FEDERAL UNIVERSITY OF TECHNOLOGY – PARANÁ
GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING

ADALBERTO SATO MICHELS

DESIGNING INDUSTRIAL PRODUCTION LAYOUTS: AN APPLICATION ON ROBOTIC WELDING ASSEMBLY LINES

MASTER THESIS

CURITIBA
2017

ADALBERTO SATO MICHELS

DESIGNING INDUSTRIAL PRODUCTION LAYOUTS: AN APPLICATION ON ROBOTIC WELDING ASSEMBLY LINES

Dissertation presented to Graduate Program in Electrical and Computer Engineering from Federal University of Technology – Paraná as a partial requirement to obtain the degree of “Master’s degree in Science” – Research Area: Automation and Systems.

Advisor: Prof. Dr. Leandro Magatão

CURITIBA
2017

Dados Internacionais de Catalogação na Publicação

Michels, Adalberto Sato
M623d Designing industrial production layouts : an application on
2017 on robotic welding assembly lines / Adalberto Sato Michels.--
2017.
73 f. : il. ; 30 cm.

Texto em inglês com resumo em português
Disponível também via World Wide Web
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Curitiba, 2017
Bibliografia: f. 68-73

1. Programação linear. 2. Métodos de linha de montagem. 3. Soldagem. 4. Indústria automobilística – Linha de produção – Layout. 5. Soldagem – Automação. 6. Engenharia elétrica – Dissertações. I. Magatão, Leandro. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. III. Título.

CDD: Ed. 22 – 621.3

Biblioteca Central da UTFPR, Câmpus Curitiba

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Câmpus Curitiba

Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

Título da Dissertação Nº. ____

Designing Industrial Production Layouts: An Application on Robotic Welding Assembly Lines

por

Adalberto Sato Michels

Orientador: Prof. Dr. Leandro Magatão (UTFPR)

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM CIÊNCIAS – Área de Concentração: **ENGENHARIA DE AUTOMAÇÃO E SISTEMAS** do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR, às 14**h** do dia 31 de maio de 2017. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores doutores:

Prof. Dr. Leandro Magatão (Presidente – UTFPR)	Prof. Dr. Cassius Tadeu Scarpin (UFPR)
Prof. Dr. Ricardo Lüders (UTFPR)	

Visto da coordenação:

Prof. Jean Carlos Cardozo da Silva, Dr.
(Coordenador do CPGEI)

Acknowledgements

In the first place, I would like to thank my parents: Solange and Nelson. To my mother, for the constant encouragement to always keep studying, even though she had already stopped understanding a few years ago about what I have been dedicated to. To my father, for always being proud of me and each of my achievements, reminding me how much they were important to him as well.

To my advisor, Professor Leandro Magatão, for the hours of dedication in meetings, work reviews and guidance throughout the master's degree. He certainly is the most dedicated professor I have ever met, besides an admirable person.

To my friends that make everything worth it at all times. From high school, Eduardo and Samir, for these eleven years of friendship, filled with countless bad taste jokes that never get old, no matter how many times they are told. Each of them is a brother from a different mother. From college, Tales, Juliano Pereira and Renato, for the companionship in all good and bad moments of university life, they have made this stage of my existence (in and out of classroom) easier and more fun. Also from college, Julio and Juliano Luiz, for the extensive and enjoyable talks about trips, bicycles and our routines. From my exchange in Canada, Nana, Mateus and César, for keeping in touch after everything we lived together in that wonderful country. From graduate school, Celso and Thiago, for the initial advices when I first got into the team and research collaborations along with it as colleagues, as well as for the pleasant company in social events as brewing partners. In addition, to friends that life recently brought me: Luiza, for the daily love, emotional support, and for always bringing an optimistic point of view into my life in every situation, Jéssica, Vinícius, Adriane, Jan and Celso (again) for the companionship in trips, at weekends, festivals and night-outs. I own them my most memorable moments, sanity and valuable (or meaningless) discussions.

Lastly, I am thoroughly grateful for the opportunity to study at UTFPR/CPGEI and to the public institutions CNPq and Fundação Araucária for the financial support during my exchange program and master's studies, respectively.

Por último, mas não menos importante, agradeço a Deus por, mesmo sendo onipotente, não ter atrapalhado (desta vez). Em tempo: fora Temer!

