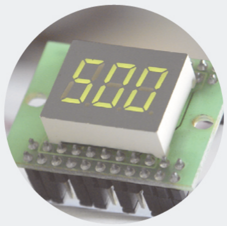
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

isep






Balancing and lot-sizing mixed-model lines in the footwear industry

PARISA SADEGHI
Novembro de 2015

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Balancing and lot-sizing mixed-model lines in the footwear industry
-Master Dissertation-

Author: Parisa Sadeghi
Supervisor: Professor José António Soeiro Ferreira
Co-Supervisor: Professora Ana Maria Marques de Moura Gomes Viana

Mestrado em Engenharia Electrotécnica e de Computadores - Sistemas de Planeamento Industrial
October-2015

Abstract

This report describes the full research proposal for the project *"Balancing and lot-sizing mixed-model lines in the footwear industry"*, to be developed as part of the master program in Engenharia Electrotécnica e de Computadores - Sistemas de Planeamento Industrial of the Instituto Superior de Engenharia do Porto.
The Portuguese footwear industry is undergoing a period of great development and innovation. The numbers speak for themselves, Portugal footwear exported 71 million pairs of shoes to over 130 countries in 2012. It is a diverse sector, which covers different categories of women, men and children shoes, each of them with various models. New and technologically advanced mixed-model assembly lines are being projected and installed to replace traditional mass assembly lines. Obviously there is a need to manage them conveniently and to improve their operations. This work focuses on balancing and lot-sizing stitching mixed-model lines in a real world environment. For that purpose it will be fundamental to develop and evaluate adequate effective solution methods. Different objectives may be considered, which are relevant for the companies, such as minimizing the number of workstations, and minimizing the makespan, while taking into account a lot of practical restrictions.
The solution approaches will be based on approximate methods, namely by resorting to metaheuristics. To show the impact of having different lots in production the initial maximum amount for each lot is changed and a Tabu Search based procedure is used to improve the solutions. The developed approaches will be evaluated and tested. A special attention will be given to the solution of real applied problems. Future work may include the study of other neighbourhood structures related to Tabu Search and the development of ways to speed up the evaluation of neighbours, as well as improving the balancing solution method.

Acknowledgements

All the work that was developed until the present day was just possible with the strong motivation and valuable guidance of my advisor Professor José António Soeiro Ferreira from University of Porto. Furthermore, Professor Ana Maria Marques de Moura Gomes Viana also guided me in the work. I want to thank you a lot for all your time given to me, to accomplish this project. Besides I also want to thank, Engineer Rui Rebelo who provided me with great support during the hardest part of the project.

Contents

Chapter 1

Introduction

1.1 Motivation and Objectives

The footwear industry has been evolving immensely. In the past there were large quantities of a few models being produced, but the situation changed when this industry started depending much more on fashion, which led to reduced orders of various models. Portugal is one of the major players in the footwear industry, ranking 6^{th} worldwide and 3^{rd} in Europe. The Portuguese footwear industry has had a successful journey over the past decades which translates into an export rate of around 95%, a large amount of product added with the second highest price in the world.

This business needs to cover key elements that include small quantities of wide model varieties, graph sequencing of tasks, limited deadlines, multi-functional operators and in-process buffers. Generally, the combination of models produces changes frequently. As a result, the work plan is constantly varying, and to cope with this situation the traditional flow lines are steadily being replaced by more flexible and sophisticated systems. Therefore, shoe manufacturing is advancing technologically and must be very innovative.

Furthermore, it is important to have efficient and effective production lines, and that is why this research will be focusing on balancing and lot sizing matters.

The work to be presented, naturally faces various difficulties, due to the necessity of dealing simultaneously with different problems, such as, job shop problem, line balancing, line scheduling, lot sizing and equipment selection. In general, the problems may be considered as combinatorial optimization problems.

Footwear manufacturing typically involves cutting, stitching and assembly processes, and prior to stitching and assembly there are other processes to prepare the workpieces to be sent to the following line. In the stitching line, the workpieces are put inside a box, which can move in any direction. Therefore, lot-sizing is a subject which will be addressed here.

The problems of line balancing and lot-sizing are a well-known and hard combinatorial optimization problems. During the last decades, these problems have captured the interest of a significant number of researchers and many solution methods have been proposed, but there is still work to be done as most research deals with particular difficulties. Furthermore, other than the scientific part, the aim of this work is to find a solution which may be useful real applications. On the other hand we face it as quite challenging and complex.

The main objective of this dissertation is considering a new and advanced (stitching) line and devising balancing and lot-sizing solution methods to automatically find good results in a short time. That will be achieved by first creating an initial solution for line balancing using an adaptation of the Ranked Positional Weighted(RPW) method, then by scheduling the line using one of the dispatching rules which is Critical Path(CP), expecting to reduce the makespan and, finally, using one method based on the concepts of the metaheuristic Tabu Search (TS), to improve the initial solution generated by the balancing algorithm. The lot-sizing problem has, as objective, minimizing the makespan. The TS method will take into account different neighbourhoods. For this method to be effective,two neighbourhood spaces are considered.

Keeping an exact module in the algorithm for balancing makes it more flexible and easier to be adapted to model extensions. This is an important feature, since companies are constantly changing their requirements and thus constraints may be added or removed frequently. Moreover, different plants may have different specificities, with different constraints. Therefore, a flexible algorithm can be easily adapted for solving different problems. Additionally, as the solution procedure is very fast, it could be used for real world cases.

This dissertation and the case study are specially motivated by significant developments and improvements taking place in a Portuguese footwear company.

1.2 Dissertation Synopsis

The motivation and objectives of this dissertation were presented in the previous section. The work developed to achieve these objectives is divided into six chapters, presented below.

Chapter 2 introduces the footwear industry. The production process in detailed and explored in order to understand its complexity and difficulty for the planning process.

Chapter 3 gives some theoretical basis for the Assembly Line Problem (ALP). It is thus presented an overview of various lot sizing and scheduling problems. Generic solution methods (exact methods, heuristics and metaheuristics and hybrid methods) are briefly explored and discussed.

In Chapter 4 a case study of a real Portuguese foot wear company is discussed. The lot sizing problem is the focus of this dissertation.

The whole solution procedure for the defined problem is addressed in Chapter 5. A constructive heuristic is used to create an initial solution for the ALP and then according to company desire lots are created furthermore a TS is used to improve the solution. Finally, the overall algorithm framework is presented.

Chapter 6 presents the computational results of all the tests performed with the different instances and check and validated them. The input instances are derived from real data from the Portuguese company of the study case. The instances vary in size (of some elements such as the number of products, available operators skill and etc).

Finally, in Chapter 7 conclusions regarding the dissertation are made and directions for future work are pointed out. The suggested future work include both academic and practical activities.

Chapter 2

The Footwear Industry

2.1 An Overview

The footwear sector is a diverse industry which covers a wide variety of materials (textile, plastics, rubber, and leather) and products ranging from different types of men's, women's, and children's footwear to more specialised products like snowboard boots and protective footwear.

In 2012, the footwear sector included around 21,000 enterprises, generated EUR 24 billion in turnover, and produced EUR 6.2 billion in added value (around 0.5 percent of total EU manufacturing). The industry directly employs 280,000 people. Two thirds of total EU footwear production is concentrated in three countries: Italy, Spain, and Portugal [1]. Therefore, in Portugal the footwear industry is a major process industry in the current global economy.

Portugal holds one of the highest export shares, lying consistently among the twelve top exporters over the last decades. Footwear was able to overcome the negative trend registered over the previous three years in 2014. According to specialists, growth in demand was already evident and it was mainly for inexpensive imported footwear. While the domestic production of footwear was of high quality and usually expensive, imports were characterised by low prices and low quality. Meanwhile, as consumers regain purchasing power, demand for footwear is expected to increase. Also, as disposable incomes grow, it is likely that consumers will become more demanding, looking for products of higher quality [1].

However, the number of companies and employment in the footwear sector has been declining in the past decades due to manufacturing moving to economies with lower labour costs.

To sum up, shoe industry is changing a lot these days, in past there were large quantity of few different models, but since this industry is much related to the fashion, they reduce orders of many different models. Since there are lots of the models at the same time on the line and each has different delivery date. Producing shoes in this situation should be under a lot of innovation.

2.2 Footwear Production Process

Shoes are further divided into many categories such as athletic shoes, classic shoes, such as high heels, school shoes and many others. Shoemaking can be considered a traditional handicraft profession. However, now it has been largely taken over by industrial manufacture of footwear. A variety of materials are used for making shoes.

Typically producing shoes involves some processes such as cutting, stitching and assembly and by changing the desire of people this industry changes toward producing lots of models, so there are various models at the same time in the line and each has different routing with various tasks and different processing times.

In the cutting section the top part of the shoe or the ”upper” is made. Materials cutting for preparing to be sent to the next line. Since they are expensive if it will be done manually, this operation needs a high level of skill. Nowadays, some companies have automatic cutting machinery, using high quality software to determine good nestings of the pieces to be cut, and thus reducing material costs.

The next section is pre-stitching which prepares workpieces for the stitching line.

Upper part of a shoe is produced in the stitching line, the sewing machine is used to sew the three dimensional upper part. Various edge treatments are also done onto the leather for giving an attractive look to the finished upper. At this stage, operator expertise are more important (see Figure 2.2).

Figure 2.1: Stitching workstation.

The completed uppers are molded into a shape of foot with the help of a *Last*. Last is a plastic shape that simulates the foot shape. It is later removed from the finished shoe to be used further in making other shoes. Finally, an insole to the bottom of the last is attached in the assembly line(see Figure 2.2).

To give them an attractive finish and to ensure that the edge is waterproof, they are polished and waxed. The bottom of the sole is often lightly buffed, stained and polished and different types of patterns are marked on the surface to give it a craft finished look. A ”finished shoe” has now been made.

The shoe is then stamped in the name of the customer. In packaging the shoe is graded according to the different sizes, packed in boxes, the boxes are labelled using barcodes and dispatched to the customers.

Entering in the shoe industry means competition and marketing is another aspect which involves in the footwear manufacturing. Therefore for being successful creativity is needed. Most people buy shoes primarily because they like the style or function of the shoe. However, customer service also plays a role in a customer’s decision to buy.

Figure 2.2: Assembly line.

So developing a marketing strategy around what sets your customer service apart can help company to attract business.

There are different client types such as online buyers and customers of store of the company. They create first and second categories, the next group is related to the shop sellers which are named as big clients. Therefore, regarding to production plan, there are three types of clients for the companies, big clients, small clients and clients from virtual stores. Orders of virtual store clients are usually put difficulties on the production plan, first because quantity of them are less and second due date restriction.

Usually, the orders should be received one season earlier, but when the company receive orders from the virtual store enough raw materials for production should be available otherwise they cannot satisfy the customer demands. Therefore, based on the prediction they should order the raw material which are needed and also recognize suppliers.

In this industry both manpower and machine are important and since the production plan change rapidly answering to the question that how many resources are needed is a crucial issue.

This work essentially concentrates on the stitching line and the way of allocating resources. Furthermore, as in the stitching lines to be considered the workpieces are put in boxes a lot-sizing problem is also dealt with.

Chapter 3

Assembly Line

This chapter is divided into three main parts; firstly, the balancing and distinct classification will be explained. The second part is related to Scheduling and Lot-Sizing (S&LS).

Then explanations are provided about different solutions such as exact and approximate methods (heuristics and metaheuristics) to tackle the problems that are in the previous sections.

3.1 Balancing

The Assembly Line Balancing Problem (ALBP) consists of assigning tasks to the workstations to optimize some functions, while satisfying the precedence constraints; ALBP is a short-term planning issue, while designing assembly line, which is a balanced process, is a long-term issue. There are two types of balancing: vertical and horizontal. Vertical balance makes the processing time in different workstations of the line similar, while the horizontal balance makes the processing time of different items in the same machine similar [2].

Various assumptions for this problem could be made and each of them is translated into a constraint, for instance, common tasks of different models must be assigned to the same workstations. Some of these constraints depend on the type of line, for example, a line with parallel workstations is different from single workstations. Furthermore, in some companies each operator can work with more than one machine, while that does not happen in other companies. Generally, different decision variables and constraints make this problem difficult to solve.

This problem in the literature is divided into some categories. In the top level there are simple assembly lines and general assembly lines [3].

The Simple Assembly Line Balancing Problem (SALBP) is the most common one, and assumes one homogeneous product. Generalized Assembly Line Balancing Problem (GALBP) is related to multi-model, mixed-model and U-lines. A multi-model line produces different products, while a mixed-model line produces different models of the same product simultaneously. The U-shape Assembly Line Balancing Problem (UALBP) considers the case of U-shaped (single product) assembly lines, where stations are arranged in a narrow U. As a consequence, workers are allowed to work on either side of the U, that is, in initial and subsequent tasks in the production process simultaneously.

The problem addressed here although with special characteristics, is the mixed-model, which is a kind of GALBP. Table 3.1 shows the classification of the GALBP [3].

Classification of the GALBP
Mixed-model line
Multi-model line
U-lines
Others

Table 3.1: The classification of the GALBP

By considering the processing times there are other categories for this problem. When the task performance is constant, it is deterministic; on the other hand, when the processing times are variable, they are stochastic. This version represents the manual assembly lines more realistically where task performance times are seldom constant. This is especially true for complex tasks [4]. Therefore, by considering this concept the ALBP problem is divided into four categories: Single Model Deterministic (SMD), Single Model Stochastic (SMS), Multi/Mixed-Model Deterministic (MMD), Multi/Mixed-Model Stochastic (MMS) [4]. As a result, by taking into account the case study, since the processing times in this line are deterministic this case is an MMD problem. However, in [5] varying task time is defined to another category. Task time changes due to several reasons, for example, when a task is given to a manual labour he can perform the task in some duration of time and after getting an experience in doing that task, he may take less time to perform the same task.

