model constraints intending to reduce the computation time. However, their model cannot find the optimal solution of a real-life refinery in a reasonable time. A mixed-time representation based MILP was developed by Kopanos *et al.* (2009). The aim of the proposed model was to address a lot sizing and production scheduling problem in multi-product semi-continuous food industry. Same holding cost for same product families, fixed demand during the scheduling time horizon, and sequence dependency of time and costs are the main assumptions of their model.

A mixed integer programming model and a solution approach for scheduling a multi-stage multi-product plant with the semi-continuous production were proposed by Kopanos *et al.* (2011) in order to minimize the production time. They tested their model in several real-life industrial case studies to illustrate the capability of it to be utilized in fresh food industries. Their model was enhanced by Kopanos *et al.* (2012) via defining new sets of constraints to tighten the search area in achieving solutions with more efficient computational efforts. In their work, a MILP formulation was also proposed based on a continuous time scale to address the scheduling problem of a real-world multistage food industry. The model is to integrate and simultaneously optimize all production stages to make an interaction between different stages easier. This model does not guarantee the global optimal solutions, however, a reasonable time needed for finding them.

2.3. Metaheuristics

During the recent decades, metaheuristics have been employed in various scheduling problems. This is due to the robustness of their solutions over heuristics and the time limitations of exact algorithms (Xhafa and Abraham, 2008). All metaheuristics are to enhance the current best solution among a set of solutions by performing guided stochastic search which is either done broadly(exploration) or locally (exploitation, refinement). An attractive feature of this approach in comparison to mathematical programming is that they can be easily matched with the simulation models that indicate many details of the problem including constraints that are difficult to be modeled by equations. The main ability of metaheuristics is to escape from the local optima in non-convex nonlinear problems or problems with disconnected feasible regions. In fact, metaheuristics direct the search algorithm into favorable regions by using penalty functions with large enough value to dominate the worst value of cost function. Generally, metaheuristics do not guarantee to find the global optimum for the problem at hand but often lead to good solutions that meet the constraints within a reasonable amount of time (Harjunkoski *et al.*, 2014). Metaheuristic methods fall into two categories: population-based and single point methods. Single point approaches such as Simulated annealing and Tabu Search (TS) modify and improve a single candidate solution in each iteration until convergence happens. Population-based methods try to improve multiple solutions often by guiding the search by population characteristic. Evolutionary Algorithms (EA) such as genetic algorithm (GA) and Genetic Programming (GP), Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), and Particle Swarm Optimization (PCO) are some of the most famous population-based methods.

2.3.1. Genetic Algorithm (GA)

The genetic algorithm is one of the population-based metaheuristics which mimics the biological evolution. A population of feasible solutions is firstly initialized, then in each iteration, it tries to improve the solutions by applying reproduction and mutation operators until the stopping criteria are met (Erdogdu, 2008). Oliveira *et al.* (2011) developed two approaches including a GA and an MILP to address the scheduling problem of a refinery in order to minimize the unmet demand, operational changes, total stock at the end of the scheduling horizon (only for MILP), and number of products not allocated to the tanks (only for the GA). In their MILP method downtime and inspection of products in tanks were not considered. Both methods provide good solutions in terms of demand, but GA could find a fewer number of operational changes since it is capable of updating the weights corresponding to the objectives during the evolutionary process. Shaw *et al.* (2000) proposed two genetic algorithms in order to find the solution for a scheduling problem in a semi-continuous production system. The purpose of their model is to find out whether or not a batch started and finished, whether or not a batch is idle, and the flow rate of the continuous stage with the objective of maximizing the profit. Santos and Dourado (1999) presented a genetic algorithm to be applied for scheduling a continuous production system in order to minimize the energy costs and maximize the production rate. In their GA, Pareto Ranking Method (PR) is used for generating the new population. They consider both shutdowns forced by maintenance necessities and consequently the effect on production rate. In order to make genetic algorithm applicable for the refinery scheduling problem Hou *et al.* (2017) firstly used schedulability conditions proposed in Wu *et al.* (2011), Wu *et al.* (2008), and Wu *et al.* (2009) to transform the problem into a resource assignment problem and then applied a Nondominated Sorting Genetic Algorithm (NSGA-II) to it. However, they did not consider the effects of pipeline flow rate in the costs of crude oil refining.

Amorim *et al.* (2011) simultaneously addressed both the lot-sizing and scheduling problems of a semi-continuous production system by considering the perishability of products. In the first step, they developed two multi-objective MILP models and then implemented Make-To-Order (MTO) and a combination of it and Make-To-Stock (MTS) environments to those and finally applied an NSGA-II to the models. The advantage of their multi-objective framework is to offer a trade-off between the freshness of the products or the total cost. Dahal *et al.* (2001) proposed a Hybrid Genetic Algorithm (HGA) including a GA and a heuristic for scheduling storage tanks in the ballast water treatment system. They decomposed the problem into integer and real-number subproblems, so the GA was used for the integer subproblems while the heuristic approach was exploited for the real-number subproblems within a GA framework. The proposed approach was tested in three case studies to evaluate its robustness over random search method. Ramteke and Srinivasan (2012) proposed a new GA for scheduling the continuous flow production in an oil refinery. To increase the computational efficiency of the model, they based the chromosomes representation on the concept of scheduling graphs, which changes the GA to a Structure Adapted Genetic Algorithm (SAGA). Toledo *et al.* (2014) presented a method in which genetic algorithm is combined with the mathematical programming to simultaneously solve the lot-sizing and scheduling problems in a soft drink industry. The GA was used for addressing the sequencing of the production lots which simplifies the linear programming model for the lot sizing problem. The setup times are supposed to occur for each preparation tank even if the raw materials are same. A multi-population GA with the ability of migrating solutions between populations was proposed by Toledo *et al.* (2009) to address the problem of lot-sizing and scheduling a semi-continuous production system. Unlike the work of Toledo *et al.* (2014), the sequence-dependency of setup times was considered in this method.

2.3.2. Simulated Annealing (SA)

Simulated Annealing is a single point metaheuristic which is inspired by the process of annealing in metallurgy. In the annealing process, in order to reach the most stability in the materials, they are heated first and then started to be cooling slowly. Simulated annealing uses this method by defining a temperature factor (T) which affects the probability of accepting a new point as the current best solution during the search steps. In each iteration, an adjacent point is randomly selected and compared to the current solution. If the newer one has a better fitness value, it would be selected as the new solution, but if not, it still has a chance to be selected according to the acceptance probability function. The temperature factor is reduced in each iteration until it becomes zero which means the current solution cannot be improved more. This feature makes SA more strong in terms of finding the global optima, since worse solutions are able to be accepted, in order to search their neighborhood.

Huegler and Vasko (2007) proposed a strategy for scheduling a steel industry with continuous casting process which combines a domain-specific heuristic with any one of three metaheuristics such as simulated annealing, Generational Evolutionary Programming (GEP), or Steady State Evolutionary Programming (SSEP). In their proposed method, a near-optimal solution is firstly generated by the heuristic and then used as the initial solution for the metaheuristic. The comparison between the performance of three different metaheuristics shows better performance of the steady-state method. Toledo *et al.* (2013) proposed a hybrid multi-population evolutionary algorithm which combines a genetic algorithm with simulated annealing and a heuristic called cavity heuristic in order to solve the lot-sizing and scheduling problem of a two-stage semi-continuous production process. In their proposed method, the role of SA is to reinforce the search for better solutions in the neighborhood of the best solution generated by the GA in the converged population. A simulated annealing was applied as a part of optimization method proposed by Chen *et al.* (2017). In this method which was employed for scheduling the delivery/injecting plans of different stations in a pipeline, the objective is to minimize variations in the pump rate. In their method SA works between two heuristics, the first one finds an initial solution and after solution improvement by SA, the second heuristic provides a solution refinement. They assumed a high cost for flow restarts in the pipeline segment, so none of the intermediate delivery stations can be full-stream. Meyr (2000) introduced a new MIP model to simultaneously address lot-sizing and scheduling problem of a multiproduct capacitated production line. In their model continuous lot sizes and deterministic dynamic demands without backlogging were considered with the objective of minimizing inventory holding cost and the sequence-dependent setup costs. A simulated annealing approach was applied to the model to find near-optimal solutions in terms of the sequence of setups.

2.3.3. Tabu Search (TS)

Tabu search is also a single point metaheuristic which uses a local search technique for solving optimization problems. In fact, it improves the performance of local search by relaxing its basic limitation in order to avoid getting stuck in local optima. Once no other improved solution is available, TS can accept a worse solution to search its neighbourhood in hopes of improvement. Meanwhile, in order to prevent tracing back to the point, it came from in next iterations, previously visited solutions are memorized in a memory called tabu list, so they cannot be visited again. Toledo *et al.* (2011) proposed a Tabu search approach to solve the scheduling and lot sizing problem of a soft drink plant simultaneously. Sequence-dependent setup time and setup cost, as well as inventory cost for an excessive number of products, are considered in their model. Hence, the objective of their model is to minimize all costs including production, setup, and inventory costs for products and raw materials in lines and tanks. Zandieh *et al.* (2016) developed a tabu search approach which is called gradual transition tabu search for scheduling both batch and continuous manufacturing systems. The objectives of their model are to minimize production time and delay in meeting the demand. Hindi (1995) applied a tabu search for a capacitated single-item lot-sizing problem. Startup costs for switching the production facility on or off and reservation cost for keeping the facility on, regardless of involving with the work, were considered in their model. However, they did not use sophisticated measures such as multiple starts, diversification and intensification strategies to improve the performance of the Tabu search. Göthe-Lundgren *et al.* (2002) firstly formulated a MILP model for the problem of scheduling in a refinery where the capacity of inventory is limited. Secondly, two approaches were applied for solving the MILP: using a MILP solver and using tabu search. The objective was to minimize the costs of changing the mode of operation and completely satisfying the demand.

2.3.4. Ant Colony(AC) and Bee Colony(BC) Optimization

ACO and BCO are both population-based methods inspired by the behaviour of ants and bees for finding foods respectively. In the real world, ants initially search for food randomly and once they find a source return to the colony while laying down pheromone trails. So other ants can stop random search once they find a path, and lay down their own pheromone each time coming back. Thereby, continuing to do so, have other ants to focus on more strong paths. Real-world bees, however, wait for a small portion of their colony to search randomly and bring back food. The profitability of food is evaluated by scouts and other bees focus on more profitable food resources. In both methods, the population is the colony, solutions are food sources, the fitness value is the profitability of food and pheromone strength, and species are the search operators. Pan *et al.* (2013) proposed a bee colony algorithm for steel-making manufacturing system with the continuous casting process. They first presented a mixed integer programming model and then applied BC for solving the scheduling problem. Also, two heuristics were used to increase the performance of the BC. They did not consider breakdowns in the machines or error when processing products. Pan (2016) proposed a co-evolutionary bee colony algorithm for scheduling a steel-making process with continuous casting. They employed the decomposition method to divide the problem into two sub-problems including a hybrid flowshop and continuous casting scheduling problems. Furthermore, the proposed BC method has two sub-swarms, each addressing one sub-problem. Two heuristics were applied to enhance the performance of the proposed co-evolutionary BC. They also considered deterministic and uninterrupted setup times of all casts and transportation time between stages. Gravel *et al.* (2002) presented an ant colony method for solving the scheduling problem of the continuous aluminium casting centre. Four objectives were considered in their model: minimizing the unused capacity of facilities due to setup times, satisfying demand on-time, minimizing the total number of draining when changing alloys, and minimizing the total unused capacity of transportation vehicles. Atighehchian *et al.* (2009) proposed a method which combines ant algorithm with non-linear optimization in order to solve the scheduling problem of a steel-making process with continuous casting. They considered assurance of continuity of the production process and minimizing costs as the objectives of their method. Ferretti *et al.* (2006) applied an ant algorithm to a scheduling problem in a steel-making process in order to find the most profitable schedule. Their main emphasis is to consider finished product warehouse as a cooling area which is a part of the production process.

2.3.5. Research Motivation

As already discussed, continuous production systems are used in industries with high levels of production rates and high volume of products. Consequently, they are accounted for a larger percentage of the Gross Domestic Product (GDP) of the country. Scheduling continuous productions and process industries is often not as simple as in discrete production systems such as job shops and flow shops. This could be due to the lack of flexibility in the system layouts and machines for different products in this type of production. Moreover, scheduling approaches may change in various process industries with different products which are also considered as a problematic issue.