Sponsorship

The author acknowledges the financial support of Fundação Araucária and Renault do Brasil – Agreement 21/2016 (FA - UTFPR - RENAULT).

"In a dark place we find ourselves, and a little more knowledge lights our way."

(Master Yoda)

Resumo

As linhas de montagem que envolvem soldagem por pontos estão amplamente presentes na indústria de fabricação automotiva. O processo de montagem da estrutura do veículo emprega vários robôs equipados com ferramentas de soldagem por pontos. Estes robôs e ferramentas possuem altos custos, fazendo surgir a necessidade de projetar a linha cuidadosamente, atendendo à demanda do produto e reduzindo as despesas ao mesmo tempo. Nesta dissertação, propõe-se e estuda-se o problema do Projeto de Linha de Montagem Robótica (PLMR), com base nas características práticas de uma empresa automotiva localizada no Brasil. Desenvolve-se um modelo de Programação Linear Inteira Mista (PLIM) que permite: (i) paralelização de estações, (ii) seleção de equipamentos e (iii) múltiplos robôs por estação de trabalho. O modelo matemático visa minimizar o custo total à uma taxa de produção desejada, o que envolve robôs, ferramentas e instalações. O modelo proposto considerou o tempo morto durante um ciclo, restrições de espaço, restrições de alocação de tarefas e possibilidades de paralelismo. O tempo morto é um tempo improdutivo e fixo do trabalho de manuseio de peças que acompanha o tempo de movimento dos robôs transportadores capacitados. Experimentos computacionais foram realizados para evidenciar a influência dos parâmetros sobre a solução ótima do projeto de linha. Além disso, foram realizados estudos de casos práticos com parâmetros reais coletados de uma linha de montagem robotizada para soldagem, localizada na região metropolitana de Curitiba-PR, chegando à otimalidade. Em comparação com as linhas estritamente seriais, o modelo mostrou grandes vantagens ao permitir paralelizar as estações no sistema de produção, permitindo avaliar uma compensação entre a taxa de produção e o custo total.

PALAVRAS-CHAVE

Programação linear inteira mista; Problema do projeto de linhas de montagem; Linhas de montagem robotizadas para soldagem; Estações paralelas; Seleção de equipamentos; Tempo morto

Abstract

Spot welding assembly lines are widely present in the automotive manufacturing industry. The procedure of building the vehicle's body employs several robots equipped with spot welding tools. These robots and tools are quite costly, arising the necessity of designing the line consciously, meeting the product demand and reducing expenses at the same time. In this master thesis, the Robotic Assembly Line Design (RALD) problem is proposed and studied based on practical characteristics from an automotive company located in Brazil. A Mixed-Integer Liner Programming (MILP) formulation is developed allowing: (i) station paralleling, (ii) equipment selection, and (iii) multiples robots per workstation. The mathematical model aims at minimising the total cost at the desired production rate, which involves robots, tools and facilities. The proposed model considered dead time during a cycle, space constraints, task assignment restrictions, and parallelism possibilities. Dead time is an unproductive and fixed work-piece handling time included in the capacitated transporter robots' movement time. Computational experiments were performed in order to evidence the parameters' influence over the optimal line design solution. In addition, practical case studies were conducted with parameters collected from a real-world robotic welding assembly line located on the outskirts of Curitiba-PR (Brazil), reaching optimality. Compared to strictly serial lines, the model led to great advantages by allowing parallel station in the production system, making it possible to evaluate an expected trade-off between production rate and total cost.

KEYWORDS

Mixed-integer linear programming; Assembly line design problem; Robotic welding assembly line; Parallel stations; Equipment selection; Dead time

List of Figures

List of Tables

List of Labels

ALBP	Assembly Line Balancing Problem
ALDP	Assembly Line Design Problem
CT	Cycle Time
DT	Dead Time
GA	Genetic Algorithm
GALBP	General Assembly Line Balancing Problem
LB	Lower Bound
MILP	Mixed-Integer Linear Programming
RALB	Robotic Assembly Line Balancing
RALD	Robotic Assembly Line Design
RSW	Resistance Spot Welding
SALBP	Simple Assembly Line Balancing Problem
SH	Simplification Hypotheses
UB	Upper Bound

Contents

Chapter 1

Introduction

1.1 Overview

The automotive industry is a segment in which robotic welding assembly lines are widely used. Such lines are designed and implemented with years of planning; however, the line's configuration is rarely guided by modelling and solution approaches provided by the Operational Research (OR). The optimisation conducted by OR seeks, for instance, to abstract as faithfully as possible real-world problems using mathematical models, and solve them by applying computational methods, in order to obtain the best answer for several configurations of systems or operations.