Another category could be depicted by way of movement. There are basically two distinct, non-mechanical and mechanical lines. The mechanical lines are divided into moving line, paced line and unpaced line [6].

Operators on non-mechanical lines are normally free of any mechanical pacing effect, but in a *"moving line"* a transport system moves (at a constant speed) the units evenly distributed along the line. In a *"paced line"* the transport system is periodically moving: when a unit arrives at a station, it remains there for a period of time called *"cycle time"*, and then it is taken to the following station. An *"unpaced line"* is equipped with buffers located between stations. At each station the operator takes a unit from the buffer upstream, performs all the required assembly tasks and then moves the unit to the buffer downstream [6]. By considering the mechanism of the movement, the stitching line is an unpaced line, but the assembly line is a kind of moving line. The classification of the assembly line is shown in figure 3.1.

3.1.1 Single Model

Single-model lines are suitable for large-scale production since they ensure quite low production costs [6]. There are various objectives for this kind of problem and according to those objectives, it is possible to define four versions of this problem.

1. SALBP-1 minimizes the number of stations according the cycle time;

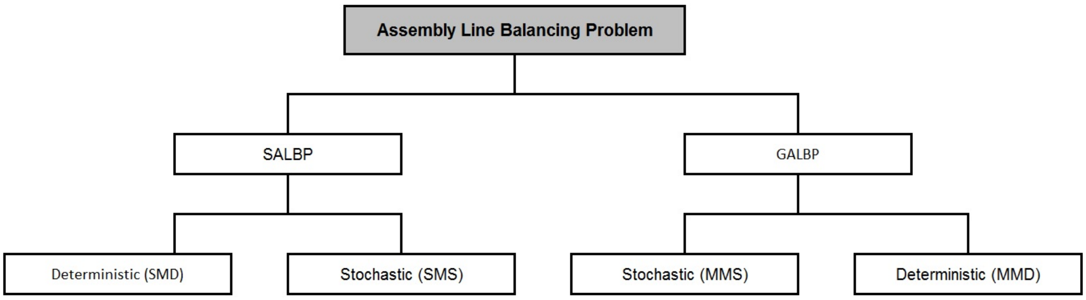


Figure 3.1: Classification of the ALBP.

2. SALBP-2 minimizes cycle time for a given number of workstations. These two versions are the dual of each other [7], but in the real world instance the cycle time is usually given and the aim is to minimize the resource as much as possible so that the SALBP-1 is used more frequently;

3. The goal of the SALBP-E to maximize the line efficiency and minimize the number of workstations and cycle time at the same time;

4. The goal of the SALBP-F is to find a feasible solution;

   Table 3.2 shows four versions of the SALBP based on the number of workstations and cycle time [7].

5. The objective of the SALBP-3 is to maximize the workload smoothness for a given number of workstations. This version is defined based on [8];

   Two other versions of this problem are introduced in [9]:

6. SALBP-4 corresponds to the maximization of workload relatedness;

7. SALBP-5 is conducted for multiple objectives.

	Cycle time	
	Given	Minimize
Number of workstations		
Given	SALBP-F	SALBP-2
Minimize	SALBP-1	SALBP-E

Table 3.2: Various versions of the ALBP according to the objectives.

Note that all types above are also available for different versions of the Mixed-model Assembly Line Balancing Problem (MALBP), for instance MALBP-1, MALBP-2.

All these type of SALB are based on a set of limiting assumptions ([10],[11], [7]), [2]:

- (S-1) Mass-production of one homogeneous product;
- (S-2) All tasks are processed in a predetermined mode (no processing alternative exists);
- (S-3) Paced line with fixed common cycle time according to a desire output quantity;
- (S-4) The line is serial with no feeder lines or parallel elements;
- (S-5) The processing sequence of tasks is subjected to precedence restrictions;
- (S-6) Task times are deterministic (and integral);
- (S-7) No assignment restriction of tasks besides precedence constraints;
- (S-8) A task cannot be split between two or more stations;
- (S-9) All stations are equally equipped with regard to machines and workers

Nearly all of these assumptions have been relaxed or somehow modified by various model extensions considered in the literature [2].

- (S-1) Turns to (A-1) - the products to be manufactured (one or more) are known with certainty;
- (S-2) Turns to (A-2) - a set of processing alternatives is given;
- (S-3) Turns to (A-3) - the line is configured such that target production quantities are satisfied for a certain planning horizon. This might be realized by setting the (average) cycle time(s), and thus production rate(s), or by seeking to produce as much as possible if maximum sales are not a limiting factor;
- (S-4) Turns to (A-4) - the line flow is unidirectional;
- (S-5) Equal to (A-5) - the processing sequence of tasks is subject to precedence restrictions;
- (S-6) to (S-9) are relinquished.

As mentioned above, there are different methods for solving this problem. Figure 3.2 shows the subgroup of exact methods for the SALBP-1 and the SALBP-2 based on [7].

SALBP-1: There are various mathematical formulations for this kind of problem, but the first mathematical formulation in this field was published by Salveson (1955) who tried to find the optimum solution of the linear programming. Different operation researchers have tried to solve the SALBP-1 with Exact methods, but when the numbers of tasks are large, all exact algorithms fail in the sense that the CPU times increase rapidly [10] and the solutions of this type can be divided into two categories: *"B&B"* and *"Dynamic Programming (DP)"* [7].

There is another definition for the line balancing problem which shows that this problem could be solved using the bin packing problem.

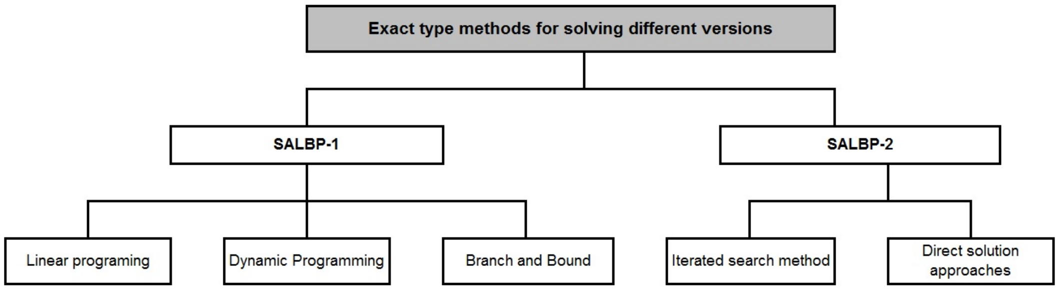


Figure 3.2: Different categories of exact methods for solving SALBP-1 and SALBP-2

Given a directed acyclic graph *G=(T,P)* (nodes T representing the tasks and arrows P representing the precedence constraints) with a constant L_i (task length) assigned to each node T_i, a constant C (the cycle time) and a constant N, nodes T can be partitioned into N or less subsets S_j (the j^{th} station's tasks) such that (i) for each of the subsets, the sum of Lis associated with the nodes in the subset does not exceed C, and (ii) there exists an ordering of the subsets such that whenever two nodes in distinct subsets are joined by an arrow in G, the arrow goes from a higher-ordered (earlier) to a lower-ordered (later) subset [12].

The lower bound on the number of workstations in SALBP-1 is calculated by various researchers. There are different ways of calculating this bound: one of them is the lower bound of the bin packing problem which is equal to the lower bound of this problem [13].

Different exact algorithms try to find the solution by applying the constructive scheme, and assigning subset of tasks to the workstations.

As previously mentioned, the DP procedure is one of the methods for solving this problem in an exact way. This procedure focuses on a workstation and allocates tasks until it becomes full, and then starts on another workstation. The SALBP-1 is transferred to the shortest path problem; each path in this graph corresponds to a feasible solution and each shortest-path corresponds to an optimal solution [14]. In this article, an algorithm based on finding a shortest-route in a finite directed network is provided to the assembly line balancing problem. Arc lengths are so complete that it is sufficient to find any path from the origin to the destination node containing a minimal number of arcs. Computational results are presented and the algorithm is compared using prior analytical line balancing methods. However, applying this procedure to large problems is time consuming.

By increasing the size of the tasks, the time for running the algorithm dramatically increases. Therefore, by using a storage-saving variant improvements occur and the necessary time and memory are reduced [15].

The B&B procedures are frequently used in the literature, and there are different ways of solving this problem by using the B&B method.

Heuristic procedures: this part is related to the heuristic procedures which do not guarantee an optimal solution and their goal is to find reasonably good solutions.

The SALBP is a class of the NP-hard optimization problems, and effective and exact heuristic procedures are available to solve medium-sized instances with enough quality for practical use. However, further algorithmic improvement is necessary for solving large-scale instances [7].

For the SMD version of the problem, most of the time the existing heuristic procedures provide near-optimal, if not optimal solutions [4].

The complexity of the ALBP renders optimum seeking methods that are impractical for instances of more than a few tasks and/or workstations. If there are *"m"* tasks and *"r"* preference constraints, then there are *"m!/2r"* possible task sequences [10]. Therefore, it can be time consuming for optimum seeking methods to obtain an optimal solution within this vast search space.

Different metaheuristic procedures which some of them will be explained in section 3.3 are used to solve proposed problem, Genetic Algorithm (GA) received an increasing attention from the researchers since they provide an alternative to traditional optimization techniques by using directed random searches to locate optimum solutions in complex landscapes [9].

Priority rule based constructive procedures for SALBP-1, Station-oriented procedures and task-oriented procedures are mentioned in [7].

Station-oriented procedures: start with the first station *(k=1)*. The following stations are considered successively. In each iteration, a task with highest priority which is assigned to the current station k is loaded maximally. When station k is loaded maximally, it is closed, and the next station $k+1$ is opened.

Task-oriented procedures: a task with highest priority is chosen among all available tasks and assigned to the earliest station to which it is assignable. Depending on whether the set of available tasks is updated immediately after assigning a task or after assigning all currently available tasks, task-oriented methods can be subdivided into immediate-update-first and general-first-fit methods.

In [16], a heuristic method is used for SALBP-1 and SALBP-2, but the general attitude is station-oriented. An improvement procedure is done with TS.

A new heuristic algorithm and new reduction techniques are presented in [17] for the SALBP-1. The new heuristic is based on the well-known Hoffmann heuristic and builds solutions from both sides of the precedence network to select the best. The reduction techniques aim at augmenting precedence, conjoining tasks and increasing operation times.

The heuristic is tested on its own and also combined with the reduction techniques. The tests, which are conducted on a well-known benchmark set of problem instances, confirm the efficacy of the combined algorithm, in terms of both solution quality and optimality verification, as well as its computational efficiency. This makes it one of the most effective heuristics available for SALBP-1 [7].

There are also other methods for solving this kind of problem in approximate ways.

Genetic algorithm: An efficient GA for SALBP-1 is introduced in [12], demonstrating the algorithm's ability to handle real world data sizes.

Furthermore, a GA for solving SALBP-2 is studied in [18], where the authors try to demonstrate the effective use of GAs for solving combinatorial optimization problems. Firstly, the authors described a fairly typical serial implementation of GA for the ALBP and studied the effects of various GA variables on the performance of the GA. Secondly, they introduced an alternative parallel version of the GA, where each individual in the population resided on a processor. The comparative study between serial GA and parallel GA demonstrated that the quality of the solutions from the parallel implementations was worse than the best solutions obtained from the serial implementation [9].

In [19], the authors examined the application of GA to the SALBP-1. This study provides a comparison between the performance of the GA and a traditional solution method, that is, the ranked positional weight (Helgerson and Birnie, 1961). In [20], the authors performed computational experiments in order to identify suitable genetic operators and parameter values. These two papers complement each other [9].

In [21], the authors used the GA for the two-sided assembly line, which is a type of production line where tasks are performed in parallel on both sides of the line. This paper presents a mathematical model and a GA for the SALBP-2 problem. In the algorithm, the strategy adopted is localized evolution and steady-state reproduction to promote population diversity and search efficiency.

A Grouping GA (GGA) was used for solving grouping optimization problems, and the goal was to group members of a set into a small number of families in order to optimize objective functions under given constraints (see [22]).

The work in [8] aims at maximizing the workload smoothness and it is a kind of SALBP-3.This paper develops a GGA for ALBP of sewing lines with different labor skill levels in the garment industry. The GGA can allocate workload among machines as evenly as possible for different labor skill levels so that the mean absolute deviations can be minimized. In this article, the GGA's best parameter setting of population size, crossover rate and mutation rate are determined, respectively. Assigning more operators to workstations and training operators with multiple skills and high efficiency can result in low MAD, short cycle time, and high throughput.

Tabu search: Some scientists used other methods such as TS; however, it is not as frequently used as GAs. Developing TS for SALBP-1 is not as much as SALBP-2 due to some difficulties, but an elementary TS procedure is proposed in [23] for SALBP-2.

In [23], researchers investigate and depict the similarities and the performance of SA and TS. Different modern search methods used for combinatorial problems have a common structure. The realization of both methods is shown for the ALBP and their performance is compared. The problem set consists of well-known literature problems. In this comparison, the TS is successful and superior to other methods.

The application of TS is described in [24]; it is a metaheuristic technique for optimization problems applied to the ALBP. Four different versions of algorithms have been developed and the results show that, with the exception of a few cases, the TS always finds optimal solutions. This enhances the belief that the TS can be a very promising tool for solving problems in the manufacturing area where various problems can be modeled as discrete problems. Future research will focus on the application of TS to the multiple assembly line problem and the UALBP in the JIT environment.

A new TS algorithm was presented in [25], where its performance on the SALBP-1 was evaluated, and the algorithm was tested on a real industrial data set. Computational results on both standard benchmark instances and real life problems from industry have shown that the method is efficient when compared to the existing ones. The flexibility of metaheuristics makes it possible to adapt it to new specifications. In this richer and more complex problem, the TS outperforms the most popular priority-based heuristics and makes it possible to confirm the efficiency and adaptability of this type of solution in complex problems such as the one addressed in the paper.