Conducting research and developing models case by case can cover different scheduling problems in continuous production systems which is a good way of understanding certain features and aspects of problems by the research community. So that necessary knowledge can be accumulated based on the area of research for the specific problem under consideration which may not be applicable for a wider range of process industries, and eventually, different algorithms could be developing based on those approaches. A local process industry has been considered in this thesis as a scheduling problem since it is not commonly found in the literature and is very different from the standard problems which make it a unique problem which deserves more research.

Chapter 3

Problem Definition

3.1. Problem Description and notations

In this section, the scheduling problem under consideration and its corresponding notations are described. The scheduling problem is based on a case study which belongs to food processing manufacturing system with a continuous flow production. Products are categorized based on their flavors and packaging sizes. So, those with the same flavors and different package sizes are considered as different products as well as products with different flavors. With this classification, there manufactures 57 different products in the system (P_i, i=1,...,57). The materials move continuously on a conveyor which passes through a fryer in the first step. The fryer should process 2770 kilograms of them per hour to keep the quality at a high level. Changing the speed of the conveyor and consequently, the amount passes through the fryer can make materials burned or oil-soaked. In the next step, processed materials go through the seasoning and packaging machines and are used in different product types. There are 10 stations each includes a seasoning and a packaging machine (M_i, i= 1, ...,10). There are 2 important factors that affect the machine scheduling procedure. Machine capacity and machine efficiency. Different machines can process different amount of materials based on the type of seasoning or packaging. It means that the capacity of machines can differ by changing the products on which they are going to work. In addition to the capacity, there exist different rates of efficiencies for processing different products by different machines. Efficiency rates are the probability with which the machines can keep processing without failing on average. This means if a machine is not capable of processing a special type of product, its efficiency for that product is zero and would be called an ineligible machine.

Here, the scheduling problem can be defined as the problem of finding the amount of time that each seasoning and packaging machine should take for each product in such a way that the demand can be met on-time while minimizing three objectives discussed in section 3.1.1.

3.1.1. Objective functions

The scheduling problem in this thesis is defined as a multi-objective problem. There are three objectives to be optimized simultaneously, but different weights may be assigned to objectives based on the priority of decision makers. All calculations are done based on the demand for products which may vary from one schedule horizon to another. This fact makes implementation of the scheduling procedure necessary before starting every schedule horizon.

Minimizing the average percentage over-pull (A)

As discussed in section 3.1, the performance of machines can differ based on the product on which they work. There are two different indices defined for the capacity of machines: theoretical and actual capacity. In theory, a machine may be able to process a certain number of products, but when it comes to real production, there exist some problems that may lead machines to fail to process well. It cannot be certainly specified that when a machine fails to process a product properly; however, it can be estimated that how many breakdowns a machine has during working on a product on average. This estimation is called efficiency of that machine for a specific product.

When the system is running, machines are assigned to products based on the schedule. So, in order for reaching the production level specified by the schedule, all machines need to be supported by the materials on time. However, in some cases, when some machines work with their theoretical capacities, the volume of fried materials needed by machines at a certain period of time would become higher than the maximum capacity of the fryer. The amount of material that machines require more than the fryer capacity is called over-pull. The first objective of this study is to minimize the average percentage over-pull, in order to minimize the risk of having idle machines at the end of the production line.

Minimizing the difference between the target KG and fryer capacity in each time slot (B)

The time representation used in this scheduling problem is a discrete time model, as discussed in section 1.2.1, so the scheduling time is divided into equal time-slots. This approach can simplify the scheduling formulation. On the other hand, as it was already discussed, the speed of the fryer and consequently its capacity are limited. By determining the sequence of products for each machine, all machines can be working simultaneously. Considering 10 minutes for each time-slot and the total capacity of the fryer, the amount of raw material that the fryer can process at each time-slot will be $\frac{2770}{6} = 461.66$ KG. The term *target KG* can be defined as the amount of material that actually needs to be processed by the system within a time-slot. So the scheduling is basically done based on the target KG which considers the efficiencies of machines, and the summation of all allocated materials to all machines cannot be more than the actual capacity of the fryer in a certain time-slot. On the other hand, if the amount of processed materials required by machines at a specific time-slot is less than the provided amount by the fryer, *under-pull* happens. Having under-pull for a specific amount of time could cause waste of materials, since they would not be allocated to any machines and eventually have to be used as animal food.

Minimizing the number of products with production rate outside the target range (C)

In this case study the level of demand is not a fixed number and may change in the different period of times, but based on the history of the customers demand in the past, a target range can be defined to show the approximate demand of different products. This range has a minimum and a maximum target between which the quantity produced needs to fall. Being in the target range can help the production system to have lower wastage and backorder costs which can somehow affect the first two objectives. So, it can be said that, in the scheduling problem of this work, it is desirable to determine which machine should be allocated to which product and for how long to keep the production level between the targets.

Once the objectives A, B, and C are defined individually, the total objective function can be formulated based on the importance of the objectives and their corresponding weights. The general form of the objective function can be defined as shown in Eq 3.1 where W_1, W_2, and W_3 are the weights that can be selected at the discretion of the decision maker and the importance of the objectives.

$$\textit{Minimizing the total weighted sum} = (W_1 * A) + (W_2 * B) + (W_3 * C) \tag{3.1}$$

3.2. Numerical Example

In this section, a small size numerical example is provided which is a simulation of the real problem considered in this thesis. The aim here is to clarify different concepts of the problem and illustrate what has been provided in the previous section. So, this example is regardless of the optimization methods that could possibly be used to solve the problem, and algorithm steps which are explained in chapter 4 in more details. In this example, twenty products are produced by five machines over a one-day production. For the sake of simplicity of demonstration, each time-slot is thirty minutes and the hourly capacity of the fryer is 1300 KG. So in each time slot, only 650 KG of products can be processed. Also, for products with the same flavours, only one time-slot is considered as setup time, while it takes two time-slots to setup the machines when flavours are different. Product weights, flavours used in products and their demands, production rates, the efficiency of machines according to products, minimum and maximum target for each product, and trial scheduled quantity are shown in table 3.1.

Figure 3.1 also shows the schedule of the example by considering data such as production rates and efficiencies of the machines provided in table 3.1. The scheduling horizon is only a day including three shifts starting 7 AM for twenty-four hours. As mentioned in section 3.1.1, there is a theoretical capacity for each machine processing a product, which is higher than the actual capacity of the machine. So the concept of *Instant KG* is the summation of all scheduled theoretical capacities in different time slots. Then there is a *Target KG* which is the instant KG multiplied by the efficiencies of machines and the scheduling is done based on this quantity. So a product cannot be processed in time-slots with target KG more than the capacity. This has been demonstrated in figure 3.1 for products 20 and 11 on machines 4 and 5 respectively.
Clearly, instant KG is higher than the target KG in each time-slot, and could also pass the maximum capacity of the fryer. So the concept of over-pull is defined here which has been shown in table 3.2. Over-pull can vary in different time slots based on the difference between instant KG and the capacity. For instance, the over-pull at 7 AM is around 14

As it can be seen in table 3.3, eleven products are scheduled to be produced outside of the demand range which is fifty-five per cent of all products. This

Table 3.1: Processing data of the example problem

Product				**Capacity (bags per minute)**					**Efficiency of machines**				
No	Weight(gr)	Demand	Flavor	M 1	M 2	M 3	M 4	M 5	M 1	M 2	M 3	M 4	M5
1	70	790	4	0	60	55	55	0	0	0.89	0.88	0.82	0
2	62	880	5	0	60	55	55	0	0.82	0.88	0.88	0.82	0
3	194	855	1	0	0	40	40	0	0	0	0.88	0.82	0
4	165	325	1	0	0	40	40	0	0	0	0.88	0.82	0
5	210	445	1	0	0	40	40	0	0	0	0.88	0.82	0
6	56	411	1	0	70	60	0	0	0.88	0.88	0.88	0	0
7	69	560	3	0	90	64	0	0	0	0.88	0.88	0	0
8	56	395	3	0	70	60	0	0	0	0.88	0.88	0	0
9	56	325	4	0	70	60	0	0	0	0.88	0.88	0	0
10	220	700	7	36	0	40	40	0	0.75	0	0.88	0.82	0
11	122	3978	4	0	70	60	0	38	0	0.88	0.88	0	0.75
12	75	816	6	0	60	55	50	64	0	0.88	0.88	0.88	0.75
13	75	854	2	0	60	55	50	0	0	0.88	0.88	0.88	0
14	75	820	6	0	60	55	50	0	0	0.88	0.88	0.88	0
15	98	758	5	0	60	55	50	0	0	0.88	0.88	0.88	0
16	197	664	4	0	60	55	50	0	0	0.88	0.88	0.88	0
17	96	3425	7	0	60	55	50	0	0	0.88	0.88	0.88	0
18	220	3256	7	36	0	40	40	0	0.75	0	0.88	0.82	0
19	220	3603	3	36	0	40	40	38	0.75	0	0.88	0.82	0.75
20	300	2360	4	32	0	28	30	38	0.75	0	0.88	0.82	0.75