For "best answer", one can take into account several factors: the greatest efficiency or productivity of a system or operation, the lowest cost or time to carry out a project, among others. All of this depends on what criteria are being considered at the time the problem will be solved. Currently, the vast majority of industrial systems operate under sub-optimal conditions. A long-term application proposal, therefore, is to formally represent the available resources of a company in a mathematical model capable of responding which is the best system design to be configured in order to meet the desired requirements. The optimised design is a fundamental issue to allow an industrial system to operate in an optimal condition, according to the adopted criterion.

The process of designing production lines is present at a strategic level of global decisions. Before the implementation of a new line, several studies are carried out and a large amount of data is collected, including the probable installation costs by consulting suppliers' prices, the available space for the construction of this new line, market demand surveys are conducted for the new product, among other factors. After this step, the design of the line actually begins, which could be guided by an optimization process that uses the previously collected data. This optimization process consists in modelling the industrial characteristics and solving the model by computational methods, so that, the best manner to allocate resources (capital, time, etc.) can be found. The optimal answer in this scenario, therefore, would be the one that allows the minimisation of costs for the line design that provides the aimed productivity.

This master's thesis focuses on the automotive industry, where there is already a research effort for possible applications of optimization methods for the production line reconfiguration (Chapter 2). However, the studies carried out here apply more specifically to the design of robotic welding assembly lines, illustrated in Figure 1.1, a field that still lacks studies due to the gap between literature and practical problems.

Figure 1.1: Example of a robotic welding assembly line. A robot holding a spot welding tool (1) and a vehicle's body (2) are illustrated.

Some factors contribute to this gap. The several welding tasks, which imply both in the possibility of leaner arrangements in the line and physical limitations in the assembly order is one of them. The use of different robots and tools for each assembly stage, the movement of the work pieces to be produced, and the difficulties in measuring the trade-off between costs and productive efficiency in the robotic assembly line are also practical characteristics that make the problem hard to be modelled. These extensions are further discussed in Chapter 3.

Thus, a new problem class arises: the Robotic Assembly Line Design (RALD) problem, which is proposed and studied throughout this master's thesis. The problem's main idea consists in designing a fully automated assembly line at the lowest cost, taking into account cost parameters and practical features found in the automotive industry, so that the model generates answers that guide the implementation of new configurations.

1.2 Objectives

1.2.1 Main Objective

The main objective of this master's thesis is to propose, study and model the Robotic Assembly Line Design (RALD) problem. The specific objectives are detailed as follows in Section 1.2.2.

1.2.2 Specific Objectives

- Provide a literature review and identify the gap for the studied problem;
- Propose a new class of problem, the RALD problem, considering practical extensions;
- Develop a mathematical model in an MILP formulation that describes the proposed problem accurately;
- Validate the model on real-world based datasets in computational tests;
- Apply the model on automotive industry case studies, achieving solutions that provide a potential economy.

1.3 Motivation and Contributions

1.3.1 Optimisation Industrial Project

The complete industrial project in which this master's thesis is embedded has been sponsored by Fundação Araucária and Renault do Brasil under agreement 21/2016, and developed in cooperation with this company, located on the outskirts of Curitiba.

The project regarding the line design optimisation is amongst other industrial projects previously conducted in the company, which also involves balancing, vehicle sequencing and implementation scheduling. The combination of such projects are part of the automotive industry plans to adapt the body-in-white welding procedures of all vehicle models into a unified production system, which was denominated *mono-flux* project and cover the optimisation potential gains in the said sector.

On the balancing problem, Sikora, Lopes, Schibelbain and Magatão (2017) proposed an MILP formulation that solves the Simple Assembly Line Balancing (ALBP) problem by using integer variables instead of binary ones to treat multiple identical tasks, which works better for resistance spot welding (RSW) procedures, since the car body assembly tasks can be gathered into groups of similar points with the same properties. Also applying the simplified formulation, Lopes, Sikora, Molina, Schibelbain, Rodrigues and Magatão (2017) present a balancing model for the robotic spot welding manufacturing line problem, in which several practical extensions such as equipment accessibility, movement time between welding points and interference among multiple robots assigned to the same workstation were included.