Simulated annealing: In [26], the problem of balancing assembly lines with stochastic task processing times is addressed. The size of the problems that can be solved by optimal methods is limited and hence different heuristics have been developed, which give sub-optimal solutions. This work presents an approach for solving the problem of using the SA technique. The proposed approach tries to reach the global optimum by not getting trapped at the different local optimum points. Another feature of this method is the non-dependence of the final solution on the initial solution. Solutions for ALBP obtained using the above mentioned method are compared to the results of other greedy heuristics. The results of the experiment indicated that the lower the rate of reduction of the control parameter, the better will be the quality of the solution. If the cooling rate is too rapid or if the number of configurations changes (iterations) is not sufficient, then the best results will not be obtained. Also it was noted that increasing the number of iterations beyond a certain level does not have as much impact on the final solution as the cooling rate.

A hybrid SA to solve a multi-objective optimization problem with stochastic times approach is addressed in [27]. The objectives are (i) minimization of the smoothness index and (ii) minimization of the design cost. To deal with multi-objectives and enable the decision maker to evaluate a greater number of alternative solutions, a multinomial probability mass function was implemented in the proposed SA. The result reveals this algorithm can be used for multi-objective single-model/mixed-model, U-shaped deterministic/stochastic ALBP.

Ant colony: The work in [28] addresses the SALBP-2. The problem used ACO metaheuristics with different features and obtained good results. Afterwards, an adaptation of the previous implementation is used to solve a real case problem found in a bike assembly line with a hierarchical multi-objective function and additional constraints between tasks.

An ACO optimization for the single model UALBP is used in [29]. The authors conduct an extensive experimental study in which the performance of the proposed algorithm is compared against best known algorithms reported in the literature. The results indicate that the proposed algorithms display very competitive performance against them.

Other methods: A new heuristic algorithm is presented in [30] to solve SALBP-1. The presented algorithm makes an order of firing sequence of transitions from Petri net model of precedence diagram. Task is assigned to a workstation using this order and backward procedure. Computational study shows the algorithm efficiency.

In [31] a multi-objective evolutionary algorithm with strong convergence for solving ALBP by considering worker capability is addressed. The problem goals are to minimizing of the cycle time and total worker cost. The comparison at the end shows the efficiency of the new algorithm.

3.1.2 Mixed-Model

Several manufacturers are switching their production lines from single product or batch production to mixed-model production. In the mixed-model production, different models of a product are produced on the same line. This helps manufacturers provide their customers with a variety of products in a timely and cost-effective manner [32].

The models may differ from each other in terms of size, color, used material, or equipment, because their production requires different tasks with distinct task times and precedence relations. As a consequence, finding a line balance where station loads have the same station time and equipment requirement whichever model is produced is almost impossible.

3.1.2.1 Reduction Rule and Horizontal Balance

The MALBP relies on the same basic assumptions as the SALBP. Two basic approaches can be distinguished for modelling and solving the MALBP: (i) reduction to single-model problems, (ii) horizontal balancing in the multi-model context [33].

There is an assumption for using technique one, if task *"A"* in a model comes after task *"B"*, this precedence relation should be obeyed in other models. This means it is not possible in a model task *"B"* to be allocated before task *"A"*. In this case, this procedure is useless.

Reduction rule: By relaxing the assumption that identical tasks have to be assigned to the same station for all models, the mixed-model problem is decomposed into P independent SALBP instances. A respective generalization of the shortest-path formulation of SALBP-1 is presented by Roberts and Villa (1970), as well as by Rao (1971). However, the assignment of identical tasks to different stations is usually not desired with regard to additional facility requirements, loss of specialization effects, complicated production control, and setup inefficiencies. Only in the case of multi-model production, where batches of models are processed, can this relaxation of the MALBP be useful [33].

A MALBP-1 with a model dependent cycle time is modelled by [34] and solved through a shortest path procedure. The problem is NP-hard since with a single model and tasks with no precedence relations it is easy to reduce the problem to a bin packing problem which is NP-hard in the strong sense. Hence, the combinatorial nature of the mixed-model line balancing problem makes it difficult to obtain optimal solutions, though the mixed-model line is the most frequently encountered type in industry due to the pressure of producing several models to attain higher customer satisfaction. This paper presents a binary integer programming model for the MALBP in which some tasks are common to different models. The attempt is to decrease the size of the model by utilizing a combined precedence diagram and some variables that limit the increase in the number of decision variables and constraints. The resulting model is significantly superior to the one reported in the literature with regard to the number of decision variables and constraints. The experimentation revealed that the model is capable of solving problems with up to 40 tasks in the combined precedence diagram. However, due to the NP-hardness of the problem, the model would be too large to obtain optimal solutions with larger problems. The model serves as a starting point for researchers in the field, and may be used as a validation tool for heuristic procedures.

A binary integer formulation is introduced in [35], where the size of the model is reduced by utilizing a combined precedence diagram. This model used the concept of earliest and latest station tasks.

Horizontal balancing: A mixed-model line in a make-to-order environment is addressed in [36]. Here, a heuristic is designed for minimizing the number of workstations for a predetermined cycle time consisting of three stages: the balancing of a combined precedence diagram, balancing each model type separately subjected to the constraints from the first stage, and a neighbourhood search based on the improvement procedure. The horizontal balancing objective is used in the second stage.

3.1.2.2 Other Methods

In [37] a heuristic and a B&B method are used. In the heuristic method, the first step is to divide the task into several sub-tasks so that the relative task time does not exceed 1. The next step is to insert the subtasks into workstations. The B&B method consists of two components. The first is the lower bound calculation. For the branching procedure the lower bounds will be used as a threshold to continue a branch. Since each branch presents an alternative to the current solution, the strength of the bounds will also determine the ability of the proposed algorithm to find an optimum solution procedure. However, if the global lower bound value is less than the heuristic method's result, then a back-tracking and branching procedure is initiated to obtain the optimal solution.

Generally speaking, this study proposes a new algorithm which combines a heuristic search with a B&B technique to optimize the solution. Moreover, this study presents a case study on the footwear industry where the publication of research on line balancing field was nonexistent [38].

Some of the heuristic procedures which are used will be explained:

A mathematical programming model and an iterative GA-based procedure for the MALBP-2 with parallel workstations is presented in [39]. The goal is to maximize the production rate of the line for a predetermined number of operators. The problem addressed accounts for some relevant issues that reflect the operating conditions of real-world assembly lines, such as zoning constraints and workload balancing, and also allows the decision maker to control the generation of parallel workstations.

A classical GA was proposed in [40] and the aim is to minimize the number of workstations. A Hybrid (HYB) GA approach was incorporated that used the solution from the modified ranked positional method for the initial solution to reduce the search space within the global space, thereby reducing search time. Several examples illustrate the approach. Results considering the minimization of workstations show that the GA approach has produced better outcomes than the modified rank position weight. Furthermore, the HYB GA results have improved performance more significantly than a classical GA.

In [41] a modified procedure is proposed to solve MALBP-1 with using GA frame work. The standard encoding scheme directly assigns the tasks to the workstations in a balancing solution. The proposed model minimizes the total number of workstations and allows the user to control the replication process. The major contribution of this study is that it clearly defines the reassignment process to handle numerous conditions that can evolve during task allocation. It also controls the process of replicating workstations.

In the solving process each chromosome is a string of length N (number of tasks), where each element represents a task and the value of each element represents the workstation to which the corresponding task is assigned. The main genetic operator is the crossover, which combines pieces of information from different individuals in the population [41].

An SA approach for a two-sided line is proposed in [42]. Two-sided line occurs in plants producing large high-volume products such as auto mobiles, buses or trucks. An SA metaheuristic approach is proposed in [42] to effectively solve cost-oriented two-sided ALBP. The proposed SA approach builds balancing solutions for the problem based on a priority list of tasks. The task numbers are used in the proposed algorithm to represent the priority value of tasks for constructing a priority list. In the initial solution, each position has its own task number. A neighbour solution is a new solution which is obtained from a current solution by a move. The proposed algorithm minimizes the total cost per product unit. By conducting computational experiments, the performance of the proposed SA algorithm has been compared to the Mixed Integer Programming (MIP) model for the small problems and a heuristic procedure for the medium and large problems. The experimental results show that the proposed SA algorithm generates optimal solutions in small problems and outperforms the other approach in terms of solution quality.

The work in [43] presents an approach based on ACO techniques to effectively address the assembly line balancing problem with the complicating factors of parallel workstations, stochastic task durations, and mixed-models. This methodology is used to address several assembly line balancing problems from the literature. The assembly line layouts obtained from these solutions are used for simulated production runs so that output performance measurements (such as cycle time performance) are obtained. Output performance measurements resulting from this approach are compared to output performance measurements obtained from several other heuristics, such as SA. A comparison shows that the ant approach competes with the other heuristic methods in terms of these performance measurements.

An ACO optimization algorithm for balancing mixed-model assembly lines is explained in [44]. The planned algorithm accounts for zoning constraints and parallel workstations and aims at minimizing the number of operators in the assembly line for a given cycle time. In addition to this goal, the procedure used looks for solutions that smooth the workload in workstations, which is an important aspect to account for in balancing mixed-model assembly lines. Computational experiments show that the proposed algorithm performs better.

A simulation analysis of sequencing rules in a flexible flow line in a footwear industry was proposed in [45]. This paper was a great help to perform a comparison of different sequencing rules, related to which box, with shoes components should be selected amongst those available in the Work In Process (WIP) store of the transporter. This article also considers the buffers between the workstations.

3.2 Scheduling and Lot-Sizing

Scheduling uses mathematical programming techniques or heuristic methods to allocate limited resources to the processing of tasks. Based on [46], there are various important machine configurations for this problem. Table 3.3 describes the most basic configurations.

The case of this study is may be considered as job shop. In the Job Shop Scheduling Problem (JSSP), a set of jobs and a set of machines are given. Each machine can handle a maximum of one job at a time. Each job consists of a chain of operations, each of which needs to be processed during an uninterrupted time period on a given machine. The purpose is to find a schedule, that is, an allocation of the operations to time intervals on the machines with the minimum duration required to complete all jobs [47], [48]. For this problem there are some characteristic and constraints, the most important being provided in table 3.4 [46].

In [49], a good classification for this problem is denoted with a tuple-notation,

Machine configurations
Single-machine models
Parallel-machine models
Flow shop models
Job shop models

Table 3.3: Machine configurations.

Characteristics and constraints
Precedence constraints
Routing constraints
Material-handling constraints
Sequence-dependent setup times and costs
Preemption
Storage-space and waiting-time constrains
Make-to-stock and make-to-order
Machine-eligibility constraints
Tooling constraints and resource constraints
Personnel scheduling constraints

Table 3.4: Characteristics and constraints.

which considers features such as station boundaries, reaction on imminent work overload, processing time, concurrent work, setups, parallel stations, number of stations, homogeneity of stations, launching discipline, return velocity, line layout and objectives.

As cited prior, the situation under study is JSSP with a make-to-order environment. There are differences between stitching and assembly lines. The most crucial one is related to the lots, in the stitching line the problem interfaces with S&LS together but not at the assembly line.

The next item is parallelization which in the stitching line for specific products could be considered. Furthermore, there are boundaries on the skill of the operators on both of the lines.

There are two main categories for solving this problem, exact and approximate methods and each has different sub-classes.

A mathematical formulation of the problem and decomposition approach in footwear industry was used in [50]. This paper deals with the scheduling of operations on a multiprocessor machine in the context of footwear manufacturing. Multiprocessor machines are composed of several parallel processors. Unlike parallel machines, the entire machine needs to be stopped whenever a single processor needs a setup. There are different models in the line and for each of these models, all the sizes proposed by this manufacturer must be produced in various quantities. The objective is to schedule the production of the required sizes on the machine's different processors in order to minimize the global makespan, which includes both the production time and the set up time. A mathematical formulation of the problem first presented. Then, a decomposition procedure based on the mathematical model is used. Since solving the model presented, could be prohibitively time consuming, an approximate solution can be reached by using a decomposition procedure. The results show that the decomposition procedure is able to produce very good results for small problems. However, as problems become larger, the decomposition procedure becomes less interesting, and the one of the proposed heuristic which name is look-ahead becomes a better choice for producing high quality solutions in a fraction of a second.

An assembly sequencing problem for mixed-model assembly lines in a JIT production environment is studied in [51]. The model for this problem was not linear which is difficult to solve especially when the number of products considered, are large, and the total production quantity, are also large. A B&B algorithm (an enumeration scheme) is proposed as a solution methodology. A simple heuristic is used for finding an upper bound of the optimal objective function value of problem. Finally, the performance of the algorithm has also been evaluated by using randomly generated test problems. The computational results obtained have indicated that the algorithm developed is indeed an effective means for generating good assembly sequences for mixed-model lines in a JIT production environment.

A dispatching rule belongs to the approximate approaches and this rule prioritizes all the jobs that are waiting for processing on a machine. The prioritization may take into account the jobs attributes and the machine attributes, as well as the current time. Whenever a machine has been freed, a dispatching rule inspects the waiting jobs and selects the job with the highest priority. There are many basic dispatching rules (see table 3.5) [46].

Base on [52] during years 2000 to 2009 there are 116 articles which are related to the dispatching rule. In [53] sixteen dispatching rules are selected from the literature and used as the benchmarks. Their features and design concepts are also discussed. Then a dispatching rule is proposed with the goal as achieving good and balanced performance when more than one objective is concerned at the same time. This article shows that the rules usually focus on one objective and cannot provide good performance on multiple objectives at the same time.

A new dispatching rule for scheduling in a job shop is proposed in [54]. These rules are based on the additive combination of the process time, total work-content of jobs in the queue of next operation of a job, arrival time and slack of a job. These new rules are combination of the other rules.