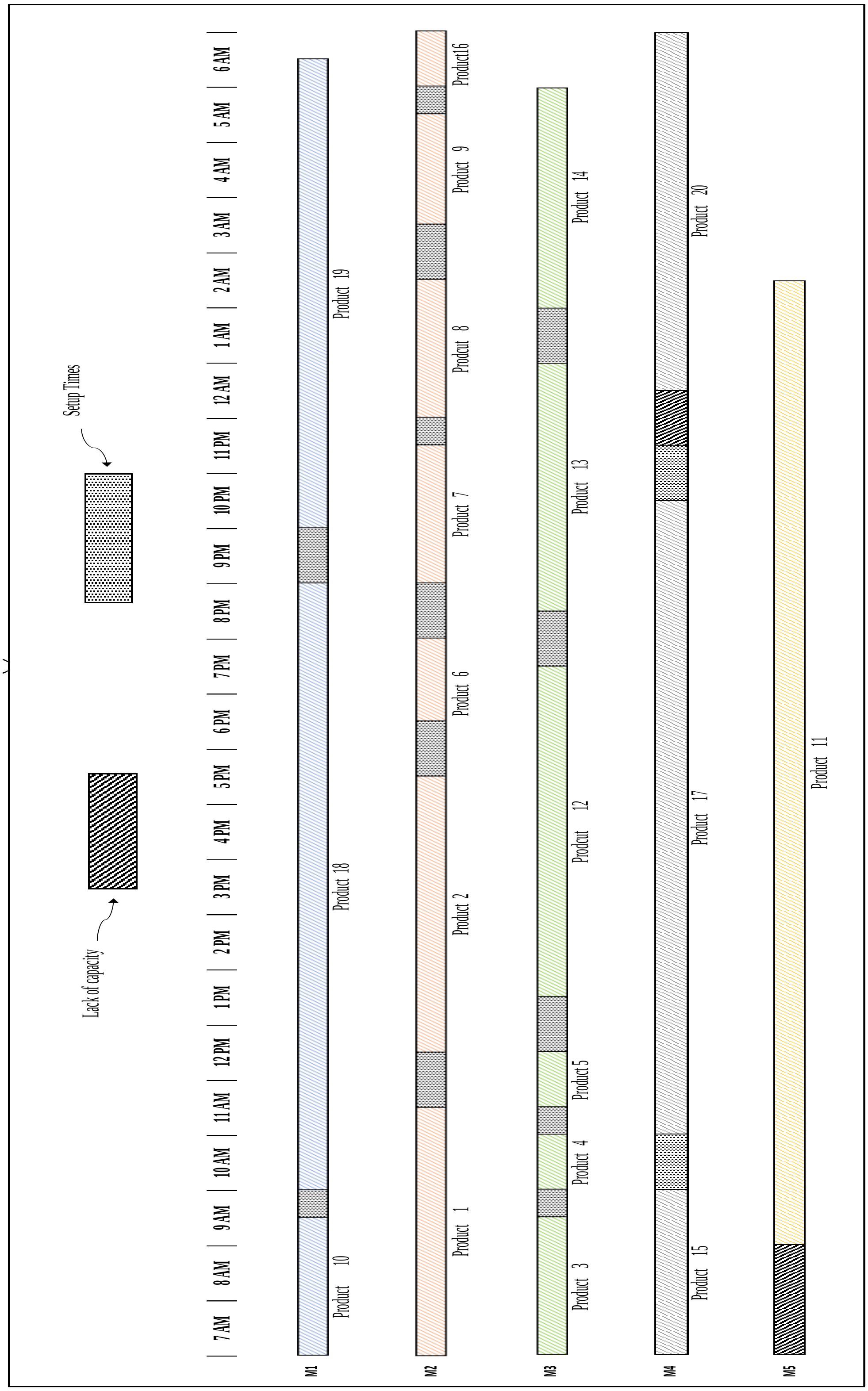


Figure 3.1: Production schedule for the numerical example

Table 3.2: Results achieved from figure 3.1

	7AM		8AM		9AM		10AM		11AM		12PM	
Capacity	650	650	650	650	650	650	650	650	650	650	650	650
Instant KG	744	744	744	744	883	505	700	700	660	786	786	645
Over-pull	14%	14%	14%	14%	36%	-22%	8%	8%	2%	21%	21%	-1%
Target KG	624	624	624	624	650	425	568	568	533	643	643	519
	1PM		2PM		3PM		4PM		5PM		6PM	
Capacity	650	650	650	650	650	650	650	650	650	650	650	650
Instant KG	645	769	769	769	769	769	769	769	769	658	658	776
Over-pull	-1%	18%	18%	18%	18%	18%	18%	18%	18%	1%	1%	19%
Target KG	519	628	628	628	628	628	628	628	628	530	530	633
	7PM		8PM		9 PM		10PM		11PM		12AM	
Capacity	650	650	650	650	650	650	650	650	650	650	650	650
Instant KG	776	652	534	658	607	607	844	686	686	500	618	888
Over-pull	19%	0%	-18%	1%	-7%	-7%	30%	6%	6%	-23%	-5%	37%
Target KG	633	524	421	530	516	565	650	555	555	391	600	650
	1AM		2AM		3AM		4AM		5AM		6AM	
Capacity	650	650	650	650	650	650	650	650	650	650	650	650
Instant KG	764	764	888	631	631	749	749	749	749	631	862	625
Over-pull	18%	18%	37%	-3%	-3%	15%	15%	15%	15%	-3%	33%	-4%
Target KG	606	650	650	508	508	611	611	611	611	569	650	533

shows the concept of the third objective of the problem explained in section 3.1.1 which is the minimization of the number of products with trial scheduled quantity outside of the predefined range.

Finally, considering $W_1 = 800$, $W_2 = 5$, and $W_3 = 10$ objective values of the example would be as follows:

- Average Over-pull = 10% $\rightarrow$ The first objective value = 800,
- Difference between capacity and target KG = 3211 KG $\rightarrow$ The second objective value = 16055,

Table 3.3: Difference between scheduled and target values (KG)

Products 1 - 10										
	1	2	3	4	5	6	7	8	9	10
Scheduled QTY.	1017	990	1020	348	442	309	820	515	412	890
Min Target	645	720	775	285	425	344	425	325	285	575
Max Target	930	1040	940	365	465	521	865	465	370	830
Products 11 - 20										
	11	12	13	14	15	16	17	18	19	20
Scheduled QTY.	3640	1308	981	872	780	624	3220	3916	3026	2873
Min Target	3612	747	762	621	544	546	2904	3147	3264	1455
Max Target	4160	1074	878	894	784	845	3607	3624	3987	2187

- Total KG outside target ranges = 1907 KG $\rightarrow$ The third objective value = 19070.

Therefore, Total objective value of the schedule is $= 800 + 16055 + 1907 = 18762$ which should be minimized.

Chapter 4

Selected Metaheuristic Algorithms

4.1. Introduction

In this chapter two metaheuristics are selected from several methods reviewed in section 2.3 and applied to the scheduling problem of a case study which was described in chapter 3: Genetic Algorithm (GA) and Multiple Path Simulated Annealing (MPSA). The organization of this chapter is as follows: in sections 4.1.1 and 4.1.2 algorithms and their terminologies are introduced, then steps of both algorithms are discussed in section 4.2.

4.1.1. Genetic Algorithm

Among all metaheuristics available for scheduling continuous process industries, GA is the most adopted one. This could be due to the high quality of the solutions and low computational time required. The algorithm starts with generating the initial population which includes a number of solutions called chromosomes, depending on the predefined population size. Chromosomes are formed by several genes which include information about the machines (Solution representation). All chromosomes are evaluated based on the fitness function, and then fitness values are assigned to the corresponding chromosomes (Evaluation). Most fitted solutions are then selected to form the next population (Selection) which is possible through different methods. Finally, cross-over operator (reproduction) and mutation operator (perturbation) are used with a predefined probability of applying to manipulate chromosomes with the hope of generating fitter solutions. Tuning the GA parameters (probabilities and the population size) results in a better final solution. Algorithm 1 shows the template of the GA method employed in this thesis.

Algorithm 1 Pseudocode of applied Genetic Algorithm

Begin
Generate the initial population (P_0)
while termination criteria are not met **do**

  for ($c = 1$ to S) **do** /*S=population size*/
    Compute fitness value of chromosome c
  end for

  Use a selection operator to form next population ($\mathbf{P}_{j+1}$)

  Perform cross-over on **r%** of randomly selected chromosomes /***r**=cross-over probability*/

  Perform mutation on **m%** of randomly selected chromosomes /***m**=mutation probabiliy*/
end while
  Select the most fitted solution in the last population as the final solution

End

4.1.2. Multiple Path Simulated Annealing (MPSA)

The basic simulated annealing was previously introduced in section 2.3.2. However, in order to improve the performance of simple SA in terms of quality of the solutions, multiple path SA is adapted in this work. MPSA is a population-based simulated annealing with the ability to simultaneously search different parts of the solution space. This means that solutions with lower probabilities of error can be found even in a shorter period of time in comparison to single path SA (Defersha, 2015). The pseudocode of proposed SA can be seen in Algorithm 2.

4.2. Components of Proposed Algorithms

Optimization steps of both proposed algorithms are discussed in this section. It is noteworthy that some algorithm steps are done similarly in both algorithms, while others not. We categorize optimization steps into two main subcategories: common steps and unique steps. Hence, the steps are not necessarily in the order of their own algorithms.

4.2.1. Common steps

Solution representation, fitness evaluation, and perturbation mechanism are the steps that both proposed methods employ in order to find the optimum solution. However, they are not necessarily done through the same steps (e.g. perturbation mechanism).

4.2.1.1 Solution representation

There has been presented numerous methods for solution representation in literature such as : (Shaw *et al.*, 2000) , (Hou *et al.*, 2017) ,(Dahal *et al.*, 2001), (Toledo *et al.*, 2014),(Toledo *et al.*, 2013), and (Ramteke and Srinivasan, 2012). However,

---

Algorithm 2 Pseudocode of Multiple Path Simulated Annealing

Begin
Set $k = 0$, $n = 0$, T_0 = initial temperature

Generate S solutions randomly $X_{0,1}$,$X_{0,2}$, ..., $X_{0,S}$

while $T_k > T_{min}$ **do**

  for ($q = 1$ to Q) **do**   /* Q= Maximum number of iterations in (T_k) */

    for ($j = 1$ to S) **do**   /* S= number of independent search paths */

      Generate $X'_{n,j}$ from $X_{n,j}$

      if $(E(X'_{n,j}) - E(X_{n,j}) < 0)$ **then**

        $X_{n,j} = X'_{n,j}$

      else if $exp\frac{-(E(X'_{n,j})-E(X_{n,j})}{T_k} > r$ **then**

        $X_{n+1,j} = X'_{n,j}$

      else

        $X_{n+1,j} = X_{n,j}$

      end if

    end for

      $n = n + 1$

  end for
      $k = k + 1$
      $T_k = \gamma \times T_{k-1}$
end while

  Consider X_{n,j^*} as the best solution

End

---

the method proposed in this thesis has not been applied before. The first step in both GA and MPSA implementation is to properly design a scheme from a particular problem to which optimization steps can be applied. This scheme in GA is a chromosome which can also be used as a solution in MPSA. Each chromosome contains 57 genes which is equal the total number of products. There are two elements by which a gene is formed. The first element is for indexes of products(P_i) and the second one is for indexes of product factors (F_i). Product factors (F_i) are random numbers and used for calculating the (Q_i) which is the quantity needs to be produced from the product exists in gene i during a schedule horizon. Figure 4.1 shows how a sequence of products with corresponding factors is encoded to a chromosome (solution).

Products	52	14	33	38	11	43	7	...	26	44	4	23	55	40
Factors	F_{52}	F_{14}	F_{33}	F_{38}	F_{11}	F_{43}	F_7		F_{26}	F_{44}	F_4	F_{23}	F_{55}	F_{40}

Figure 4.1: Solution representation for the scheduling problem of the case study

Thousands of such solutions are generated randomly in each iteration to form the initial population. Equation 4.1 shows the role of production factors in the calculation of their corresponding products quantity. Where TRG_{min} and TRG_{max} are the minimum target and the maximum target of that product respectively.

$$Q_i = F_i \times \frac{TRG_{imax} + TRG_{imin}}{2} \tag{4.1}$$

The reason for using F_i as a random number is to help to prevent the total production rate to pass the capacity limit of the production system as much as possible. For instance, by ignoring the factors from the equation 4.1 ($F_i = 1$), the production quantity for all the products would be placed on the middle of the target range which is not a feasible quantity regarding the capacity of the manufacturing system.

Once a population is fully generated in a particular iteration, each chromosome must be decoded in order to get ready for the evaluation procedure. Thinking of a small example can be a good way to achieve a better understanding of the decoding procedure. Table 4.1 shows an example of the problem with the situation in which 5 products must be produced by 2 machines.

Table 4.1: A small example of manufacturing system under consideration

Products	**Machine capacity (bags per minute)**		**Machine efficiency**	
	Machine 1	**Machine 2**	**Machine 1**	**Machine 2**
1	55	55	0.85	0.88
2	0	40	0	0.88
3	50	55	0.85	0.88
4	64	0	0.85	0
5	100	60	0.85	0.88

In table 4.1 it can be seen that for product 1 the theoretical capacity of both machines is similar, but machine 2 can work 0.03% more without failing and eventually can produce around two more bags per minute. It also can be seen that the machine 1 and machine 2 are not eligible for products 2 and 4 respectively. So, the actual capacity of them and consequently their efficiency are zero for these products. The first eligible product which is product 1 is allocated to machine 1. Once machine 1 is done with product 1, the next eligible product is selected to be allocated to it. As it can be seen in table 4.1 product 2 is not eligible for machine 1, so the next one will be product 3. It is noteworthy to point out that depends on the variation in flavours of product 1 and 3 the setup time can differ. If they both have same flavours, the setup time will be only for adjusting the machine for different packaging, but if flavours are also different, the time of cleaning the machine should be considered in the setup time. This can be seen in figure 4.2

that the first two allocations of machine 1 have the same flavours, so the setup time for starting the second allocated product is less than other setup times. After assigning the whole available capacity of machine 1 during the scheduling

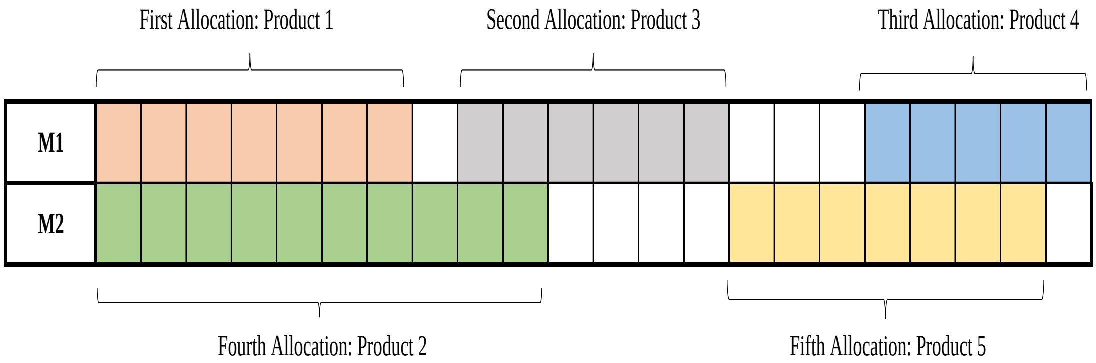


Figure 4.2: Decoding procedure of example table 4.1

horizon, products are assigned to machine 2 based on products eligibility as well as considering the total capacity of the fryer in each time slots. Hence, machine 2 cannot start processing product 5 until finishing product 3 (system capacity in each time slot). Total steps of solution representation can be seen in figure 4.3 as a flowchart.