On the vehicle sequencing problem, the first step to integrate the product sequence of mixed-model lines into the balancing model was taken by Lopes, Sikora and Magatão (2016) in a congress paper. This model is aware of the product sequence for different vehicle models and the buffer layout found in the line. Further development of this work is presented in Section 1.3.2.

However, the solutions for optimal configuration could not be applied into the production system due to the massive amount of required modifications caused by the necessary task reallocation. Therefore, the implementation scheduling model was proposed by Sikora et al. (2016), which divides the optimal configuration implementation into smaller phases.

The line design problem was included into the project with the idea that the production layout could be planned since from the beginning, in order to avoid further reconfigurations. So far in the studied industrial project, the production layout has been considered with defined workstations, robot allocations and fixed sub-assemblies. Nonetheless, according to Baybars (1986), by relaxing this layout simplification hypothesis one has an assembly line design problem at hand, which is the main topic in the author's work (Section 1.3.2) and in this master's thesis.

1.3.2 Author's Works

Firstly presented as a poster congress publication (Lopes, Michels, Molina, Sikora and Magatão, 2016), the Robotic Assembly Line Design (RALD) problem model has been enhanced in order to allow it to choose the assembly line layout at the lowest cost. This last formulation takes into account characteristics of robots, equipment and facilities, decides the number of stations, and allocates multiple robots in each of them when it is necessary. All these characteristics were addressed in a cost-oriented model. Concomitantly, station paralleling possibilities are allowed to diminish the piece handling time negative effects, and equipment selection is done depending on which tasks are to be performed. This improved work had analysed the parameters' sensibility and had been applied on an industrial study case. Currently, it has been submitted as a journal paper and is under review (Michels et al., 2017). In Chapter 3, the RALD problem is thoroughly discussed and in Chapter 4 its mathematical model is proposed.

Moreover, for the project's balancing and sequencing core, a journal paper was submitted and is under review (Lopes, Sikora, Michels and Magatão, 2017b). This paper analyses several goal functions for the balancing of mixed-model lines and still takes into account given sequencing and buffer allocations. However, it has been repeatedly stated that dealing with balancing, sequencing and buffer allocation at the same time would result in better solutions. Therefore, a model that decides these problems together was proposed and is under review (Lopes, Sikora, Michels and Magatão, 2017a).

Along with the projects, a linear programming model was proposed for the design of integrated renewable energy systems for villages in Brazil (Michels et al., 2016). This work was also presented in a conference.

Given the aforementioned papers, the author's work is embedded in an automotive industrial project involving balancing, sequencing and designing robotic welding assembly lines. The main author's contribution within this scenario is related to the line design branch, such contribution is hereafter exploited in the presented master's thesis.

1.4 Document Outline

This master's thesis is divided into six chapters, including this introduction (Chapter 1), whereas the remaining document outline is described as follows.

Chapter 2 is a detailed review on Assembly Line Balancing Problem (ALBP). A set of simplification hypotheses (SH) defines the Simple Assembly Line Balancing Problem (SALBP), and, according to Baybars (1986), the relaxation of any (or a combination of some) of these hypotheses defines the General Assembly Line Balancing Problem (GALBP). Nevertheless, when costs for stations, operations and equipment are considered, the problem is no longer a balancing problem, and turns into a designing problem, which is denominated as Assembly Line Design Problem (ALDP). In each case, comprehensive surveys and several papers with different solution methods are reviewed. Moreover, previous works related to cost-oriented models, equipment selection, parallelism possibilities, unproductive movement time (dead time), multi-manned workstations, applications and procedures of resistance spot welding (RSW) tasks, and multiple identical tasks are posed.

In Chapter 3, the RALD problem is proposed. To the best knowledge of the author, this can be considered a new class of problems. Each practical extension considered on the RALD problem is explained in detail, justifying the benefits and the necessity of their inclusion.

The mathematical model is presented in Chapter 4 and was implemented in a Mixed-Integer Linear Programming (MILP) formulation. Throughout this chapter, each expression has its use explained and the reason of why it was included in the formulation is stated.