Furthermore, by applying a simulation method, different dispatching rules in a mixed-model line were tested in [45].

One of the most successful heuristic procedures developed for the minimization of

	Rule	Objectives
Rules dependent on release dates and due dates	Earliest Release Date first (ERD)	Variance in throughput times
	Earliest Due Date first (EDD)	Maximum lateness
	Minimum Slack first (MS)	Maximum lateness
	Longest Processing Time first (LPT)	Load balancing over parallel machine
Rules dependent on processing times	Weighted Shortest Processing Time first (WSPT)	Weighted sum of completion times, WIP
	Critical Path (CP)	Makespan
	Largest Number of Successors (LNS)	Makespan
Miscellaneous	Service In Random Order (SIRO)	Ease of implementation
	Shortest Setup Time first (SST)	Makespan and throughput
	Least Flexible Job first (LFJ)	Makespan and throughput
	Shortest Queue at the Next Operation (SQNQ)	Machine idleness

Table 3.5: Basic dispatching rules.

the makespan is the shifting bottleneck heuristic [46]. The structure of the shifting bottleneck heuristic shows the relationship between the bottleneck concept and more combinatorial concept such as critical (longest) path and maximum lateness.

A modified shifting bottleneck heuristic is developed for minimizing the total weighted tardiness in a semiconductor wafer fabrication facility [55].

All the explained examples for finding an approximate solution start without a schedule and gradually construct a schedule by adding one job at a time. But, the local search algorithms are from improvement type. They start with a complete schedule, which may be selected arbitrary, and then try to obtain a better schedule by a local search, what does not guarantee an optimal solution. It usually works as an attempt to find a better schedule than the current one. GA, TS and SA are used more than the other Metaheuristics in recent years [53].

How to adapt GAs to the JSSP is very challenging [56].

GA for a multi-objective sequencing problem were used in [57], where variation of production rates and the number of setups are to be optimized simultaneously. The developed multi-objective GA uses three basic genetic operators as crossover, inversion and mutation. It also exploits a non-dominated sorting idea along with a niche mechanism to obtain quality as well as diverse locally pareto-optimal solutions. An elitist strategy was also employed to preserve locally non-dominated frontiers found over generations from getting lost. A novel scheme for selection was used that selects individuals from the population as well as the elite set. It was also observed that using the high rates for crossover and mutation would increase diversity whereas the low rates would improve quality.

A GA for scheduling staff of mixed skills is proposed in [58]. The problem as a multi-criteria optimization model, where the primary objective is to minimize the total cost for assigning staff to meet the manpower demands over time, the secondary objective is to seek a solution with the maximum surplus of staff among the solutions with almost same level of assigning cost, and the tertiary objective is to reduce the variation of staff surplus over different scheduling periods. The proposed GA differs from traditional GAs in the following components: (i) it performs its parent selection by using a ranking scheme that considers successively the three criteria; (iii) it uses a multi-point crossover operator based on the hamming distance between schedules; and (iii) it adopts a heuristic to resolve the problem of infeasibility created by crossover operations.

The generic TS concept and its ability to move smoothly on a range of increasingly sophisticated implementations make it an interesting tool for real world applications [59].

Base on [60] exact algorithms cannot solve instances larger than 250 operations within a reasonable time (hours, days). A new approximate algorithm that is based on the big valley phenomenon used, and then a tabu list is prepared and finally, results in a quick time on a personal computer are revealed.

In [61] a TS is used to minimize the makespan. Alternative operation sequences and sequence dependent setups are two important factors that are considered simultaneously in this paper. The proposed TS algorithm is composed of two parts: a procedure that searches for the best sequence of job operations, and a procedure that finds the best choice of machine alternatives. Randomly generated test problems are used to evaluate the performance of the proposed algorithm with the optimal solution. A mathematical model is solved by a B&B method and the authors show that the proposed algorithm generates good quality solutions.

Neighborhood structures and move evaluation strategies play the central role in the effectiveness and efficiency of the TS for the JSSP. In [62], a new enhanced neighbourhood structure is proposed and applied to solving the JSSP by TS approach. Using this new neighbourhood structure combined with the appropriate move evaluation strategy and parameters, then the TS approach are tested on a set of standard benchmark instances and found a large number of better upper bounds among the unsolved instances. The computational results show that for the rectangular problem this approach dominates all others in terms of both solution quality and performance.

The quality of solutions found by TS approach depends on the initial solution. To overcome this problem and provide a robust and efficient methodology for the JSSP, the heuristics search approach combining SA and TS strategy is developed in [63]. The main principle of this approach is that SA is used to find the elite solutions inside big valley so that TS can re-intensify search from the promising solutions. This HYB algorithm is tested on the standard benchmark sets and compared with the other approaches. The computational results show this proposed algorithm could obtain the high-quality solutions within reasonable computing times.

An effective HYB TS algorithm to solve the flexible JJSP is proposed in [64]. Three minimization objectives – the maximum completion time (makespan), the total workload of machines and the workload of the critical machine are considered simultaneously. In this study, a TS algorithm with an effective neighbourhood structure combining two adaptive rules is developed, which constructs improved local search in the machine assignment module. Then, a well-designed left-shift decoding function is defined to transform a solution to an active schedule. In addition, a variable neighbourhood search algorithm integrating three insert and swap neighbourhood structures based on public critical block theory is presented to perform local search in the operation scheduling component. The proposed HYB TS algorithm is tested on sets of the well-known benchmark instances. The statistical analysis of performance comparisons shows that the proposed HYB TS algorithm is superior to four existing algorithms.

A SA based heuristic approach is planned in [65] which simultaneously consider both setups and the stability of parts usage rates when sequencing jobs for production in a JIT environment. Varying the emphasis of these two conflicting objectives is explored. Several test problems are solved via the SA heuristic, and their objective function values are compared to solutions obtained via a TS approach from the literature. Comparison shows that the SA approach provides superior results to the TS approach. It is also found that the SA approach provides near-optimal solutions for smaller problems.

In [66] the JSSP is studied with the objectives of minimizing the makespan and the mean flow time of jobs. The simultaneous consideration of these objectives is the multi-objective optimization problem. A metaheuristic procedure based on the SA algorithm called Pareto Archived SA (PASA) is proposed to discover non-dominated solution sets for the JSSP. The seed solution is generated randomly. A new perturbation mechanism called segment random insertion scheme is used to generate a set of neighbourhood solutions to the current solution. The PASA searches for the non-dominated set of solutions based on the pareto dominance or through the implementation of a simple probability function. The performance of the proposed algorithm is evaluated by solving benchmark JSSP instances provided by the OR-library. The results obtained are evaluated in terms of the number of non-dominated schedules generated by the algorithm and the proximity of the obtained non-dominated front to the pareto front.

The challenge in the stitching line is the sizing and scheduling of the lots. Researchers tended to be divided between those investigating lot-sizing and those interested in scheduling problems. However, in many industrial applications the close relationship between LS&S forces both decisions to be made simultaneously.

Usually the batching is done to avoid the set up costs [67]. Another reason for batching occurs when a machine can process several jobs simultaneously also it may be cheaper or faster to process jobs in a batch than to process them individually.

A new model was presented in [68] that permits integrated LS&S of multiple items on a single, capacitated production line when inventory holding and sequence-dependent setup costs are given a heuristic solution procedures with a threshold for the accepting is used.

A dual re-optimization algorithm is combined with a local search heuristic for solving a MIP problem [69]. This idea is applied to a LS&S of several products on a single, capacitated production line by embedding a dual network flow algorithm into threshold accepting and SA.

The author in [70] solves S&LS problems simultaneously while considering non-identical parallel production lines (heterogeneous machines). The problem is heuristically solved by combining the local search meta-strategies threshold accepting and SA, respectively, with dual re-optimization. Such a solution approach has already proved to be successful for the single machine case. The solution quality and computational performance of the new heuristics are tested by means of real-world problems gathered from industry.

A fuzzy rule-based system is developed in [71] which its purpose is determining the size of lots by using the following premise variables: size of the job, the static slack of the job, workload on the shop floor, and the priority of the job. Both premise and conclusion variables are modeled as linguistic variables represented by using fuzzy sets. The objectives that are used to measure the quality of the generated schedules are average weighted tardiness of jobs, the number of tardy jobs, the total setup time, the total idle time of machines and the total flow time of jobs.

The size of a batch appears to be a sensitive issue. Large batches improve the movement in the line and save time. On the other hand, a batch may group jobs of different priorities, and consequently, jobs of high priorities can be processed in different batches. Therefore, a customer service may be improved by having smaller batches.

In [72] the focus is on models and algorithms, thus the literature in sequencing and detailed scheduling is ignored. This paper states many practically important problems are still far from being solved in the sense that they could routinely be solved close to optimality in industrial practice.

Balancing and scheduling are two most important short-term planning issues. The balancing objective is to determine an allocation of assembly tasks for a mix of products among the assembly stations with limited work space so as balance the station workloads. In contrast, the scheduling objective is to determine the detailed sequencing and timing of all assembly tasks for each individual product, so as to maximize the line's productivity, which may be defined in terms of throughput, makespan, average flow time for a given product, etc [46].

A lot of researches have considered balancing and scheduling separately or sequentially. However, in an integrated approach the balancing and scheduling decisions are made simultaneously or try to optimize both at the same time which is a challenging job.

An innovative balancing–sequencing procedure that aims to optimize the assembly line performance is proposed at [73]. At the same time the buffer dimensions was contained in function of different market demand and production mix. The model is validated using a simulation software and an industrial application is presented. The proposed procedure, with an integrated, step-by-step procedure based on B&B methodologies, goals are to minimize both idle time and overload time across workstations both in the balancing and in the sequencing, improving the performance of the assembly line and reducing the demand of buffer capacity. Literature, on the other hand, is more focused on the sequencing problem regarding the JIT environment and does not consider the presence of buffer in case of minimization of the deviation of workloads across workstations.

The authors in [74] try to minimize total utility work. One of the most important contributions of this paper is the development of a Mixed-Integer Linear Programming (MILP) model to obtain the exact solution, but because of the NP-hardness of the problem, an evolutionary algorithm has been employed. Finally, a comparison study among the developed MILP, the proposed algorithm and the modified version of the co-evolutionary GA has been conducted. The effectiveness of the proposed EA has been statistically demonstrated from two perspectives: the quality of solutions and the time taken to achieve the best solution. A SA approach is used in [75] for solving the proposed problem such as [74] but station-dependent assembly times is inserted.

Balancing and scheduling of flexible mixed model assembly lines with parallel stations is formulated in [76]. A MIP model is proposed and then a decomposition scheme for large scale applications is developed. Finally, a comparison between the performance of the proposed model and decomposition scheme using various size test instances is done.

3.3 Optimization Methods

There are different methods for solving balancing, S&LS problems have been described in the previous sections (3.1 and 3.2) and publications referred but in general they are aggregated in the following categories: Optimum seeking algorithms or Exact methods, and Approximate procedures. Figure 3.3 shows a classification of solution methods used were mentioned [77].

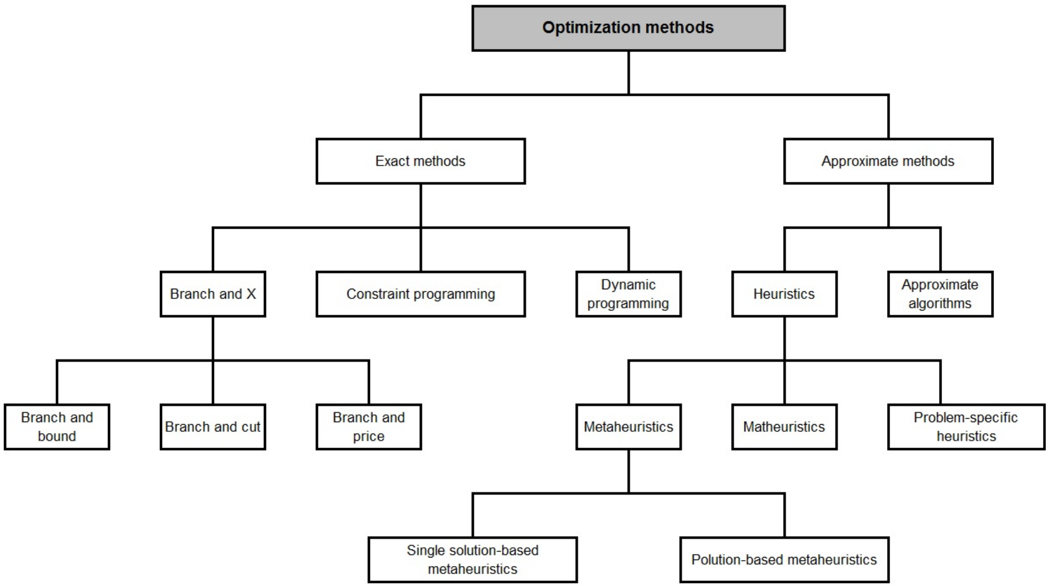


Figure 3.3: Optimization methods.

Many common optimization problems are classified as NP-hard. Although the efforts made to solve the optimization problems efficiently have produced important progress in the last years, there is no universal method. Consequently, there is much interest in approximation algorithms that can find near-optimal solutions within a reasonable computation time [9].

However, the heuristics based on local search algorithms do not effectively explore the complete solution space and it is possible to be trapped in the local optima. This limitation could be overcome by using a metaheuristic or repairing strategy [7].

Constructive heuristics gradually develop a solution, but an improvement starts with providing a complete answer and then trying to obtain a better result. Generally, local search algorithms or improvement procedures try to improve a given feasible solution by iteratively transforming it into another feasible solution.

Different classes of optimization models (figure 3.4) are used in practice to formulate and solve decision making problems.