4.2.1.2 Solution perturbation

Perturbation is implemented on solutions in order to avoid trapping in local optimum points. Although it is performed quite similar in both methods, the steps in which it is done are different between these two algorithms. In the GA, perturbation (mutation) is done after selection and cross-over while in the MPSA, a perturbed solution is a basis for the selection operator. Solution perturbation includes 3 steps as follows:

1. **Swap**: Swap operator randomly selects two genes from the chromosome and exchange them.

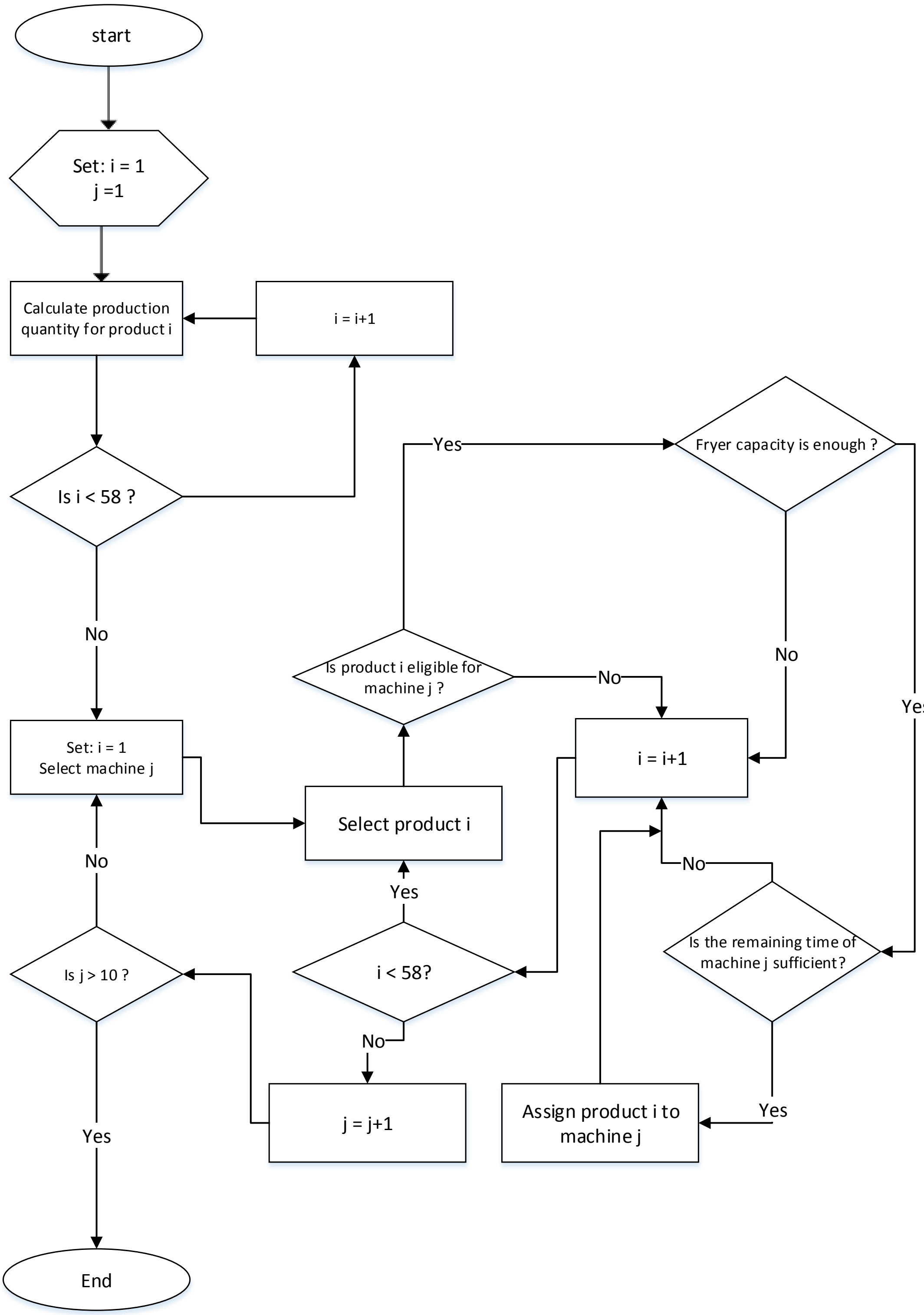


Figure 4.3: Solution representation flowchart

2. **Shift**: In this operator, a gene is randomly selected and shift to a random location.

3. **Factors perturbation**: Here, unlike the last two steps, the perturbation is performed only on product factors and not the whole gene. In the last two steps, the products could either exchange their positions or shift to another position since the indexes of products are limited and no other product is defined in the problem while factors are able to change to any other random numbers within a predefined changeable range.

It is important to highlight that in all three above steps only a certain percentage of genes (products and factors) are perturbed which is based on the perturbation probability defined by the scheduler. Figure 4.4 shows the three steps of the perturbation mechanism.

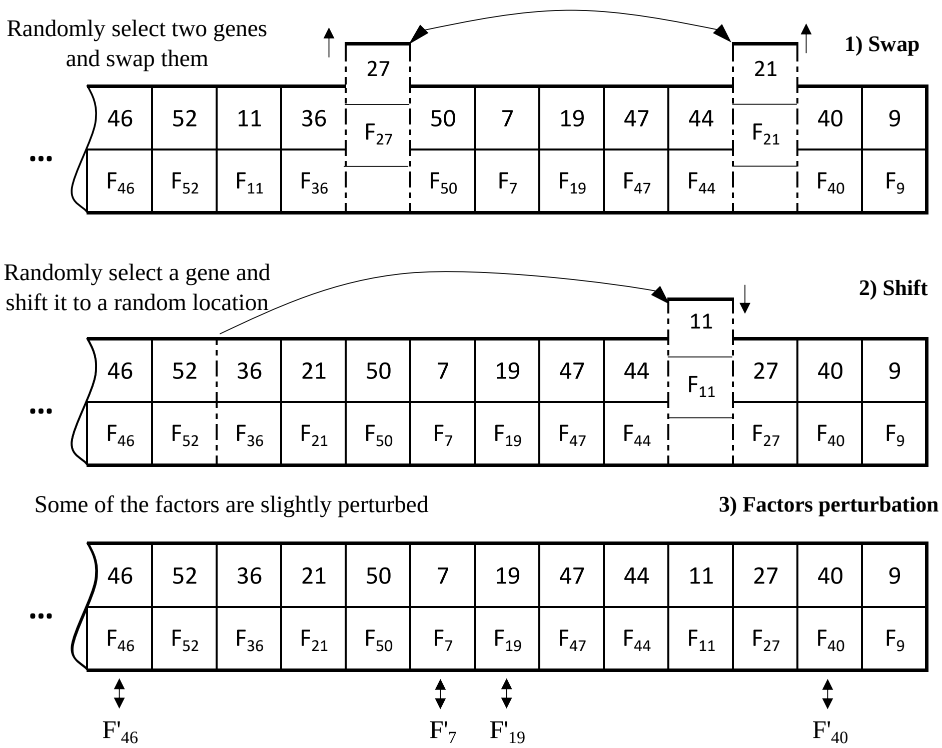


Figure 4.4: Steps of perturbation mechanism

The procedure of factors perturbation has some details that are shown in figure 4.5 . Below are the notations used in the flowchart.

i	Gene counter
P_S	Step Factor Probability
F_{max}	Maximum acceptable value of factors
F_{min}	Minimum acceptable value of factors
F'	New value of F after perturbation
S_{max}	Maximum step that can be taken down or up

The flowchart shown in the figure illustrates that the procedure starts with selecting (P_S) per cent of factors to be perturbed. Then for each selected factor, a random decision is made to decide whether the factor should be stepped up or down. So the factors values do not follow a continuous trend. Factor values cannot change more than the maximum step (S_{max}) which is a predefined value. Then a random number is multiplied by the maximum step and the result is added or subtracted to the current factor value (F_i). However, multiplying a random number into a S_{max} can produce very large factors (while stepping up) and negative ones (while stepping down) which cannot be accepted. To keep the production quantities at target ranges, the factor values need to be between a minimum and maximum value (F_{max} and F_{min}).

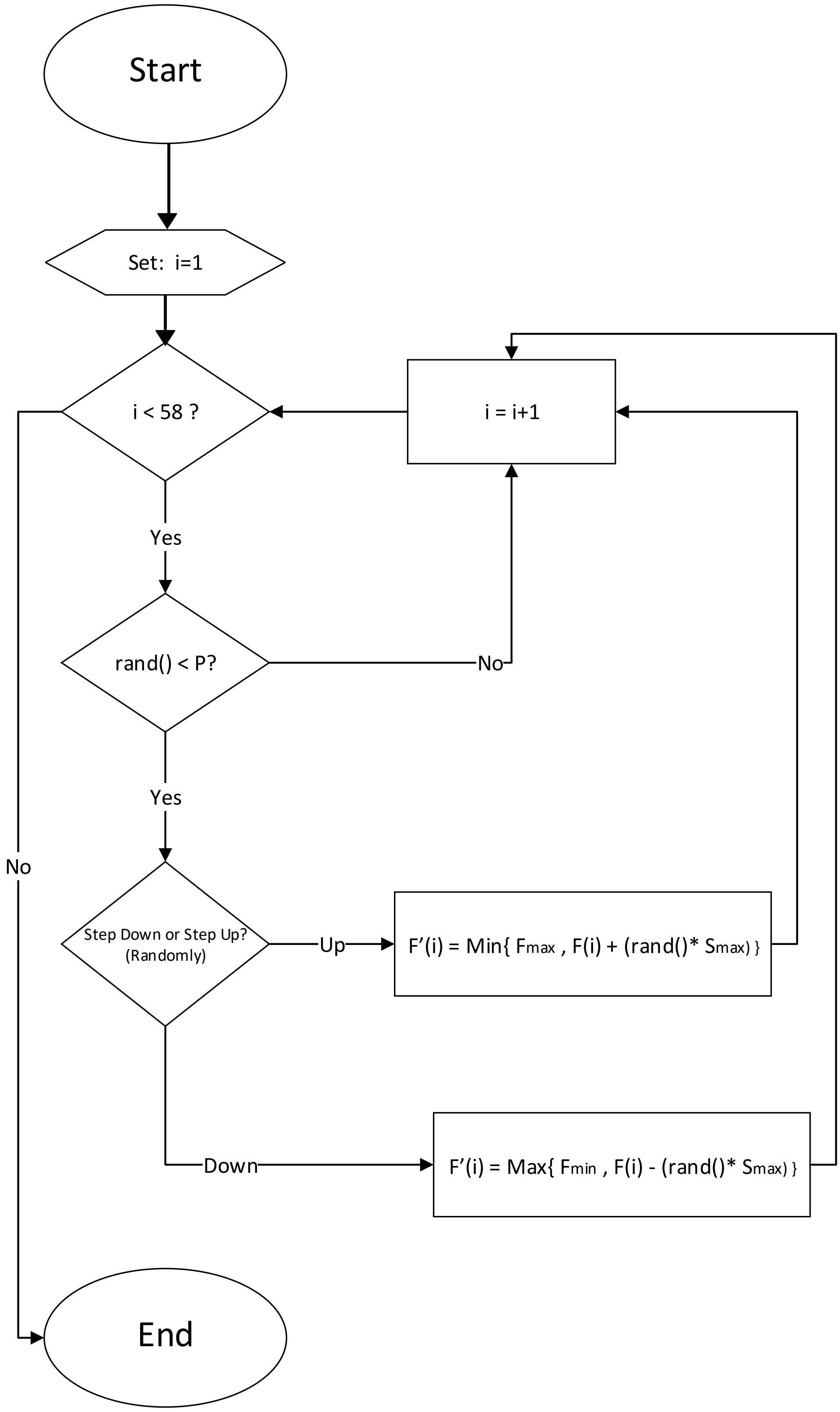


Figure 4.5: Procedure of factors perturbation

4.2.2. Unique steps

In this section, steps that are not common between proposed algorithms are discussed. These steps are either done in different ways (e.g. selection operator)or done only in one of the algorithms (e.g. cross-over).