The real-world based adapted dataset for robotic welding assembly lines with multiple identical tasks on which the model was applied is presented in Chapter 5. Along with that, the computational and practical results are discussed. Computational tests show the model's sensibility for parameters' variation by demonstrating the effects on the production system layout once the line designing cost ratios and unproductive times are altered. In addition, industrial study cases are solved and compared to the as-build configuration functioning nowadays in the company.

Lastly, in Chapter 6, concluding remarks are summarised along with the main contributions of this master's thesis. Future research directions are also suggested.

Chapter 2

Bibliography Review

This chapter presents a bibliography review for production layouts, the (Simple and General) Assembly Line Balancing Problem, the Assembly Line Design Problem. Simplification hypotheses, solution methods, and problems' variations with practical extensions are presented. Besides, stations paralleling, equipment selection and resistance spot welding procedures are further exhibited in order to more specifically define the proposed problem.

2.1 Production Layout

According to Krajewski et al. (2013), the most usual forms of layout design used in production systems are: flow-shop, job-shop, fixed position and hybrid. Their organisations depend on the product variety, volume, and complexity, and the layout pattern is chosen for a better system efficiency.

2.1.1 Flow-shop Layout

Production systems used in the automotive industry are frequently based on flow-shop layouts, which are considered to be product-oriented design, i.e. the machinery and the operations are organised for the product continuous, synchronous or asynchronous flow (see Figure 2.1). Assembly lines in a flow-shop configuration are generally dedicated to given homogeneous products, enabling the mass production of them. The advantages of this production layout for regular operation are, among others, the high system utilisation, constant and simple material flowing and handling, low setup time and high throughput. However, the unavoidable drawbacks that comes along with the flow-shop configuration are due to the costly capital investment and the line stoppage. The entire system is shut down temporarily due to maintenance and failure as each stage is dependent on the integral system's operation for the proper product flow (Krajewski et al., 2013).

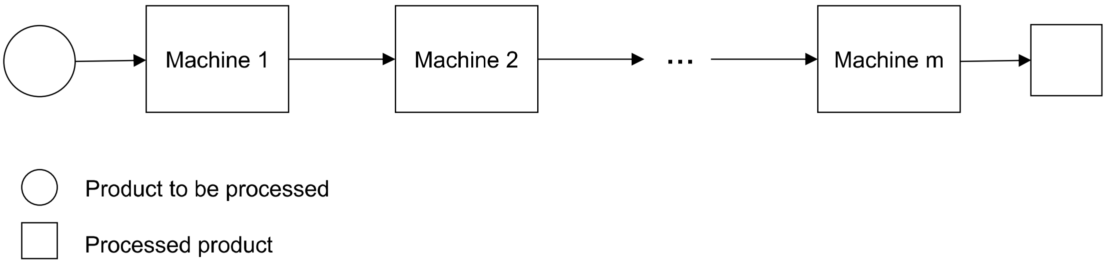

Figure 2.1: Example of flow-shop.

Designing and balancing a flow-shop assembly line is a long-term decision precisely because of the high costs and utilisation of the line, which makes it an important problem that can be aided by optimisation methods in order to achieve potential economy and better efficiency. Sequencing and scheduling problems might arise if this layout is employed to a family of similar products.

2.1.2 Other Layouts

Job-shop layouts are used for high flexibility systems. When a wide set of products have to be produced for low volumes in order to meet uncertain demand rates, this type of configuration is adopted to provide product customisation at reduced installing costs in the same physical space. The product path complexity is the main disadvantage of this layout, making mass production impracticable due to difficult product piece handling and orientation. Figure 2.2 is an example of job-shop in which four different products are processed simultaneously by using different machines and paths. Machine positing and product scheduling optimisation problems arise along with the use of this production layout (Krajewski et al., 2013).