As illustrated in the previous sections, there are many different optimization methods to deal with line balancing and scheduling problems, and they were classified according to figure 3.3. In the following we make a short presentation of the most used ones:

Branch and Bound (B&B): This method is a general way for finding optimal

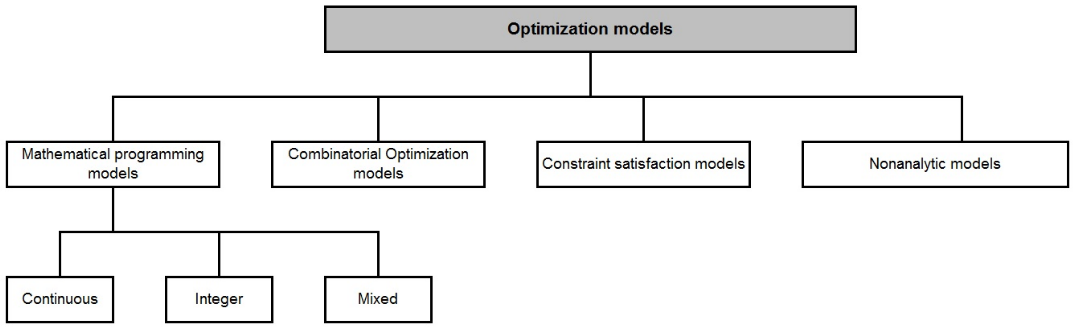


Figure 3.4: Optimization models (The different classes are possibly overlapping).

solutions. B&B is essentially a strategy of *"divide and conquer."* The idea is to partition the feasible region into more manageable subdivisions and then, if required, to further partition the subdivisions. In general, there are a number of ways to divide the feasible region, and as a consequence there are a number of B&B algorithms [78]. The disadvantage of this method is that it can be extremely time-consuming, since the number of nodes is often very large [46].

Genetic Algorithm (GA): GAs, which have been introduced by John Holland (1975) in the 1970s have become increasingly popular among approximation algorithms for finding near optimal solutions to large optimization problems.

GA is a stochastic search method inspired by concepts from Darwinian evolution theory and belongs to a class of metaheuristic methods known as Evolutionary Algorithm (EA). As a solution approach, GA has two advantages: (i) GA searches a population rather than a single point and this increases the likelihood that the algorithm will not be trapped in a local optimum since many solutions are considered concurrently, and (ii) GA fitness function may take any form and several fitness functions can be utilized simultaneously [9].

Tabu search (TS): TS algorithms are among the most effective approaches for solving the JSP which is one of the most difficult NP-complete problems. TS is a metaheuristic deterministic search technique and designed for the purpose of escaping local optima. It employs local search methods for mathematical optimization [79] and moves from one schedule to another with the next schedule being possibly worse than the one before [46]. Generally, the main idea behind TS is very simple. A *"memory"* forces the search to explore new areas of the search space. We can memorize some solutions that have been examined recently and these become tabu (forbidden) points to be avoided in making decisions about selecting the next solution [79]. In the scheduling area one of the metaheurestics which are used more is TS therefore, in this work TS is used for the S&LS problems.

Simulated Annealing (SA): SA also known as Monte Carlo annealing, statistical cooling, probabilistic hill-climbing, stochastic relaxation, and probabilistic exchange algorithm. This method designed for the purpose of escaping local optima. At each step, the SA heuristic pick a point from a neighbourhood and if it is better then select it otherwise select it with some probability. These probabilities ultimately lead the system to move to states of lower energy. Typically this step is repeated until the system reaches a state that is good enough for the application, or until a given computation budget has been exhausted [79].

Ant Colony (ACO): Another methodology is inspired by the behaviour of social insects which name is ACO, Ants are known for being able to find the shortest path between their nest and a food source, without making use of visual cues but only by following pheromone trails released by other ants. The more intense the trail, the higher the probability an ant will follow it and thus reinforce the trail with its own pheromone. So, it is the colony as a whole that coordinates the activities without a direct communication between individual ants, as an isolated ant basically moves at random. During this procedure the ant takes into account the information left by other ants (pheromone trails) and, eventually, other available information about the problem. At the end, good solutions emerge resulting from the indirect communication between the ants [44].

Chapter 4

The Case Study

This project is based on a real case of the footwear industry. Depth research on the case study is conducted and the developments and findings are generalized to other problems in the footwear industry or other similar process industries.

This work considers balancing and lot-sizing of stitching line.

4.1 The Stitching Line

As mentioned manufacturing shoes involve various processes which take place in different sections of a factory. One of the challenging matters is the management of stitching line. Besides the need to balance the line, a lot-sizing issue problem, involving the size of the boxes moving in the line, must be tackled. The company creates boxes around size 10 (components of pairs of shoes) but the "right" size is not really known. This work will focus, in particular, in the study and test of different lot-sizes.

Generally the following topics are involved in the stitching line:

Variety of models: Production in this industry covers various groups such as men, women and children and each has different categories, and every category has different models. For example, in the beginning of the summer ladies prefer sandals with high heels, but in the middle of summer they prefer slippers or sandals without heels. The attitudes of people have changed, and companies compete with each other in this area, reducing number of orders of various different models. This means that there are several models at the same time on the production line. Therefore, regarding to this issue and balancing this problem is MALBP.

Bottlenecks: As mentioned, the footwear industry involves process cutting, stitching and assembly. The speed of cutting and assembly processes is higher than stitching, and that is why this operation is considered a bottleneck in the footwear industry. As a result, to solve that bottleneck problem some companies send the pieces to other firms for stitching. The rest uses more labour, machines or apply further strategies to make the cycle time similar to the rest of the lines.

To achieve the work plan, the stitching lines need to attain a daily target; this goal turns to an hourly goal and finally the cycletime will be determined. The pieces are moved by a transporter with limitations in speed. Furthermore, there are stop points on the line close to each workstation. The maximum number of stop points in this line is 46. Therefore, all these restrictions should be met. For instance, in the balancing section with a special takt time, if the number of workstations required is 47 that is not possible since the number of workstations should not exceed 46.

Different tasks: To produce a shoe on a line, it is essential to divide the total work into a set of tasks, each task requiring a certain time; each model has a specific production routing what increases the difficulty of the planning. For example, it is not possible to sew some parts before they have been glued and assembled. Each task will have, in general, a set of predecessor tasks. There are direct and indirect predecessors. For instance, figure 4.1 shows a graph with 9 tasks. Task 40 expresses that its processing requires tasks 10 and 50 (direct predecessor) and 60 (indirect predecessors) to be completed. Task 40 must be completed before its (direct and indirect) successors 20, 70, 80 and 90. As previously mentioned, each task also has different processing times. For instance, the processing time of task 70 in model A is 2 minutes and in the model B the processing time is 2 minutes and 30 seconds. This problem is a JSP and in the literature this kind of problem considers a balancing process called mixed-model line balancing.

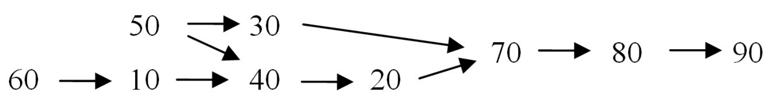


Figure 4.1: Precedence graph.

Clients: The clients are generally of different types. Some of them are big clients, while others are retailers, and the third category is related to people who order shoes from the internet through the virtual store. The company gets the order from the big clients and retailers usually one season before, which means that the pieces on the line in the summer are related to the autumn and winter. However, if one person orders a shoe from the virtual store it should be ready within two or three days and that order should be on the line with the other client orders.

The first and second categories of customers usually determine the week of delivery dates, but the online orders should usually be produced very quickly.

Boxes in the stitching line: In the stitching line pieces are put inside boxes and the boxes move between workstations. Each box contains parts with the same model, equal size and identical color, although they can be from different client orders. One of the aims of this project is to put pieces in a box in a way one client order is produced more or less at the same time and packed as soon as possible. Each box can move in any direction and there is no limit for that. One of the critical issues of this project is finding the ideal number of items inside each box because by changing this amount the processing time of the work pieces also changes, and this item can influences the quality of the solution in the scheduling sector. Could makes the solution better or worse. However, the company prefers to create boxes with size 10. As an instance if a production order of a model is 35 pairs they create 4 boxes with sizes, 10, 10, 10 and 5 but some exceptions are considered. So, lot-sizing is another crucial issue for this work.

Movement of pieces in the stitching line: The stitching line consists of workstations arranged along a conveyor. The workpieces are put inside a box and the box move in any direction. In the assembly line, each shoe is put on a place in the transporter and the system moves at a constant speed. Then the operator takes the unit, performs determined tasks and then puts it again on the transporter. But stitching line tackled with lot-sizing issue.

Buffers in the stitching line: The stitching line in the case study is unpaced because of the buffers, which are located between stations. In each station the operator takes the workpieces from the buffer upstream, performs all the defined operations and then sends the unit to the next buffer. Balancing and scheduling such a line is important because the wrong answer can cause blocking or starvation and since this project should respond quickly to some specific clients it is important to have an effective line.

Places for the boxes in a workstation in the stitching line are similar to what is presented in figure 4.2:

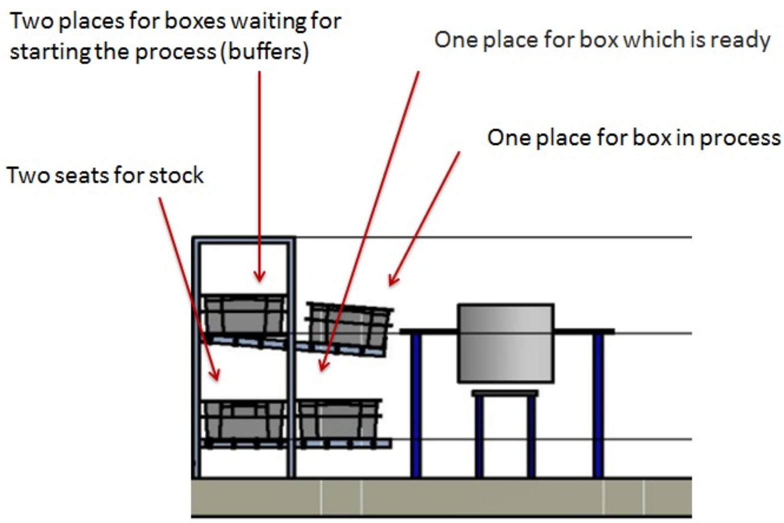


Figure 4.2: Workstation in the stitching line.

There are two seats for the stock, two places for boxes waiting to start the process or buffers, one place for the box in process and one place the for the box which is ready to be sent to the next station.

Operators and skill levels: In this line there are operators with different skill levels, some of them are experts and others are not. The next challenge of this work is to allocate the correct task to the right operator.

Special operators and tasks: There are some special operators in the company who are the only one with the ability to perform some tasks. Therefore, if there is only one operator available to perform a task, this operator will be considered special.

Machine restrictions: The next matter is the type of machine, which should be allocated to the workstations. The machines in this factory are also grouped based on their ability to perform different tasks. The other difficulty of this work is balancing and scheduling the line while considering the constraint on the type and number of machines.

Equipment selection: Generally, since workstations are composed of machines and operators, this work engages with an equipment selecting process. If a process needs distinct equipment (machines or manpower), it should be confined to that specific workstation and in this case the way of manufacturing a product depends on that equipment.

In the assembly line each workpiece moves separately, but as mentioned before this work concentrates on the stitching line and workpieces are put inside a box and the box moves between stations in any direction. Figure 4.3 shows the stitching line.

Figure 4.3: Stitching line.

4.2 Assembly Line Balancing Problem

Since the assembly line consists of different sections involving different operations being performed at different production rates, a balancing is necessary to make sure that the right person and machine are assigned the right task.

In this line the production plan is known and the first question is how many workstations are needed to satisfy this plan. Therefore, cycle time of the line can be calculated.(in the next chapter, section 5.1 more explanations are given)

Generally, since this line is a mixed-model line, the balancing problem is MALBP and regarding to knowing the cycletime and the objective which is minimizing the number of workstations, according to literature, this problem is from type one (MALBP-1). The following notation, indices and parameters, is used and it is useful to understand.

Indices:

j: index for models, $j \in \{1, 2, \ldots, J\}$
i: index for tasks, $i \in \{1, 2, \ldots, I\}$
k: index for set of boxes, $k \in \{1, 2, \ldots, K\}$

o: index for operators, $o \in \{1, 2, \ldots, O\}$
m: index for machines, $m \in \{1, 2, \ldots, M\}$

Input parameters:

P_{ij} = Processing time of task i for model j
$OP_{oi} = 1$: if operator o may perform task i; 0: otherwise
MT_m = Type of machine m
TRM_i = Type of required machine for task i
Q_{jk} = Quantity of model j in a box k
$MaxM$ = Maximum number of machines
CT_k = Cycle time relative to box k

The constraints which should be satisfied for the balancing problem are such as below:

1. Task assigned to exactly one required operator;
2. Task assigned to exactly one required machine;
3. Cycle time should be respected;
4. Each machine can only be allocated to only an operator;
5. Number of used machines should be less or equal than maximum number of machines;

The objective function for balancing problem is determining the number of needed workstations.

4.3 Scheduling and Lot-Sizing Problems

The next step for the company is minimizing makespan (maximum duration between starting and finishing tasks). Different issues are involved in this matter, one is scheduling and related topics and the other is lot-sizing.

As mentioned in section 4.1, the boxes are usually filled with 10 sets of workpieces but, as this size has not been analyzed, the company is very much interested in a in a well-grounded. This work, also intends to deal with this S&LS to find the "best" box size, so that makespan is reduced. According to the literature this case is JSSP.