4.2.2.1 Selection operator

After fitness evaluation, solutions are selected based on their fitness value to be used for next steps. Selection is a procedure which happens in both proposed algorithms. However, the way by which algorithms select solutions are different and the selected solutions are also used in different ways. Following sections explain the selection procedure in both methods.

Genetic Algorithm

In the GA, several methods can be employed as a selection operator such as roulette wheel and tournament selection. In this thesis, tournament selection is used. In this method, k solutions are randomly selected and compared in terms of fitness value, then the fitter one (the one with minimum objective value) is directly copied to the next population. This process is repeated until the number of solutions in the new population equals population size. It should be noted that this process is done with replacements which means every time a fitter solution is selected, it still has a chance to be in the previous population. Hence, in the new population, there may be several copies of a specific chromosome while some chromosomes may not be copied even once. This helps to generate a population with more fitter chromosomes which improves the performance of the GA.

Multiple Path SA

The selection procedure is done quite differently in MPSA. Once a solution in a particular path is perturbed, the algorithm decides which solution should be selected for the next iteration, hence, in MPSA selection operator is also called *Deicide*. The selection is done via the energy function (the weighted sum of the proposed multi-objective problem). If the perturbed solution (X'_n) has lower energy (lower fitness value) than the current best solution (X_n), it would be selected as the solution for the next iteration. However, the converse is not always true, so, if (X'_n) has higher cost than (X_n) while $\exp(-(E(X'_n) - E(X_n))/T_k) > rand()$, ($X'$) would be the solution for the next iteration (X_{n+1}). T_k is the temperature at k^{th} iteration, where several iterations may be done at each temperature. The sequence with which T_k decreases is called *cooling schedule* and should be done in such a way that $T_k > T_{k+1}$ when $\lim_{k\to\infty} T_k = 0$. In general, the selection procedure in single path SA which is similar to the proposed MPSA can be summarized in equation 4.2:

$$X_{n+1} = \begin{cases} X'_n & , \quad \text{if} \quad E(X'_n) \leq E(X_n) \\ X'_n & , \quad \text{if} \quad \exp(\frac{E(X_n)-E(X\prime_n)}{T_k}) > rand() \\ X_n & , \quad \text{otherwise} \end{cases} \tag{4.2}$$

In equation 4.2, r is a randomly generated number which helps stochastic decision for a new solution. Three factors should be considered while scheduling a cooling process :

- The level of starting temperature
- The time of reducing the current temperature
- The amount of reduction in the current temperature

In this thesis, a popular cooling schedule is employed in which a prespecified number of iterations are done at a constant temperature(T_k), then the temperature decreases via equation $T_k = \gamma \times T_{k-1}$, where γ is a constant number in [0,1], called cooling coefficient and usually set as close to 1. Thus, it can be concluded that the parameters of SA are the initial temperature (T_0), the cooling coefficient (γ), and the number of iterations at each temperature.

4.2.2.2 Cross-over

Cross-over is an operator which is exclusively used in the GA. After selection procedure and generating the new population, chromosomes are paired randomly to be combined by cross-over. It is applied to mix the features of each pair of chromosomes to produce new children in hopes of creating chromosomes with higher fitness (Sivanandam and Deepa, 2007). There are different types of cross-overs, which can be used based on the problem type and limitations. In this thesis, since the focus is on a limited number of specific products, the content of genes is fixed and cannot be changed or duplicated. So, the Systematic Single Point Cross-over (SSPC) is employed which has two main steps. The process of implementing the method is as follows:

- **Step 1**:
  In the first step two operations are done. Firstly, a random cross-over point is selected and applied to both selected chromosomes (parents), then all genes before the cross-over point in the parents are exchanged and consequently, two children are produced.

- **Step 2**:
  Since the first element of each gene contains a random number between 1 and 57 (number of products), and the parents are not necessarily in a same order of products, copying genes of one to the other will make duplicate genes(products) which is meaningless. So the second step of the applied cross-over is to remove duplicate genes from both children. Once the exchange is done between the parents, all genes of each parent are compared

to the genes before the cross-over point of the corresponded child. Those genes which are not placed before the cross-over point in the corresponding child are copied to the first vacant gene after the cross-over point in it. This is done until all genes after the cross-over point in both children are filled with new genes. Figure 4.6 shows two steps of cross-over procedure. For the sake of illustration, a chromosome with only 15 genes is shown in this figure.

4.3. Chapter Summary and discussion

In this chapter two metaheuristics were selected and implemented in a case study. Both GA and SA are considered as powerful methods for solving permutation problems with wide search area where implementing direct methods is not reasonable in terms of time required and even modelling the problem. The proposed GA is a typical one with systematic single-point cross-over and tournament selection. Mutation is the same as the perturbation mechanism in the second proposed algorithm (MPSA) where Move, Shift, and Factors Perturbation are done to generate a new solution. In order to improve the performance of conventional simulated annealing in terms of quality of solutions and required time, in this thesis, a population-based simulated annealing is adopted to widen the search area to reduce error probability exponentially while the implementation time increases linearly. Figure 4.7 shows the steps of the applied MPSA in more details, which is inspired by the flowchart proposed in Defersha (2015).

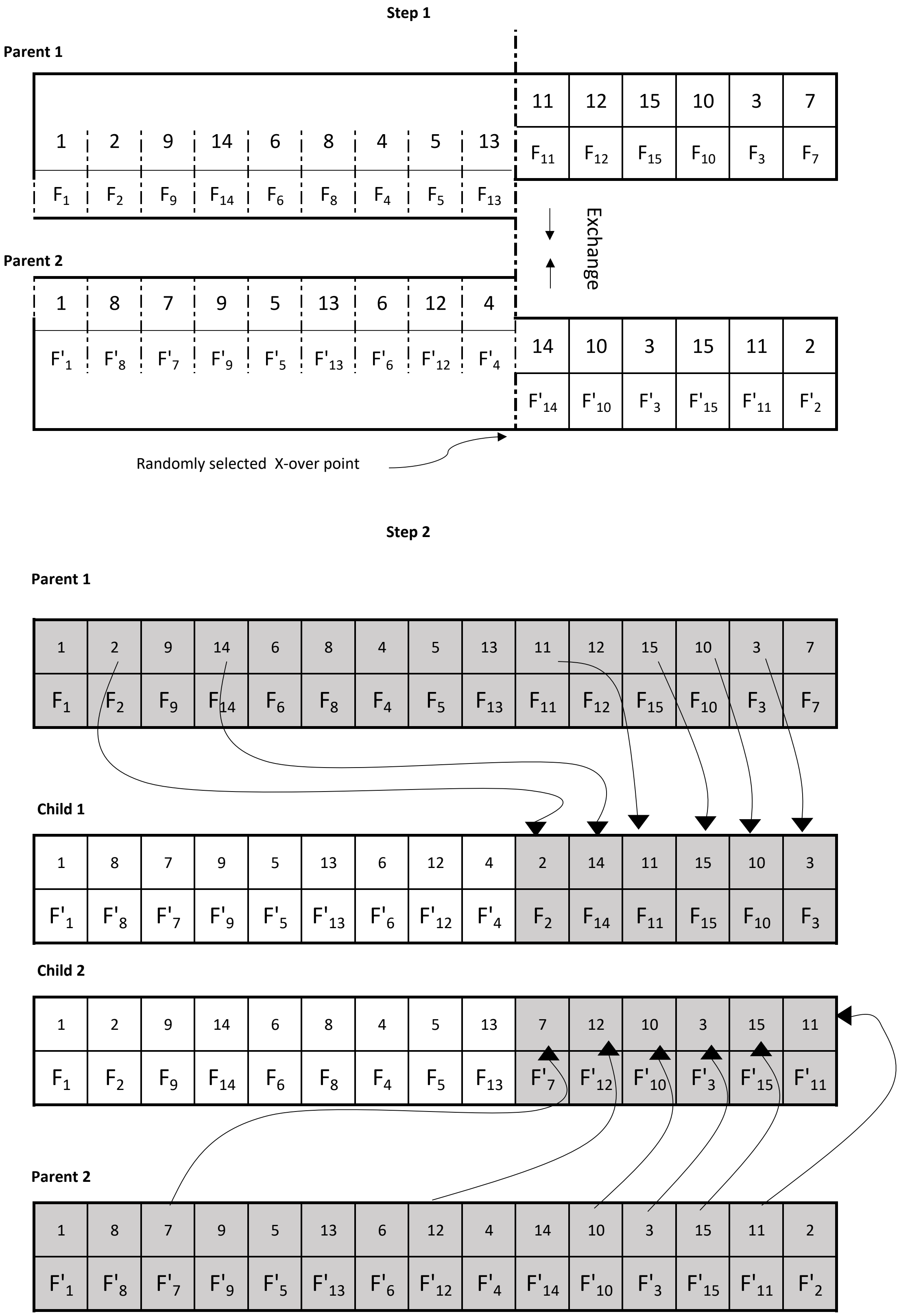


Figure 4.6: Illustration of cross-over procedure

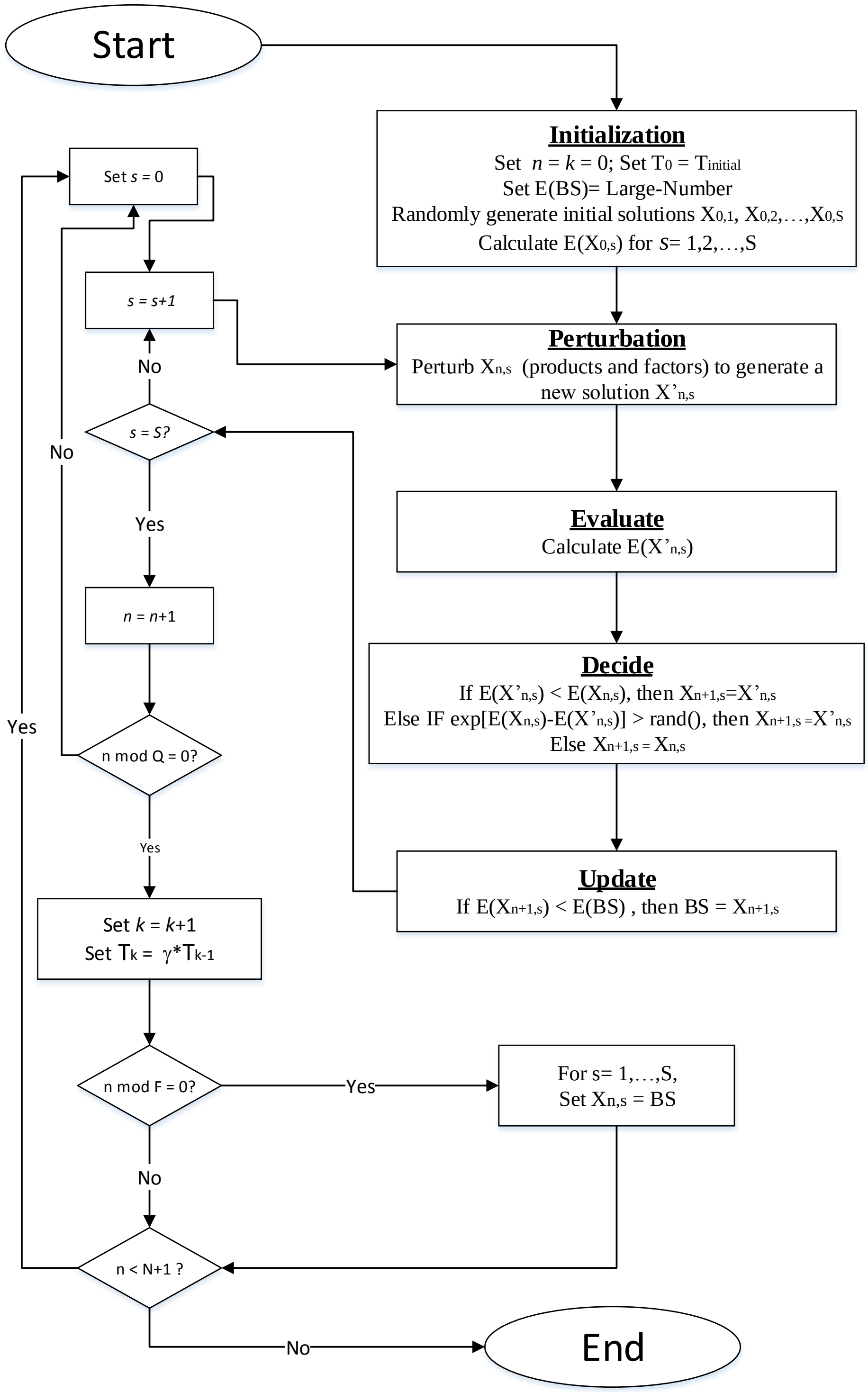


Figure 4.7: Steps of the MPSA

Chapter 5

Numerical Example

In this chapter, firstly the performance of the GA is compared to the manual scheduling method in terms of quality of the solutions, then the effects of altering the parameters will be illustrated in order to find the best ones for both GA and MPSA. At the last step, the performance of both proposed algorithms will be compared in terms of time required and quality of answers.