The fixed position layout is a very specific configuration that is applied on huge dimension products, such as airplanes, ships, and the construction of building. In both

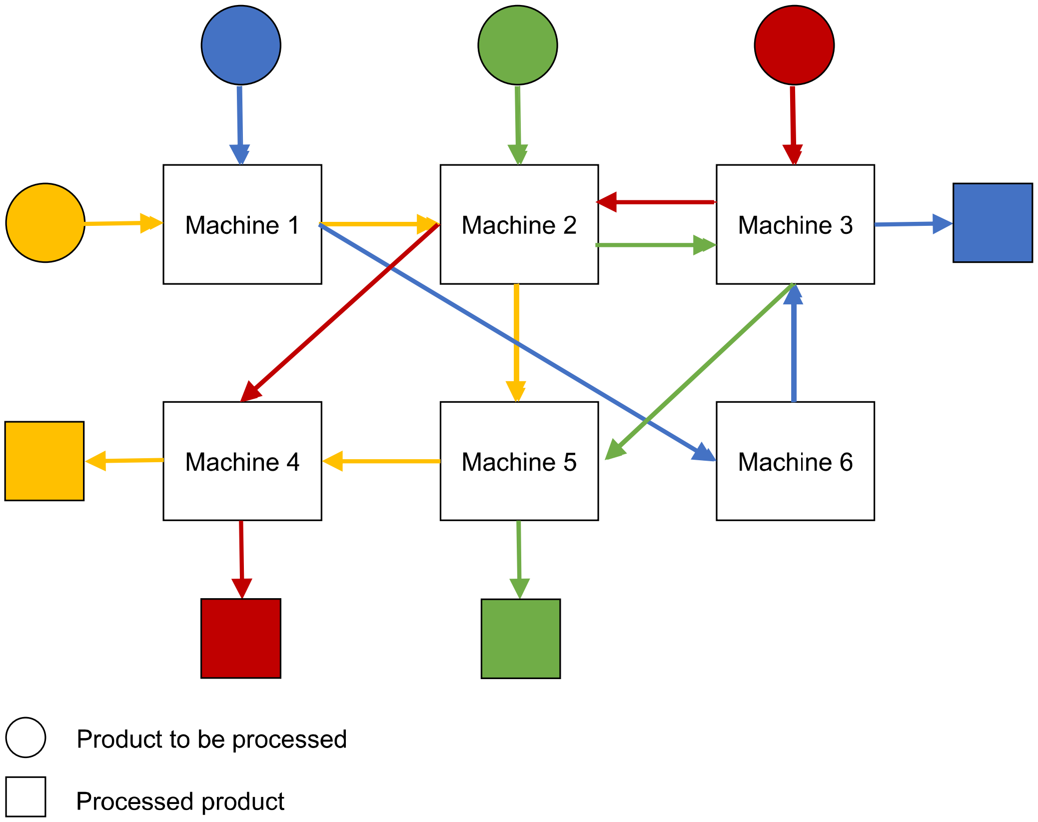

Figure 2.2: Example of job-shop. Notice that the products use different machines in different orders, that is, they take different paths.

cases, machines are moved and operations are performed around the product, which is kept static since the movement of the product as a piece is infeasible. This concept is depicted in Figure 2.3, a situation in which labour, equipment and material resources are taken into the construction yard. Project scheduling problems are observed in this system (Krajewski et al., 2013).

Hybrid layouts are a combination of flow-shops and job-shops, ideally used in industries with a high variety of products that still have very similar features (Krajewski et al., 2013).

2.2 Assembly Line Balancing Problem

As stated in Section 2.1.1, assembly lines in automotive industries are production systems based on flow-shop layout and, therefore, they are product-oriented systems. This kind of configuration has given rise to the traditional Assembly Line Balancing Problem

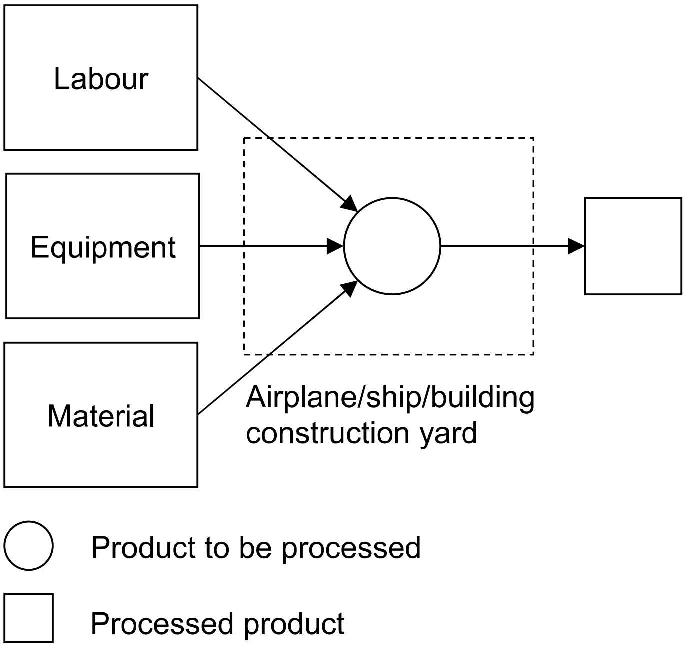

Figure 2.3: Example of fixed position layout.