The indices and input parameters used follow the notation:

Indices:

j: index for models, $j \in \{1, 2, \ldots, J\}$
i: index for tasks, $i \in \{1, 2, \ldots, I\}$
k: index for set of boxes, $k \in \{1, 2, \ldots, K\}$
o: index for operators, $o \in \{1, 2, \ldots, O\}$
m: index for machines, $m \in \{1, 2, \ldots, M\}$

Input parameters:

P_{ij} = Processing time of task i for model j
Se_{ij} = Sequence of tasks i in model j
$OT_{oik} = 1:$ if operator o perform task i in set k; 0: otherwise
$MP_{mik} = 1:$ if machine m perform task i in set k; 0: otherwise
Av = Available time (in a shift)

The constraints which could be satisfied for S&LS are such as below:

1. Task of a model in a set assigned to the determined operator;
2. Task of a model in a set assigned to the determined machine;
3. Operator and machine workload should not be more than available time;
4. Sequence of tasks should be respected.
5. Starting time of a task of a model in a set on a workstation could not be greater than finishing time of previous task of the same model in the equal set in the precedence diagram.

The objective function is minimizing makespan. The solution methods will presented in the next chapter.

Chapter 5

Solution Methods

This chapter takes account of the proposed solution methods for the case study and problems described in the previous chapter (sections 4.2 and 4.3). The first method deals with the mixed-model assembly line balancing problem and consists of a constructive heuristic which will be outlined in the next section; afterwards scheduling will take place, based on the balancing solutions; finally, a Tabu Search (TS) based method is applied to improve the existing lot-sizing problem solution.

5.1 Balancing Solution

Given a Production plan for a time horizon, the first question is how many workstations should be opened to satisfy it. Line balancing takes place here. Consequently, cycle time should be calculated. According to [44], if a line required to manufacture J model each with the amount D_j, in a planning horizon Av, then a cycletime could be calculated such as below:

$Cycletime = \frac{Av}{\sum_{j=1}^{J} D_j}$

Furthermore each task i is the smallest work element and could not be divided to a minor share, therefore at the same time cycletime of the line could not be less than this number. Although in this work the unity is not a pair of shoes but a box with dissimilar quantities, consequently cycletime for set of boxes is considered as:

$Cycletime_k = \frac{Av}{\sum_{j=1}^{J} D_j} \sum_{j=1}^{J} Q_{jk}, k \in [K]$

Precedence Constraint: All precedence diagram indicate the different order where tasks have to be performed. For the purpose of accomplishing this order, precedence constraints between tasks within workstations should be considered.

In this case study, boxes are moved from one station to the other in a forward and backward way and generally the line is not unidirectional one. Therefore according to [80], for the balancing section no precedence constraints have been taken into account.

Balancing Algorithm: In this section a constructive heuristic is used to create an initial solution quickly and if possible, with a reasonable quality. A well-known heuristic, the Ranked Positional Weighted method (RPW), is adapted to this case. This heuristic is introduced by Helgeson and Birnie in 1961 on single model line.

The RPW takes to account two elements. First, the summation of task times that follow a selected task in the precedence graph and second the position of the selected task in the precedence diagram. Then, the tasks are assigned to the workstations according to their RPW values.

Procedure:

1. Calculate the RPW for each task by summing the processing time of all tasks that follow it in the arrow chain of the precedence diagram.

2. List the elements in the order of their RPW, largest RPW at the top of the list.

3. Assign elements to stations according to RPW, avoiding precedence constraint and cycle time violations. [81]

An small example elucidates the procedure. Figure 5.1 shows the precedence diagram of a model.

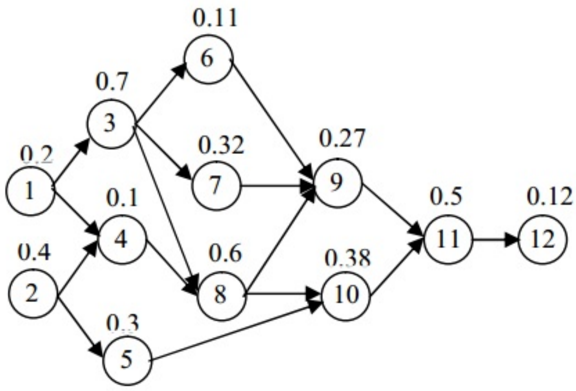


Figure 5.1: Precedence Diagram.

Solving procedure:

Step 1. Calculation: For task 1, the tasks that follow it in the precedence graph are $3, 4, 6, 7, 8, 9, 10, 11$, and 12. The RPW for task 1 would be the sum of the processing time for all tasks plus task processing time of task 1.

Step 2. Sorting: According to table 5.1 sorting will be done.

Step 3. Task Assignment: Tasks will be assigned to the workstations while satisfying the cycle time constraint (see table 5.2).

In this instance the number of required workstations are 5. However, as clear for applying this heuristic to the mentioned case adaptations are needed.

Task	RPW	Pij	Immediate Predecessors
1	3.30	0.2	-
2	3.00	0.7	1
3	2.67	0.4	-
4	1.97	0.1	1,2
5	1.87	0.6	3,4
6	1.30	0.3	2
7	1.21	0.32	3
8	1.00	0.11	3
9	1.00	0.38	5,8
10	0.89	0.27	6,7,8
11	0.62	0.5	9,10
12	0.12	0.12	11

Table 5.1: Tasks and RPW.

Workstation	Task	Pij	$\sum Pij$ at Workstation
1	1.00	0.2	0.9
	3.00	0.7	
2	2.00	0.4	0.91
	4.00	0.1	
	5.00	0.3	
	6.00	0.11	
3	8.00	0.6	0.92
	7.00	0.32	
4	10.00	0.38	0.65
	9.00	0.27	
5	11.00	0.5	0.62
	12.00	0.12	

Table 5.2: Tasks Assignment.

As mentioned, the case study covers a mixed-model line, not a single-model line, operators have different abilities, machines are of various types and lots are included. All these matters are considered to obtain the initial solution for the line balancing problem. Sequencing is not considered in the balancing solution.

5.2 Scheduling and Lot-Sizing Solutions

S&LS problems are important optimization problems, due to their relevance for the industry and also for possible high complexity. The goal of a scheduling problem is to allocate a sequences of tasks to given workstations while taking into account convenient objective(s). In this problem, the objective is minimizing the completion time (see section 4.3).

5.2.1 Scheduling

The scheduling algorithm starts allocating tasks to operators and machine while obeying the balancing solution. Obviously, the planning, must assure that the tasks do not overlap in time, meaning that the starting time of a given task on a machine must be after the ending time of the previous task. For every model, the sequence of tasks has to be respected. The makespan is the completion time of the last operation and thus, it is greater or equal to any of them.

One of the dispatching rules with the objective of minimizing makespan which is CP is used in this section ([46]). The scheduling algorithm calculates processing times of successors and sort them in descending manner and start allocating a task of a model that is in the top of the list and continues until the allocation of all the tasks.

Regarding lots, company prefers to create boxes with size 10 (as mentioned in chapter 4), therefore set of boxes are defined with this size for allocation. This issue will be explained by an small instance.

If there are 3 models in the line and the production order of them are such as table 5.3 then the boxes are created according to table 5.4 . The scheduling algorithm starts with first set of boxes and allocates all the model tasks and then goes to the next set and this process is repeated until all the tasks are allocated.

Model	Order Quantity
A	18
B	30
C	23

Table 5.3: Production Order.

Boxes	Sets		
	1	2	3
1	10	8	0
2	10	10	10
3	10	10	3

Table 5.4: Box Creation.

Finally, a detailed allocation of operators, machines and starting time and finishing time of tasks are determined. Moreover, makespan can be calculated at this phase.

Furthermore, and to accomplish a study of the impact of creating boxes with different sizes, the maximum initial amount inside each box will be changed from 7 to 13 and, consequently, makespan will vary. TS, which has been introduced in section 3.3, is the support for the method devised to cope with this matter.

5.2.2 Tabu Search

TS is a general search approach, which has been applied with success to large set of combinatorial optimization problems (see section 3.3).

A method based on TS is used to improve the initial solution generated by the balancing algorithm (constructive heuristic). The TS algorithm keeps current optimal solution and the best solution while exploring a neighbourhood.

As mentioned, the lot-sizing problem has, as objective, minimizing the makespan. The TS method will take into account different neighbourhoods. The search space is simply the space of all possible solutions that can be visited during the search [82]. Different neighbourhood searches lead towards different local optima. Therefore, the basic idea is to escape from the local optimum of a given neighbourhood, by changing the neighbourhood space to another neighbourhood. For this method to be effective, the various neighbourhoods have to be dissimilar enough to be able to escape from the local optimum of each other. Two used neighbourhood spaces are according the following:

Neighbourhood 1: Changing the order of the sets is one neighbourhood space. If the sequence of the sets are changed the quality of the solution will be changed, e.g. in the previous instance, if the order of set 1 and 2 are changed a neighbour solution will be created (see table 5.5). This changes could make the solution better or worse.

Boxes	Sets		
	2	1	3
1	8	10	0
2	10	10	10
3	10	10	3

Table 5.5: Neighbour Solution.

Neighbourhood 2: Changing the set by summing them is another neighbourhood space, table 5.6 shows another neighbour solution for the presented example.

Boxes	Sets		
	1	2	3
1	18	0	0
2	20	0	10
3	20	0	3

Table 5.6: Neighbour Solution.

Two TSs methods are applied to this problem. TS-One only uses neighbourhood 1, and TS-Two uses both neighbourhood spaces.

In the TS-One a short term memory is used and only 4 recent solutions are saved.

General Termination Conditions:

1. after a fixed number of iterations (or a fixed amount of CPU time);
2. after some number of consecutive iterations without an improvement in the objective function value (the criterion used in most implementations);
3. when the objective function reaches a pre-specified threshold value (according to [82]).

In this case the termination condition, is a fixed number of iteration while if improvement is not appear the algorithm will stop.

In TS-Two both neighbourhood spaces are used and when a move which transfer a solution to another one occurs, the neighbourhood spaces which causes the move become tabu for the next iteration. If both neighbourhood spaces in any iteration give equal solutions the TS algorithm makes tabu the neighbourhood spaces which was not tabu before.

The used TS was designed in order to escape from local optima convergence as most as possible and the termination condition is such as TS-One.

Chapter 6

Computational Results

This chapter includes some computational results relative to the test and evaluation of the solution methods developed, and described, for the balancing and lot-sizing problems. These are an adaptation of the RPW method for balancing, TS-One and TS-Two methods for lot-sizing. They were implemented in C++, compiled using Microsoft Visual Studio 2008 and run on an Intel Core (TM)i5 2.20 GHz processing unit with 8 GB of random access memory. The experiments are also carried out in real data from the case study.

6.1 Test Instances

The case study has provided data for five days of demand, The goal is to minimize an entire production cycle. In addition, in order to know the behaviour of the algorithm under different conditions, it has to be tested on different instances.

Five main parameters were chosen and they will vary during the experiments.

1. Number of available models in the line;
2. Number of tasks among models;
3. Number of available operators;
4. Number of available machines;
5. Different number of sets of boxes.

The diverse skills of operators generate more complexity, since different operators could be selected for a chosen task. Furthermore, generally the running time of the algorithm increases with the number of sets of boxes and models since it starts allocating tasks to the first set of boxes and continues by order of the sets. Therefore, size and complexity of the problem affect the computational time.

Detailed information related to the available models in Day-1 is shown in Table 6.1. On that day the production plan for each model is 100 pairs and available time is one day shift which is 7 hours and 10 minutes.

The number of available machines in Day-1 is based on a real situation while, in other instances, that number was created.

Gama	Sequência	Operação	Gama	Sequência	Operação	Gama	Sequência	Operação
500453	1	Riscar palmilha (colar estrela)	500542	1	Colar estrela	500564	1	Cravar estrela
500453	2	Cravar estrela	500542	2	unir forros	500564	2	unir forros
500453	3	Colar fivela	500542	3	Fazer Costura	500564	3	Fazer Costura
500453	4	cravar colher	500542	4	Queimar s/ forro	500564	4	coser orelhas
500453	5	colar forros	500542	5	Colar fivela	500564	5	Queimar s/ forro
500453	5	colar forros	500542	5	Colar fivela	500564	6	D.C. + colar forros
500453	6	Cravar tiras	500542	6	Rebater costuras	500564	6	D.C. + colar forros
500453	7	cravar fivela	500542	7	D.C. + colar forros	500564	7	Cravar almofadado ao cano
500453	8	Queimar c/forro	500542	8	Cravar almofadado	500564	8	Cravar almofadado
500453	9	Rentear forro	500542	9	cravar vivo	500564	9	Queimar c/forro
500453	10	Colar palmilha	500542	10	Queimar c/forro	500564	10	Rentear forro
500453	10	Colar palmilha	500542	10	Queimar c/forro	500564	11	Colar palmilha
500453	10	Colar palmilha	500542	10	Queimar c/forro	500564	11	Colar palmilha
500453	11	cravar corte a palmilha	500542	11	Rentear forro	500564	12	cravar corte a palmilha
500453	12	colar fita de reforço	500542	12	Colar palmilha	500564	13	Rebater costuras
500453	13	cravar vivo				500564	14	Colar ilhós
500453	14	queimar linha				500564	15	cravar vivo
500453	15	Vazar palmilha				500564	16	queimar linha

Table 6.1: Routing of the available models in Day-1.

6.2 Balancing Results

The Balancing Algorithm is compared with the real situation of the line for the real-based instances. The algorithm is very fast and in couple of seconds gives the balancing results. For comparison, boxes with the maximum size 10 were imposed. The good news were that, as a result of the algorithm, less number of operators were needed, in comparison with the real situation. Table 6.2 shows the given information related to the instances.

Instance	N. Models	N. Tasks	N. Operators	N. Machines	N. Sets
Day-1	3	31	22	49	10
Day-2	41	31	12	15	52
Day-3	16	15	13	15	42
Day-4	11	10	11	15	56
Day-5	15	15	13	15	25

Table 6.2: Instances information.

Table 6.3 shows the main and important conclusions: the adapted balancing RPW method produces better solutions, when compared with what happened in the real situation.

After balancing, next section will start with the results obtained so far, to start scheduling.