5.1. GA vs Manual Scheduling

There is no need to say that how time-consuming manually scheduling could be. In this problem, the required time for scheduling the production line takes around one week, while GA can find a good solution in far less required time (almost half an hour). So, the performance measure in this section is only the quality of the solutions for each objective. Figure 5.1 plotted the differences between the manual method and the GA for the first two objectives. Parts (a) and (b) of figure 5.1 shows hourly and average over-pull for the Manual method and the GA respectively. Clearly, the GA has a smaller average over-pull. The second objective of the problem was compared in two methods in parts (c) and (d) of figure 5.1 which shows the difference between target KG and the capacity of the system decreased considerably in the GA (part (d)).

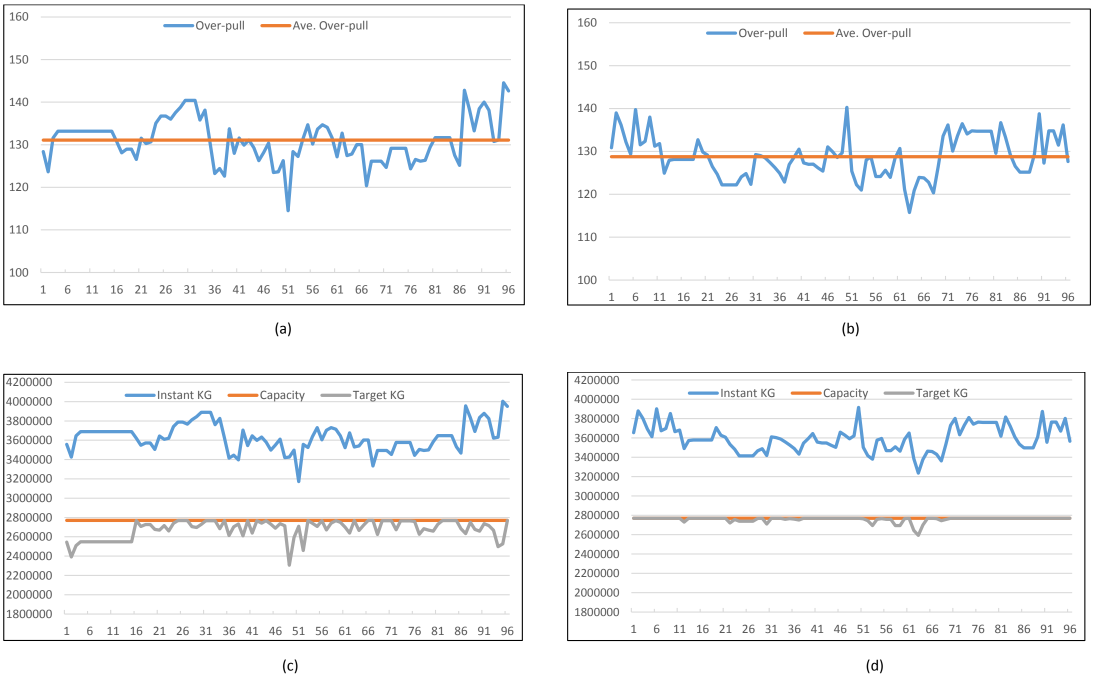
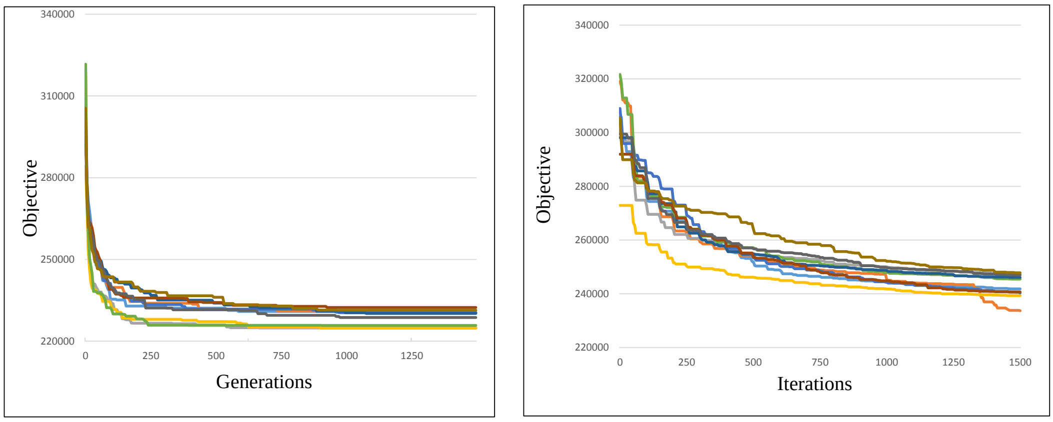


Figure 5.1: Comparison of Manual Scheduling (a and c) and the GA (b and d)

In terms of the third objective, GA also performs better. It could find much better solutions and improve them for around 40 per cent.

5.2. GA vs MPSA

In this section, the performances of the two proposed algorithms are compared. Since both algorithms are metaheuristic, the initial random number that they take to start calculation can alter the search directions and affect the performance of them accordingly. figure 5.2 plotted the trend of both algorithms for ten runs with different seed numbers at which further random numbers are chosen. It clearly demonstrates that the GA performs better than MPSA in both terms of the speed of convergence and quality of solutions. Figure 5.3 also shows the better performance of the GA in terms of the quality of solutions. It should be highlighted that for the MPSA the total number of iterations are considered

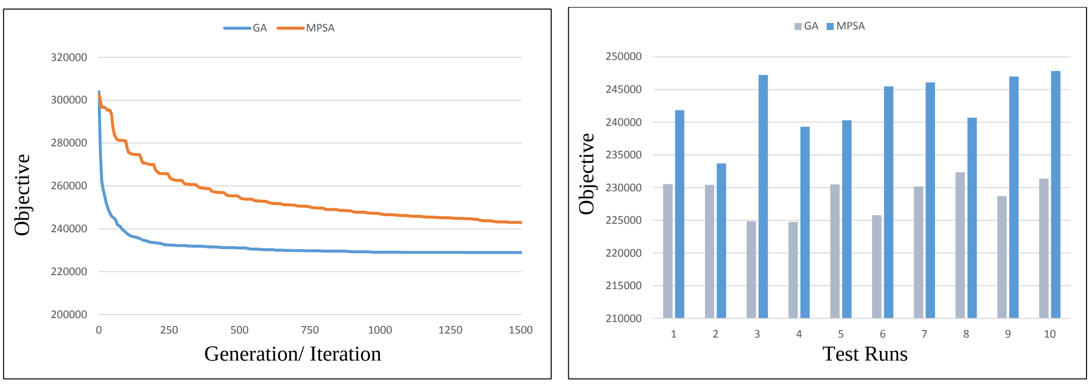


(a) (b)

Figure 5.2: GA (a) vs MPSA (b) in solving the problem with different seed numbers.

against the generations in the GA. The maximum number of iterations or generations are selected based on the convergence of both methods. As it can be seen in figure 5.2, in both methods although convergence has happened at a different number of iterations, all of them are converged before 1500 generations (iterations) and the results do not improve significantly after this number, therefore, this number can be a safe stopping criterion.

Part (a) of this figure shows performance of both methods on average in each iteration, Also in part (b), it is evident that the GA converge to a better objectives in all ten runs.



(a) (b)

Figure 5.3: Performance comparison (a) average convergence, (b) final solution objective.

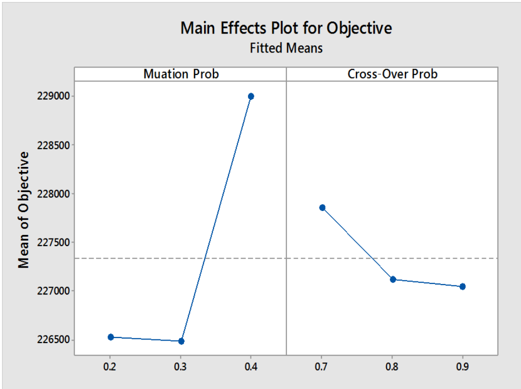
Table 5.1: Results of applying various combinations of cross-over and mutation probability

			Objective value for replications			
Experimental Run	Cross-Over Probability	Mutation Probability	1	2	3	4
1	0.7	0.4	226498	225797	226077	231081
2	0.8	0.4	231243	231970	226374	232071
3	0.9	0.4	230486	230769	225918	229766
4	0.7	0.3	225289	226153	225148	230526
5	0.8	0.3	230419	224755	225785	228697
6	0.9	0.3	224781	228823	223599	223921
7	0.7	0.2	230073	233629	224295	229709
8	0.8	0.2	224039	224227	223457	222362
9	0.9	0.2	221671	228079	227748	229077

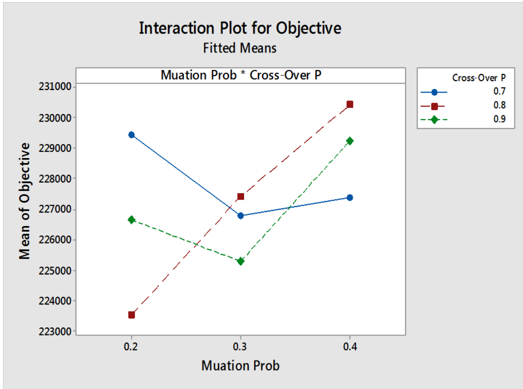
5.3. GA Performance Analysis

In this section the performance of the GA is analyzed in presence of different parameters such as cross-over probability and mutation probability. Also, effect of changing the population size will be analyzed. To investigate the effects of changing the GA parameters on the algorithm, Analysis of Variance (ANOVA) has been implemented. Here, both steps of mutation operator (shift and swap) are considered as a single operator and the probabilities are changed for both of them similarly in each run of the experiment. Table 5.1 shows the data used in 9 experimental runs with four different seed numbers which are counted as the replications of the ANOVA. The parameters were set at three levels as {0.7, 0.8, 0.9} for the cross-over, and {0.4, 0.3, 0.2} for the mutation.