Instance	Real situation	Algorithm Result		
	N. used Operator	N. used Operator	N. Workstation	Time
Day-1	21	17	30	5 Sec
Day-2	12	7	10	30 Sec
Day-3	13	10	13	10 Sec
Day-4	11	8	12	20 Sec
Day-5	13	10	15	10 Sec

Table 6.3: Balancing result.

6.3 Tabu Search Results

As mentioned before, two TS methods were applied to the test instances.

As stated, the company starts creating boxes with size 10; in this work the size of boxes will change from sizes 7 to 13. Less than 7 will start imposing more movement of boxes in the line.

Two algorithm's results are illustrated in a way to depict which one is superior and they also show the effect of having different size boxes, in given days.

Computational results are shown in Table 6.4 to Table 6.10. The algorithm's results of TS-Two contain higher outcome in more instances than TS-One. Although the differences in quality may seem small, as the gaps are somewhat close to each other but, generally, TS-Two is superior in quality and almost in running times. In addition, none of the computational experiments performed for these larger sized instances has proved to reach a local optimum, as they all were still exploring neighbourhoods when the termination condition occurred.

Computational results for TS-One and TS-Two with changes the in maximum amount in each lot:

Instance	Algorithm Result-Maximum amount in initial lots is 7						
	Initial Cmax	TS-One	Time (H:M:S)	Improvement%	TS-Two	Time (H:M:S)	Improvement%
Day-1	25750	25188	00:01:58	2%	20470	00:01:32	21%
Day-2	25800	25800	00:57:19	0%	25200	00:54:39	2%
Day-3	25794	25740	00:28:31	0%	25200	00:27:50	2%
Day-4	25800	25740	00:43:28	0.2%	25390	00:22:23	2%
Day-5	25794	25453	00:02:17	1%	25259	00:02:12	2%

Table 6.4: Maximum amount in initial lots is 7.

Table 6.11 and Table 6.12 illustrate the impact of creating boxes with different sizes.

These tables show that creating boxes with varied sizes affects the quality of the solution. The tables also show that TS-Two gives better results than TS-One. In the case of TS-One, it is observed that boxes with size 10 offer superior results.

Instance	Algorithm Result-Maximum amount in initial lots is 8						
	Initial Cmax	TS-One	Time (H:M:S)	Improvement%	TS-Two	Time (H:M:S)	Improvement%
Day-1	25718	25056	00:01:05	3%	23395	00:00:52	9%
Day-2	25800	25800	01:03:58	0%	25440	00:33:13	1%
Day-3	25800	25800	00:18:02	0%	25440	00:17:43	1%
Day-4	25740	25740	00:19:43	0%	25360	00:11:30	1%
Day-5	25728	25441	00:03:08	1%	24714	00:01:16	4%

Table 6.5: Maximum amount in initial lots is 8.

Instance	Algorithm Result-Maximum amount in initial lots is 9						
	Initial Cmax	TS-One	Time (H:M:S)	Improvement%	TS-Two	Time (H:M:S)	Improvement%
Day-1	25769	24894	00:00:49	3%	22982	00:00:39	11%
Day-2	25800	25800	00:23:37	0%	25260	00:23:53	2%
Day-3	25710	25710	00:11:29	0%	25269	00:14:24	2%
Day-4	25650	25470	00:08:04	1%	25299	00:17:17	1%
Day-5	25695	25560	00:02:20	1%	25115	00:00:55	2%

Table 6.6: Maximum amount in initial lots is 9.

Instance	Algorithm Result-Maximum amount in initial lots is 10						
	Initial Cmax	TS-One	Time (H:M:S)	Improvement%	TS-Two	Time (H:M:S)	Improvement%
Day-1	25716	24762	00:00:24	4%	23394	00:00:18	9%
Day-2	25800	25800	00:19:56	0%	25800	00:27:08	0%
Day-3	25800	25380	00:07:54	1%	25380	00:09:45	2%
Day-4	25700	25700	00:13:16	0%	25200	00:05:07	2%
Day-5	25608	25266	00:01:23	2%	25424	00:00:35	1%

Table 6.7: Maximum amount in initial lots is 10.

Instance	Algorithm Result-Maximum amount in initial lots is 11						
	Initial Cmax	TS-One	Time (H:M:S)	Improvement%	TS-Two	Time (H:M:S)	Improvement%
Day-1	25740	25199	00:00:23	2%	25213	00:00:18	2%
Day-2	25800	25800	00:12:08	0%	25740	00:10:55	0%
Day-3	25740	25740	00:05:59	0%	25191	00:06:14	2%
Day-4	25780	25780	00:03:39	0%	24750	00:03:51	4%
Day-5	25783	25413	00:01:07	1%	25188	00:00:27	2%

Table 6.8: Maximum amount in initial lots is 11.

Instance	Algorithm Result-Maximum amount in initial lots is 12						
	Initial Cmax	TS-One	Time (H:M:S)	Improvement%	TS-Two	Time (H:M:S)	Improvement%
Day-1	25769	25394	00:00:16	1%	25027	00:00:12	3%
Day-2	25800	25800	00:24:58	0%	24270	00:08:09	6%
Day-3	25800	25440	00:04:07	1%	24852	00:14:16	4%
Day-4	25800	25800	00:03:01	0%	24720	00:02:39	4%
Day-5	25692	25240	00:00:49	2%	25220	00:00:20	2%

Table 6.9: Maximum amount in initial lots is 12.

Instance	Algorithm Result-Maximum amount in initial lots is 13						
	Initial Cmax	TS-One	Time (H:M:S)	Improvement%	TS-Two	Time (H:M:S)	Improvement%
Day-1	25631	25108	00:00:10	2%	24910	00:00:07	3%
Day-2	25800	25800	00:15:52	0%	25740	00:06:40	0%
Day-3	25800	25800	00:03:32	0%	25113	00:01:28	3%
Day-4	25740	25740	00:02:25	0%	25090	00:02:28	3%
Day-5	25482	25177	00:00:38	1%	24404	00:00:16	4%

Table 6.10: Maximum amount in initial lots is 13.

Instance	TS-One and Different Lot Sizes						
	7	8	9	10	11	12	13
Day-1	25188	25056	24894	24762	25199	25394	25108
Day-2	25800	25800	25800	25800	25800	25800	25800
Day-3	25740	25800	25710	25380	25740	25440	25800
Day-4	25740	25740	25470	25700	25780	25800	25740
Day-5	25453	25441	25560	25266	25413	25240	25177

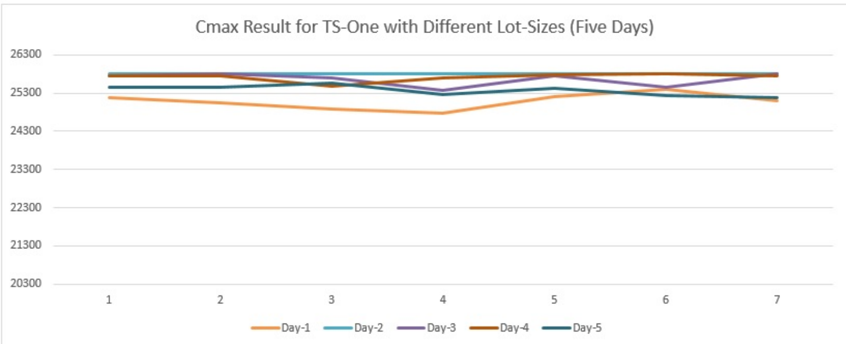


Table 6.11: TS-One and different lot sizes.

Instance	TS-Two and Different Lot Sizes						
	7	8	9	10	11	12	13
Day-1	20470	23395	22982	23394	25213	25027	24910
Day-2	25200	25440	25260	25800	25740	24270	25740
Day-3	25200	25440	25269	25380	25191	24852	25113
Day-4	25390	25360	25299	25200	24750	24720	25090
Day-5	25259	24714	25115	25424	25188	25220	24404

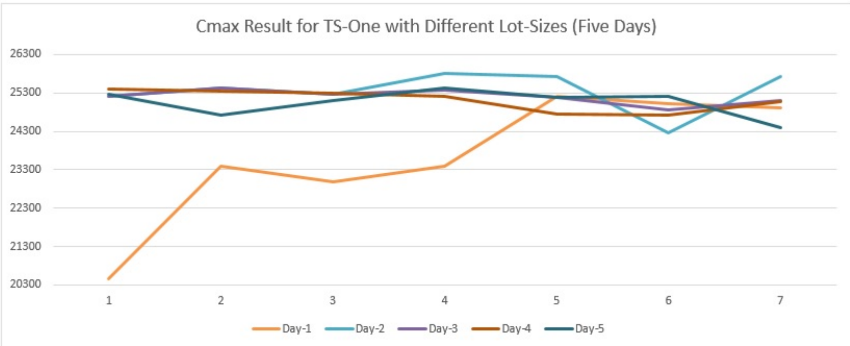


Table 6.12: TS-Two and different lot sizes.

Chapter 7

Conclusions and Future Work

This dissertation presented a study for solving a mixed-model assembly line problem and a lot-sizing problem with polyvalent operators and machines with different functions. The work was developed in close connection with a real situation occurring in a footwear company. A particular new stitching line was studied and some solution methods were devised, programmed and tested. At the same time, the possibility of extension and adaptation of the results to similar situations in other companies was taken into account.

The first objective was to determine the number of workstations which are needed for a certain level of production and then minimizing makespan for the created boxes. The dimension and complexity of the (whole) real problem was seen as high enough to resort to exact methods and so, the research addressed different approximate methods, also expecting results in a short time. The main contributions of this dissertation were first, identifying balancing and scheduling and lot-sizing problems, secondly devising a balancing algorithm, which is follows the RPW method but with different adaptations to the new problem and then using Tabu Search (TS) for the lot-sizing problem.

The solution approach consisted of creating an initial balance with a balancing algorithm which is a constructive heuristic and then impose one of dispatching rule in the scheduling phase. Improvements were attained by means of the TS based method. So, an application of TS was used to solve a lot-sizing problem.

By exploring to types of neighbourhoods two versions of the TS method were explored: TS-One and TS-Two.

TS-One worked only with one neighbourhood space but TS-Two used both spaces at the same time. TS-Two used more time but it was more efficient.

In addition various tested were performed to analyze the impact of changing the size of the lots while creating the boxes moving in the line.

Furthermore, here are three main ways to improve the efficiency of the algorithm. One of them is to study and implement new neighbourhood structures such as below:

1. Swap a box in a set with another set;

2. Transfer one item of a box to another box in another set;

The second neighbourhood space is very large and the other means to improve the efficiency of the algorithm consists of developing ways to speed up the evaluation of neighbours.

The third one is to improve the balance solution, since allocations in the scheduling phase was done according to the initial solution. Therefore, improving balancing solutions may improve the whole results.

It is expected that this work will motivate extensive further research, both in what concerns the footwear industry problems and heuristic and metaheuristic approaches. In addition, the solution methods will be now evaluated, in the production environment of the company.

Finally, this research project has still several open questions, whose answers will be obtained, hopefully, by future works.

References

[1] "The eu footwear industry." http://ec.europa.eu/growth/sectors/fashion/footwear/eu-industry/index$_e n.htm$.

[2] N. Boysen, M. Fliedner, and A. Scholl, "A classification of assembly line balancing problems," *European Journal of Operational Research*, vol. 183, pp. 674–693, 2007.

[3] N. Kriengkorakot and N. Pianthong, "The assembly line balancing problem :review articles," *KKU Engineering Journal*, vol. 34, pp. 133–140, 2007.

[4] E. Erel and S. C. Sarin, "A survey of the assembly line balancing procedures," *Production Planning and Control*, vol. 9, pp. 414–434, 1998.

[5] S. Ullah, G. Zailin, W. Baoxi, M. Jahanzeb, and H. Shiyang, "A survey on assembly lines and its types," *Front. Mech. Eng.*, vol. 9, pp. 95–105, 2014.

[6] C. Merengo, F. Nava, and A. Pozzetti, "Balancing and sequencing manual mixed-model assembly lines," *International Journal of Production Research*, vol. 37, pp. 2835–2860, 1999.

[7] A. Scholl and C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing," *European Journal of Operational Research*, vol. 168, pp. 666–693, 2006.

[8] J. C. Chen, C.-C. Chen, L.-H. Su, H.-B. Wu, and C.-J. Sun, "Assembly line balancing in garment industry," *Expert Systems with Applications*, vol. 39, pp. 10073–10081, 2012.

[9] S. O. Tasan and S. Tunali, "A review of the current applications of genetic algorithms in assembly line balancing," *Journal of Intelligent Manufacturing*, vol. 19, pp. 49–69, 2007.

[10] I. Baybars, "A suryay of exact algorithms for the simple assembly line balancing problem," *Management Science*, vol. 32, pp. 909–931, 1986.

[11] A. Scholl, *Balancing and sequencing of assembly lines*. Physica-Verlag Heidelberg, 1999.

[12] E. Falkenauer and A. Delchambre, "A genetic algorithm for bin packing and line balancing," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1186 – 1192, 1992.

[13] A. C. Alvim, C. C. Ribeiro, F. Glover, and D. J. Aloise, "A hybrid improvement heuristic for the one-dimensional bin packing problem," *Journal of Heuristics*, vol. 10, pp. 205–229, 2004.

[14] A. l. Gutjahr and G. L. Nemhauser, "An algorithm for the line balancing problem," *Management Science*, vol. 11, pp. 308–315, 1964.

[15] E. P. C. Kao and M. Queyranne, "On dynamic programming methods for assembly line balancing," *Informs*, vol. 30, pp. 375–390, 1982.

[16] A. Scholl and S. Vob, "Simple assembly line balancing–heuristic approaches," *217-244*, vol. 2, pp. 217–244, 1996.