Based on the P-Values of the experiments, (figure 5.4 (c)) the mutation probability has a significant effect on the objectives. This can be seen in main effect plot of the mutation probability (figure (a)), however, in spite of the descending trend of cross-over probability plot the main effect of it is not significant since the corresponding P-value is much higher than 0.05. This means that cross-over cannot affect the results without considering the mutation. Interaction plot (figure (b)) shows an obvious interaction between both factors (probabilities) and it is also confirmed by the interaction P-Value (0.026). Due to the interaction, it



(a)

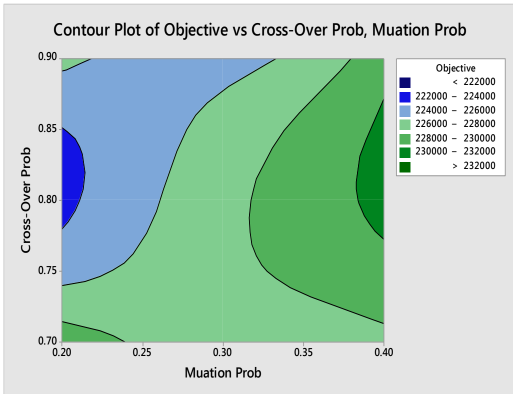


(b)

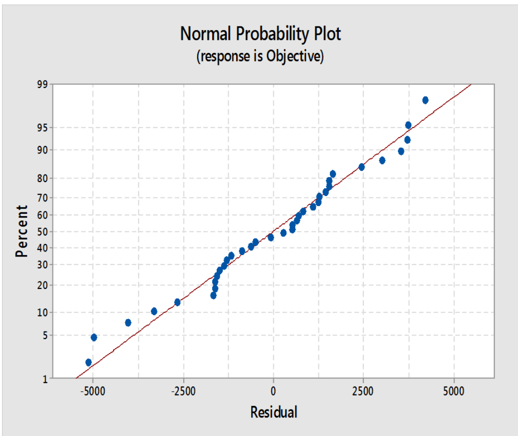
Analysis of Variance

Source	F-Value	P-Value
Muation Prob	3.46	0.046
Cross-Over Prob	0.33	0.719
Cross-Over Prob*Muation Prob	3.26	0.026

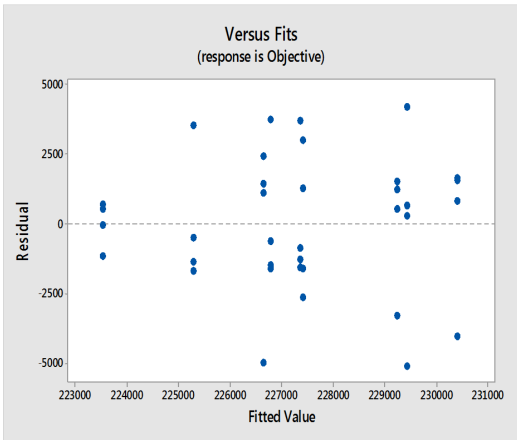
(c)



(d)



(e)



(f)

Figure 5.4: Results of ANOVA for the effects of cross-over and mutation probabilities on GA

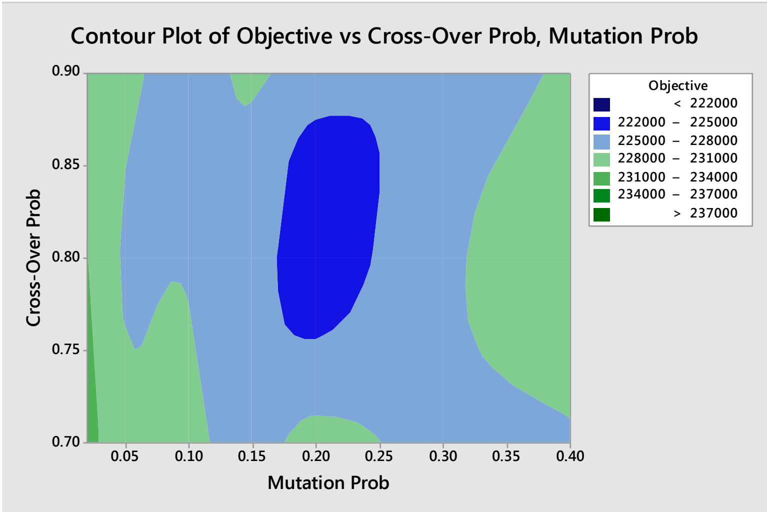


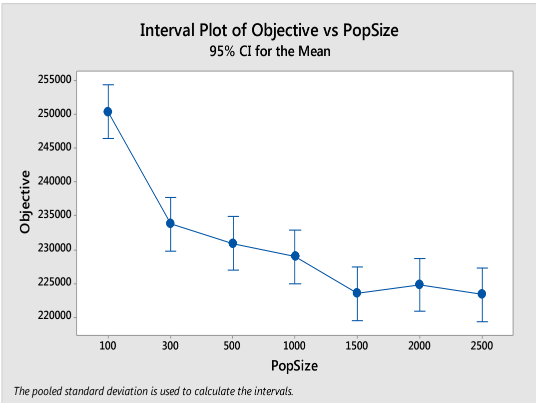
Figure 5.5: Wider range of combination of mutation and crossover probabilities

can be concluded that the performance of the algorithm depends on the factors within the specified range. Figure 5.4(d) shows the results of different combinations of mutation and cross-over probabilities. According to this figure within the specified range of factors, best results are attained when the mutation and cross-over probability are changing between {0.2, 0.22} and {0.78, 0.85} respectively. Since in this figure the total interval for the best results is not demonstrated, another experiment was run and the results were added to the figure 5.4 (d) which can be seen in figure 5.5. Figure (e) and (d) are to check two necessary conditions of running ANOVA for the data. They plot the normality of residuals and equality of variances for the experimented results respectively. So, the accuracy of the results found by the ANOVA is reliable.

One of the parameters that can also have a significant effect on the performance of the GA is the population size. Table 5.2 shows the results of different population sizes which are plotted in figure 5.6. Considering the p-value of the experiment, the null hypothesis that is equality of means of objective values with different population sizes can be rejected. Hence, it can be concluded that increasing the population size can significantly improve the performance of the GA

Table 5.2: Results of applying different population size

Population Size	Objective values for replications			
	Replication 1	**Replication 2**	**Replication 3**	**Replication 4**
100	246631	248892	245292	260805
300	237226	231041	232239	234778
500	233702	231543	226066	232562
1000	230865	229932	223744	231273
1500	224039	224227	223457	222362
2000	224095	229453	223055	222628
2500	222333	221215	227867	222132



(a)

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
PopSize	6	2163015818	360502636	24.86	0.000
Error	21	304579962	14503808		
Total	27	2467595780			

(b)

Figure 5.6: Results of ANOVA for different population sizes

within this range of population sizes. However, since the results do not improve considerably after population size of 1500 (see figure 5.6 (a)), and considering that the processing time increases by growing the population size, 1500 is selected for the size of all experiments on the GA.

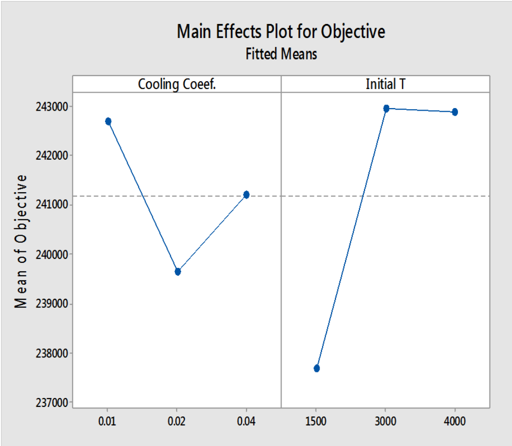
5.4. MPSA Performance Analysis

Similar performance analysis to the previous section has been done for the second proposed algorithm (MPSA) in this section. The parameters selected for this analysis are the initial temperature (T_0) and the cooling coefficient (γ). Again, nine experiments were conducted with different initial random numbers in three various parameter levels in order to gather different results for replications of the ANOVA. Initial temperatures levels are {1500, 3000, 4000} and cooling coefficients are {0.01, 0.02, 0.04}. Results of the runs can be seen in table 5.3.

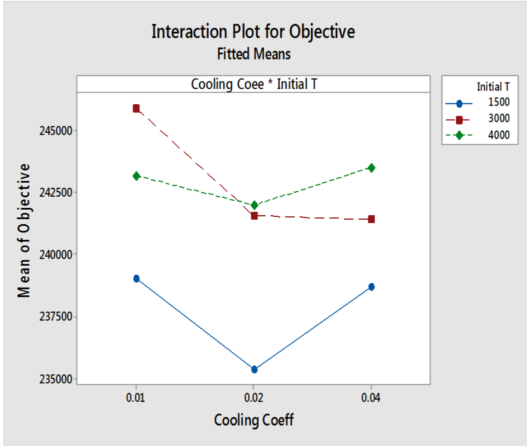
Table 5.3: Results of applying various combinations of initial temperature and cooling coefficient

			Objective value for replications			
Experimental Run	Initial temperature (T_0)	Cooling coefficient(γ)	1	2	3	4
1	1500	0.01	235345	233526	243449	243864
2	3000	0.01	245405	233867	249279	255086
3	4000	0.01	252685	236117	245029	238881
4	1500	0.02	239078	237190	239307	225915
5	3000	0.02	233699	239314	245466	247834
6	4000	0.02	248111	237113	241821	240972
7	1500	0.04	238464	234452	238533	243343
8	3000	0.04	239090	234765	248612	243181
9	4000	0.04	250079	235025	245813	243168

The result of running ANOVA on the MPSA data can be seen in figure 5.7. Figure (a) shows neither positive nor negative trend in main effect plots of both factors. It is also proved by the P-Value of them that are both higher than the significance level $\alpha = 0.05$ (see figure (c)). As it can be seen in the interaction plot (figure 5.7 (b)) except for one point with $T_0 = 4000$ all plots are almost parallel which indicates that there is no interaction between selected factors within their specified range. The P-Value is also 0.928 (figure (c)) which is much higher than the significance level and supports that the interaction between factors is not significant and the evidence for rejecting the null hypothesis (having interaction) are not enough. This demonstrates the robustness of the MPSA algorithm to these two parameters. Wider combination of factors can be seen in the figure (d). Similar to the previous ANOVA, parts (e) and (f) of the figure 5.7 shows the presence of necessary conditions for ANOVA.



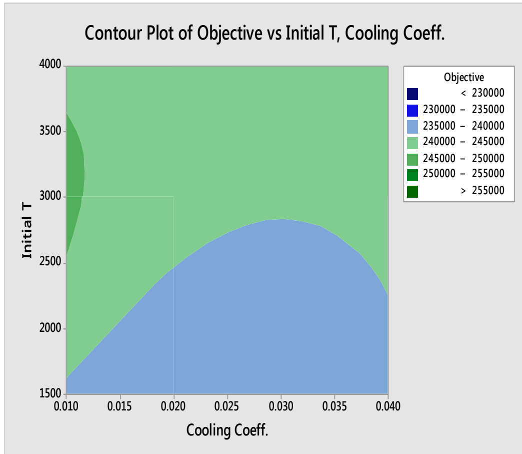
(a)



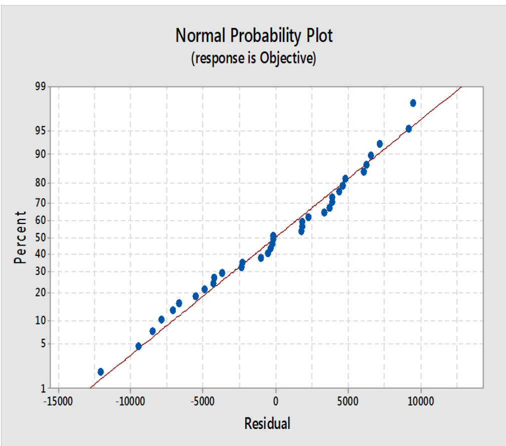
(b)

Analysis of Variance		
Source	F-Value	P-Value
Cooling Coeff.	0.72	0.498
Initial T	2.78	0.080
Cooling Coeff.*Initial T	0.22	0.928

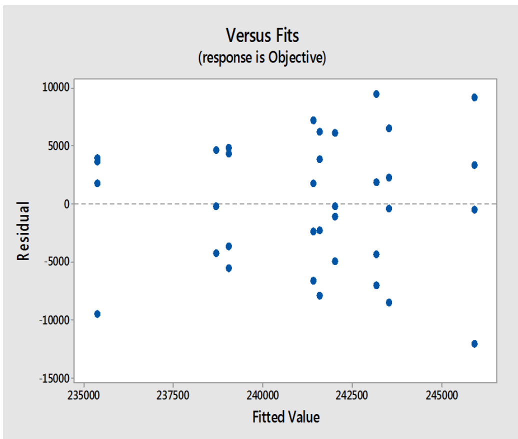
(c)



(d)



(e)



(f)

Figure 5.7: Results of ANOVA for the effects of (T_0) and (γ) on the MPSA

Chapter 6

Conclusions and Future Research

6.1. Discussions and Conclusion

With the huge development in industries and demands for products, manufacturers need to manage the time and resources more thoughtfully. Scheduling is a way to utilize resources in order to minimize their corresponding costs. However, scheduling is not always easy enough to be implemented in a reasonable amount of time, which could make it an overwhelming task. In this thesis, a food processing manufacturing system was studied and the aim was to present two metaheuristics methods in order to find satisfactory schedules and confront the shortages and defects exist in manual scheduling and even the direct methods. Manual scheduling is usually a very time-consuming job and since only some of many details of an optimized schedule are considered inevitably, the solutions do not have enough quality to meet the management criteria. So, in both terms of time and the quality of solutions metaheuristics are incomparable to the manual method. Metaheuristics proposed in this thesis are the Genetic Algorithm and Multiple-Path Simulated Annealing. Both of these methods have shown the ability to considerably improve the time and quality of solutions in comparison to the manual method which was previously used in the industry studied in this thesis. Although in the second proposed method, it was tried to reinforce the simple simulated annealing by changing it to a population-based method (MPSA), the results found by the GA seem to be more acceptable. The GA could find a better solution in a lower number of iterations (faster convergence), while the required time of each iteration of MPSA was slightly less than the GA. Analysis of Variance was run for both algorithms for two different factors to analyze the performance of them. For the GA the best combination of cross-over probability and mutation probability was found, however, MPSA showed more independence to the studied parameters within a specified range and consequently should be considered as a most robust method.