[17] K. Fleszar and K. S. Hindi, "An enumerative heuristic and reduction methods for the assembly line balancing problem," *European Journal of Operational Research*, vol. 145, pp. 606–620, 2003.

[18] E. J. Anderson and M. C. Ferris, "Genetic algorithms for combinatorial optimization: The assembly line balancing problem," *ORSA Journal on Computing*, vol. 6, pp. 161–173, 1994.

[19] R. A. Khalil, L. Quinn, and D. J. Stockton, "Use of genetic algorithms in operations management: Part 1: applications," *Journal of Engineering Manufacture*, vol. 218, pp. 315–327, 2004.

[20] R. A. Khalil, L. Quinn, and D. J. Stockton, "Use of genetic algorithms in operations management: Part 2: results," *Journal of Engineering Manufacture*, vol. 218, pp. 329–343, 2004.

[21] Y. K. Kim, W. S. Song, and J. H. Kim, "A mathematical model and a genetic algorithm for two-sided assembly line balancing," *Computers and Operations Research*, vol. 36, p. Computers and Operations Research, 2009.

[22] R. T. Brown, Evelyn C.and Sumichrast, "Evaluating performance advantages of grouping genetic algorithms," *Engineering Applications of Artificial Intelligence*, vol. 18, pp. 1–12, 2005.

[23] A. Heinrici, *A comparison between simulated annealing and tabu search with an example from the production planning.* Springer Berlin Heidelberg, 1993.

[24] W.-C. Chiang, "The application of a tabu search metaheuristic to the assembly line balancing problem," *Annals of Operations Research*, vol. 71, pp. 209–227, 1998.

[25] S. D. Lapierre, A. Ruiz, and P. Soriano, "Balancing assembly lines with tabu search," *European Journal of Operational Research*, vol. 168, pp. 826–837, 2006.

[26] G. Suresh and S. Sahu, "Stochastic assembly line balancing using simulated annealing," *International Journal of Production Research*, vol. 32, pp. 1801–1810, 1994.

[27] B. Cakir, F. Altiparmak, and B. Dengiz, "Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm," *Computers and Industrial Engineering*, vol. 60, pp. 376–384, 2011.

[28] J. Bautista and J. Pereira, *Ant algorithm for assembly line balancing.* Springer Berlin Heidelberg, 2002.

[29] I. Sabuncuoglu, E. Erel, and A. Alp, "Ant colony optimization for the single model u-type assembly line balancing problem," *International Journal of Production Economics*, vol. 120, pp. 287–300, 2009.

[30] O. Kilincci, "Firing sequences backward algorithm for simple assembly line balancing problem of type 1," *Computers and Industrial Engineering*, vol. 60, pp. 830–839, 2011.

[31] W. Zhang, W. Xu, and M. Gen, "Multi-objective evolutionary algorithm with strong convergence of multi-area for assembly line balancing problem with worker capability," *Procedia Computer Science*, vol. 20, pp. 83–89, 2013.

[32] D. Sparling and J. Miltenburg, "The mixed-model u-line balancing problem," *International Journal of Production Research*, vol. 36, pp. 485–501, 1998.

[33] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," *European Journal of Operational Research*, vol. 168, pp. 694–715, 2006.

[34] E. Erel and H. Gokcen, "Shortest-rout formulation of mixed-model assembly line balancing problem," *European J. of Operational Research*, vol. 116, pp. 194–204, 1999.

[35] H. Gokcen and E. Erel, "Binary integer formulation for mixed-model assembly line balancing problem," *Computers and Industrial Engineering*, vol. 34, pp. 451–461, 1998.

[36] J. Bukchin, E. M. Dar-El, and J. Rubinovitz, "Mixed-model assembly line design in a make-to-order environment," *Computers and Industrial Engineering*, vol. 41, pp. 405–421, 2002.

[37] J. C. Chen, H. Parwananta, T. D. D. Thao, K.-J. Wang, M. Anityasari, and C.-C. Chen, "Branch and bound method for solving line balancing problem in footwear sewing line," in *Industrial Engineering and Management Systems Conference*, pp. 1061–1069, 2012.

[38] N. Boysen, M. Fliedner, and A. Scholl, "Assembly line balancing: Which model to use when?," *International Journal of Production Economics*, vol. 111, pp. 509–528, 2008.

[39] P. M. Simaria, Ana Sofia Vilarinho, "A genetic algorithm based approach to the mixed-model assembly line balancing problem of type ii," *Computers and Industrial Engineering*, vol. 47, pp. 391–407, 2004.

[40] A. N. Haq, K. Rengarajan, and J. Jayaprakash, "A hybrid genetic algorithm approach to mixed-model assembly line balancing," *The International Journal of Advanced Manufacturing Technology*, vol. 28, pp. 337–341, 2005.

[41] A. A. Mamun, A. A. Khaled, S. M. Ali, and M. M. Chowdhury, “A heuristic approach for balancing mixed-model assembly line of type i using genetic algorithm,” *International Journal of Production Research*, vol. 50, pp. 5106–5116, 2012.

[42] A. Roshani, P. Fattahi, A. Roshani, M. Salehi, and A. Roshani, “Cost-oriented two-sided assembly line balancing problem: A simulated annealing approach,” *International journal of computer integrated manufacturing*, vol. 25, pp. 1–27, 2012.

[43] P. R. McMullen and P. Tarasewich, “Using ant techniques to solve the assembly line balancing problem,” *IIE Transactions*, vol. 35, pp. 605–617, 2003.

[44] P. M. Vilarinho and A. S. Simaria, “Antbal: an ant colony optimization algorithm for balancing mixed-model assembly lines with parallel workstations,” *International Journal of Production Research*, vol. 44, pp. 291–303, 2006.

[45] M. T. Costa and J. S. Ferreira, “A simulation analysis of sequencing rules in a flexible fowline,” *European Journal of Operational Research*, vol. 119, pp. 440–450, 1999.

[46] M. Pinedo and X. Chao, *Operations Scheduling with Applications in Manufacturing and Services*. 1999.

[47] K. R. Baker, *Introduction to sequencing and scheduling*. Wiley, 1974.

[48] P. Brucker, *Scheduling algorithms*. Springer, 2007.

[49] N. Boysen, M. Fliedner, and A. Scholl, “Sequencing mixed-model assembly lines: Survey, classification and model critique,” *European Journal of Operational Research*, vol. 192, pp. 349–373, 2009.

[50] F. F. Boctor, J. Renaud, A. Ruiz, and S. Tremblay, “Optimal and heuristic solution methods for a multiprocessor machine scheduling problem,” *Computers and Operations Research*, vol. 36, pp. 2822–2828, 2009.

[51] W. Ng and K. Mak, “A branch and bound algorithm for scheduling just-in-time mixed-model assembly lines,” *International Journal of Production Economics*, vol. 33, pp. 169–183, 1994.

[52] K. Fan and R.-q. Zhang, “An analysis of research in job shop scheduling problem (2000-2009),” in *Advanced Management Science (ICAMS), 2010 IEEE International Conference on*, pp. 282–288, 2010.

[53] T.-C. Chiang and L.-C. Fu, “Using dispatching rules for job shop scheduling with due date-based objectives,” in *IEEE International Conference on Robotics and Automation*, pp. 1426–1431, 2006.

[54] O. Holthaus and C. Rajendran, “Efficient dispatching rules for scheduling in a job shop,” *Production Economics*, vol. 48, pp. 87–105, 1997.

[55] S. J. Mason, J. W. Fowler, and W. Matthew Carlyle, “A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops,” *Journal of Scheduling*, vol. 5, pp. 247–262, 2002.

[56] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms, part ii: hybrid genetic search strategies," *Computers and Industrial Engineering*, vol. 36, pp. 343–364, 1999.

[57] S. A. Mansouri, "A multi-objective genetic algorithm for mixed-model sequencing on jit assembly lines," *European Journal of Operational Research*, vol. 167, pp. 696–716, 2004.

[58] X. Cai and K. N. Li, "A genetic algorithm for scheduling staff of mixed skills under multi-criteria," *European Journal of Operational Research*, vol. 125, pp. 359–369, 2000.

[59] P. Brandimarte, "Routing and scheduling in a flexible job shop by tabu search," *Annals of Operations Research*, vol. 41, pp. 157–183, 1993.

[60] E. Nowicki and C. Smutnicki, "An advanced tabu search algorithm for the job shop scheduling problem," *Journal of Scheduling*, vol. 8, pp. 145–159, 2005.

[61] M. Saidi-Mehrabad and P. Fattahi, "Flexible job shop scheduling with tabu search algorithms," *The International Journal of Advanced Manufacturing Technology*, vol. 32, pp. 563–570, 2007.

[62] C. Zhang, P. Li, and Y. Guan, ZaiLin Rao, "A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem," *Computers and Operations Research*, vol. 34, pp. 3229–3242, 2007.

[63] C. Y. Zhang, P. Li, Y. Rao, and Z. Guan, "A very fast ts/sa algorithm for the job shop scheduling problem," *Computers and Operations Research*, vol. 35, pp. 282–294, 2008.

[64] J.-q. Li, Q.-k. Pan, and Y.-C. Liang, "An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems," *Computers and Industrial Engineering*, vol. 59, pp. 647–662, 2010.

[65] P. R. Mcmullen and G. V. Frazier, "A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line," *IIE Transactions*, vol. 32, pp. 679–686, 2000.

[66] R. K. Suresh and K. M. Mohanasundaram, "Pareto archived simulated annealing for job shop scheduling with multiple objectives," *The International Journal of Advanced Manufacturing Technology*, vol. 29, pp. 184–196, 2006.

[67] B. Almada-Lobo, D. Klabjan, M. A. Carravilla, and J. F. Oliveira, "Multiple machine continuous setup lotsizing with sequence-dependent setups," *Computational Optimization and Applications*, vol. 47, pp. 529–552, 2009.

[68] B. Fleischmann and H. Meyr, "The general lotsizing and scheduling problem," *OR Spektrum*, vol. 19, pp. 11–21, 1997.

[69] H. Meyr, "Simultaneous lotsizing and scheduling by combining local search with dual reoptimization," *European Journal of Operational Research*, vol. 120, pp. 311–326, 2000.

[70] H. Meyr, "Simultaneous lotsizing and scheduling on parallel machines," *European Journal of Operational Research*, vol. 139, pp. 277–292, 2002.

[71] S. Petrovic, C. Fayad, D. Petrovic, E. Burke, and G. Kendall, "Fuzzy job shop scheduling with lot-sizing," *Annals of Operations Research*, vol. 159, pp. 275–292, 2008.

[72] L. Buschkühl, F. Sahling, S. Helber, and H. Tempelmeier, "Dynamic capacitated lot-sizing problems: a classification and review of solution approaches," *OR Spectrum*, vol. 32, pp. 231–261, 2010.

[73] D. Battini, M. Faccio, A. Persona, and F. Sgarbossa, "Balancing-sequencing procedure for a mixed model assembly system in case of finite buffer capacity," *The International Journal of Advanced Manufacturing Technology*, vol. 44, pp. 345–359, 2009.

[74] H. Mosadegh, S. M. T. F. Ghomi, and M. Zandieh, "Simultaneous solving of balancing and sequencing problems in mixed-model assembly line systems," *International Journal of Production Research*, vol. 50, pp. 4994–5016, 2012.

[75] H. Mosadegh, M. Zandieh, and S. M. T. F. Ghomi, "Simultaneous solving of balancing and sequencing problems with station-dependent assembly times for mixed-model assembly lines," *Applied Soft Computing*, vol. 12, pp. 1359–1370, 2012.

[76] C. Öztürk, S. Tunalı, B. Hnich, and A. Örnek, "Balancing and scheduling of flexible mixed model assembly lines with parallel stations," *The International Journal of Advanced Manufacturing Technology*, vol. 67, pp. 2577–2591, 2013.

[77] E.-G. Talbi, *Metaheuristics from design to implementation*, vol. 10. Wiley Publishing, 2009.

[78] S. P. Bradley, A. C. Hax, and T. L. Magnanti, *Applied mathematical programming.* Addison-Wesley Publishing Company, 1997.

[79] Z. Michalewics and D. B. Fogel, *How to solve it: Modern heuristics.* Springer, 2000.

[80] T. Sawik, *Scheduling in Supply Chains Using Mixed Integer Programming.* Wiley, 2011.

[81] S. T. Ghutukade1 and D. S. M. Sawant, "Use of ranked position weighted method for assembly line balancinh," *International Journal of Advanced Engineering Research and Studies*, 2013.

[82] M. Gendreau and J.-Y. Potvin, *Handbook of Metaheuristics.* International Series in Operations Research and Management Science, 2010.

Appendix A

List of acronyms

JSP	Job Shop Problem
JSSP	Job Shop Scheduling Problem
ALP	Assembly Line Problem
ALBP	Assembly Line Balancing Problem
SALBP	Simple Assembly Line Balancing Problem
GALBP	Generalized Assembly Line Balancing Problem
MALBP	Mixed-model Assembly Line Balancing Problem
UALBP	U-shape Assembly Line Balancing Problem
SMD	Single Model Deterministic
SMS	Single Model Stochastic
MMD	Multi/Mixed-Model Deterministic
MMS	Multi/Mixed-Model Stochastic
HYB	Hybrid
B&B	Branch and Bound
DP	Dynamic Programming
GA	Genetic Algorithm
EA	Evolutionary Algorithm
GGA	Grouping Genetic Algorithm
TS	Tabu Search
SA	Simulated Annealing
ACO	Ant Colony
PASA	Pareto Archived Simulated Annealing
LS&S	Lot-Sizing and Scheduling
MIP	Mixed Integer Programming
MILP	Mixed Integer Linear Programming
WIP	Work In Process
ERD	Earliest Release Date
EDD	Earliest Due Date
MS	Minimum Slack
LPT	Longest Processing Time
WSPT	Weighted Shortest Processing Time
CP	Critical Path