6.2. Future Research and Recommendations

The methods proposed in this thesis prove that in practice metaheuristics can show a quite acceptable performance without ignoring any constraints. However, there are some objectives that have not been considered for the problem studied in this thesis. As a future research, they also can be accounted for the objective functions. First one can be the minimization of setup times. As explained before, setup times between products with different flavours are longer than the ones with similar flavours. Hence, arranging products in such a way that minimum setup times is required might increase the total production rate during the scheduling horizon or even decrease the need for facilities. The second one is the allocation of products to machines. In proposed methods the eligibility of machines for processing different products is taken into consideration based on the efficiency rate of them, however, there might be other free machines with more efficiency for a specific product. Minimizing the number of allocated jobs to machines with lower efficiency can be an objective function for this study. Furthermore, in spite of outstanding performance of GA and acceptable results of SA among metaheuristics in solving NP-Hard problems, improvements in the performance of them might be possible by combining both methods to form a hybrid algorithm. Simulated Annealing can be used as the mutation operator of the GA in order to improve the randomness of the solutions and consequently extend the search space.

Bibliography

Akkerman, R. and van Donk, D. P., 2009. Analyzing scheduling in the food-processing industry: structure and tasks. *Cognition, Technology & Work*, **11 (3)**, 215–226.

Amorim, P., Antunes, C. H., and Almada-Lobo, B., 2011. Multi-objective lot-sizing and scheduling dealing with perishability issues. *Industrial & Engineering Chemistry Research*, **50 (6)**, 3371–3381.

Atighehchian, A., Bijari, M., and Tarkesh, H., 2009. A novel hybrid algorithm for scheduling steel-making continuous casting production. *Computers & Operations Research*, **36 (8)**, 2450–2461.

Bodington, C. E., 1995. Planning, scheduling, and control integration in the process industries. McGraw-Hill Companies,

Chen, H., Wu, C., Zuo, L., Diao, F., Wang, L., Wang, D., and Song, B., 2017. Optimization of detailed schedule for a multiproduct pipeline using a simulated annealing algorithm and heuristic rules. *Industrial & Engineering Chemistry Research*, **56 (17)**, 5092–5106.

Cooke, D. L. and Rohleder, T. R., 2006. Inventory evaluation and product slate management in large-scale continuous process industries. *Journal of Operations Management*, **24 (3)**, 235–249.

Dahal, K. P., Burt, G. M., NcDonald, J., and Moyes, A., 2001. A case study of scheduling storage tanks using a hybrid genetic algorithm. *IEEE Transactions on Evolutionary Computation*, **5 (3)**, 283–294.

Defersha, F. M., 2015. A simulated annealing with multiple-search paths and parallel computation for a comprehensive flowshop scheduling problem. *International Transactions in Operational Research*, **22 (4)**, 669–691.

Erdogdu, F., 2008. Optimization in food engineering. CRC Press,

Ferretti, I., Zanoni, S., and Zavanella, L., 2006. Production–inventory scheduling using ant system metaheuristic. *International Journal of Production Economics*, **104 (2)**, 317–326.

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

Floudas, C. A., Akrotirianakis, I. G., Caratzoulas, S., Meyer, C. A., and Kallrath, J., 2005. Global optimization in the 21st century: Advances and challenges. *Computers & Chemical Engineering*, **29 (6)**, 1185–1202.

Fransoo, J. C., 1993. Production control and demand management in capacitated flow process industries. Technische Universiteit,

Göthe-Lundgren, M., Lundgren, J. T., and Persson, J. A., 2002. An optimization model for refinery production scheduling. *International Journal of Production Economics*, **78 (3)**, 255–270.

Gravel, M., Price, W. L., and Gagné, C., 2002. Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research*, **143 (1)**, 218–229.

Groover, M. P., 2008. Automation, production systems, and computer-integrated manufacturing. Pearson Education India,

Gunasekaran, A., 1998. Concurrent engineering: a competitive strategy for process industries. *Journal of the Operational Research Society*, **49 (7)**, 758–765.

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G., and Wassick, J., 2014. Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, **62**, 161–193.

Hazaras, M. J., 2012. Improving process efficiency through applied process scheduling and production planning optimization. Ph.D. thesis.

Herrmann, J. W., 2006. Handbook of production scheduling. Vol. 89. Springer Science & Business Media,

Hindi, K., 1995. Solving the single-item, capacitated dynamic lot-sizing problem with startup and reservation costs by tabu search. *Computers & industrial engineering*, **28 (4)**, 701–707.

Hou, Y., Wu, N., Zhou, M., and Li, Z., 2017. Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **47 (3)**, 517–530.

Huegler, P. and Vasko, F., 2007. Metaheuristics for meltshop scheduling in the steel industry. *Journal of the Operational Research Society*, **58 (6)**, 791–796.

Jia, Z. and Ierapetritou, M., 2003. Mixed-integer linear programming model for gasoline blending and distribution scheduling. *Industrial & Engineering Chemistry Research*, **42 (4)**, 825–835.

Jia, Z., Ierapetritou, M., and Kelly, J. D., 2003. Refinery short-term scheduling using continuous time formulation: Crude-oil operations. *Industrial & Engineering Chemistry Research*, **42 (13)**, 3085–3097.

Kılıç, O. A., 2011. Planning and scheduling in process industries considering industry-specific characteristics. , .

Kopanos, G. M., Puigjaner, L., and Georgiadis, M. C., 2009. Optimal production scheduling and lot-sizing in dairy plants: the yogurt production line. *Industrial & Engineering Chemistry Research*, **49 (2)**, 701–718.

Kopanos, G. M., Puigjaner, L., and Georgiadis, M. C., 2011. Production scheduling in multiproduct multistage semicontinuous food processes. *Industrial & Engineering Chemistry Research*, **50 (10)**, 6316–6324.

Kopanos, G. M., Puigjaner, L., and Georgiadis, M. C., 2012. Efficient mathematical frameworks for detailed production scheduling in food processing industries. *Computers & chemical engineering*, **42**, 206–216.

Lee, H., Pinto, J. M., Grossmann, I. E., and Park, S., 1996. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Industrial & Engineering Chemistry Research*, **35 (5)**, 1630–1641.

Maravelias, C. T., 2005. Mixed-time representation for state-task network models. *Industrial & engineering chemistry research*, **44 (24)**, 9129–9145.

Mendez, I. E. H. I. K. P., C. A. & Grossmann, 2006. A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. *Computers & chemical engineering*, **30 (4)**, 614–634.

Meyr, H., 2000. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research*, **120 (2)**, 311–326.

Oliveira, F., Hamacher, S., and Almeida, M. R., 2011. Process industry scheduling optimization using genetic algorithm and mathematical programming. *Journal of Intelligent Manufacturing*, **22 (5)**, 801–813.

Pan, Q.-K., 2016. An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling. *European Journal of Operational Research*, **250 (3)**, 702–714.

Pan, Q.-K., Wang, L., Mao, K., Zhao, J.-H., and Zhang, M., 2013. An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process. *IEEE Transactions on Automation Science and Engineering*, **10 (2)**, 307–322.

Pinedo, M., 2012. Scheduling: theory, algorithms, and systems. Springer,

Raaymakers, W. H. and Weijters, A., 2003. Makespan estimation in batch process industries: A comparison between regression analysis and neural networks. *European Journal of Operational Research*, **145 (1)**, 14–30.

Ramteke, M. and Srinivasan, R., 2012. Large-scale refinery crude oil scheduling by integrating graph representation and genetic algorithm. *Industrial & Engineering Chemistry Research*, **51 (14)**, 5256–5272.

Rippin, D., PROCYK, L., and GIBSON, N., 1991. Batch process. *Chemical engineering*, **98 (5)**, 100–107.

Rong, A. and Grunow, M., 2010. A methodology for controlling dispersion in food production and distribution. *Or Spectrum*, **32 (4)**, 957–978.

Roy, R. N., 2007. A modern approach to operations management. New Age International, Santos, A. and Dourado, A., 1999. Global optimization of energy and production in process industries: a genetic algorithm application. *Control engineering practice*, **7 (4)**, 549–554.

Shah, N. K. and Ierapetritou, M. G., 2011. Short-term scheduling of a large-scale oil-refinery operations: Incorporating logistics details. *AIChE Journal*, **57 (6)**, 1570–1584.

Shapiro, J. F., 1993. Mathematical programming models and methods for production planning and scheduling. *Handbooks in operations research and management science*, **4**, 371–443.

Shaw, K., Lee, P., Nott, H., and Thompson, M., 2000. Genetic algorithms for multiobjective scheduling of combined batch/continuous process plants. In: Evolutionary Computation, 2000. Proceedings of the 2000 Congress on. Vol. 1. IEEE, pp. 293–300.

Sivanandam, S. and Deepa, S., 2007. Introduction to genetic algorithms. Springer Science & Business Media,

Toledo, C. F. M., Arantes, M. d. S., and França, P. M., 2011. Tabu search to solve the synchronized and integrated two-level lot sizing and scheduling problem. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM, pp. 443–448.

Toledo, C. F. M., da Silva Arantes, M., De Oliveira, R. R. R., and Almada-Lobo, B., 2013. Glass container production scheduling through hybrid multi-population based evolutionary algorithm. *Applied Soft Computing*, **13 (3)**, 1352–1364.

Toledo, C. F. M., de Oliveira, L., de Freitas Pereira, R., Franca, P. M., and Morabito, R., 2014. A genetic algorithm/mathematical programming approach to solve a two-level soft drink production problem. *Computers & Operations Research*, **48**, 40–52.

Toledo, C. F. M., França, P. M., Morabito, R., and Kimms, A., 2009. Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem. *International Journal of Production Research*, **47 (11)**, 3097–3119.

Van Wezel, W., Van Donk, D. P., and Gaalman, G., 2006. The planning flexibility bottleneck in food processing industries. *Journal of Operations Management*, **24 (3)**, 287–300.

Wu, N., Chu, C., Chu, F., and Zhou, M., 2011. Schedulability analysis of short-term scheduling for crude oil operations in refinery with oil residency time and charging-tank-switch-overlap constraints. *IEEE Transactions on Automation Science and Engineering*, **8 (1)**, 190–204.

Wu, N., Chu, F., Chu, C., and Zhou, M., 2008. Short-term schedulability analysis of crude oil operations in refinery with oil residency time constraint using petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **38 (6)**, 765–778.

Wu, N., Chu, F., Chu, C., and Zhou, M., 2009. Short-term schedulability analysis of multiple distiller crude oil operations in refinery with oil residency time constraint. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **39 (1)**, 1–16.

Xhafa, F. and Abraham, A., 2008. Metaheuristics for scheduling in industrial and manufacturing applications. Springer,

Zandieh, M., Abiri, M., and Rahmati, S. H. A., 2016. Bi-objective scheduling of flexible flow lines: a gradual transition tabu search approach. *Production Engineering*, **10 (4-5)**, 477–488.