(ALBP), vastly approached in the literature, as indicated in, for instance, Scholl (1999), Scholl and Becker (2006), Becker and Scholl (2006), Boysen et al. (2007) and Battaïa and Dolgui (2013).

An assembly line is composed of a set of minimum rational work element (hereafter referred to as *task*) for the product assemblage. Tasks are allocated into a set of workstations (*stations*, hereafter) linked together by a transport system in order to assure the product flow from one station to another (Baybars, 1986).

Tasks are the smallest indivisible work element and require an amount of *process time* to be performed, which are known as parameters. These tasks must be performed at stations by humans or robots operators using specific equipment (machinery or tools).

The line's *cycle time* is defined by the most loaded station in serial lines. The production rate is determined by calculating how much time one unit of the finished product takes to emerge from the last station along the line (Scholl, 1999).

Another characteristic of assembly lines is that tasks cannot be performed in an arbitrary order, they might have technological sequencing requirements named *precedence relations*. When all precedence relations are known, they can be schematically represent by a directed acyclic graph, namely precedence diagram. Figure 2.4 (Scholl and Becker, 2006) shows a precedence diagram with 10 tasks (number inside the circle) with process times between 2 and 9 time units (top-right corner). Task 5 is chosen to exemplify how the precedence constraint works: in order to process task 5, tasks 1 and 4 (direct predecessors) and 3 (indirect predecessor) are required to be completed. On the other way round, task 5 must be finished before its direct and indirect successors (6, 8, 9 and 10) are started.

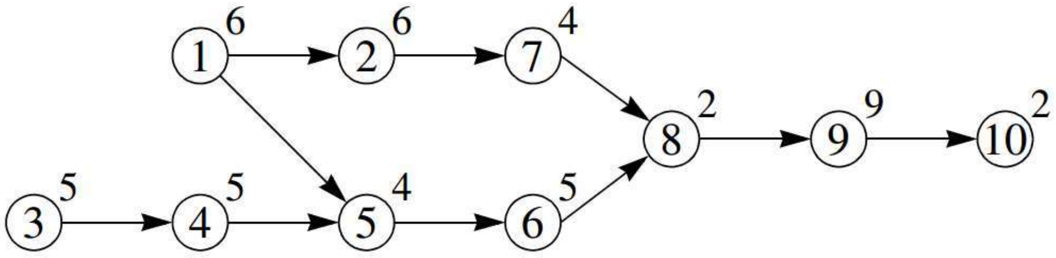

Figure 2.4: Example of a precedence diagram: the number inside the circle represents the task number and at the top-right corner the task process time. Source: Scholl and Becker (2006).

Even though the line's cycle time is defined by the most loaded station (*bottleneck*), it can be stated that each station has its own cycle time, which might be lower than the global cycle time. This time slack between the station's cycle time and the bottleneck's cycle time is considered unproductive and is called *idle time*. Figure 2.5 presents three stations: the first one is the bottleneck and defines the line's cycle time (12 time units), whereas the second and third stations have idle times of 4 and 5 time units, respectively.

The line is said *balanced* if the sum of idle times of all stations along the line is as low as possible. Theoretically, it is possible to achieve *perfect balancing*, which occurs when tasks can be gathered so that all station total process times are identical. Figure 2.6 is an example of a perfectly balanced line of the previous suboptimal task distribution presented in Figure 2.5. However, in most practical cases, the perfect balancing is unattainable.

According to Baybars (1986), the assumptions that can be applied to *any* deterministic model for Assembly Line Balancing Problem (ALBP) are stated in the following simplification hypotheses (SH):

- (SH-1): all input parameters are deterministic and known;
- (SH-2): tasks are indivisible and cannot be split into two or more stations;

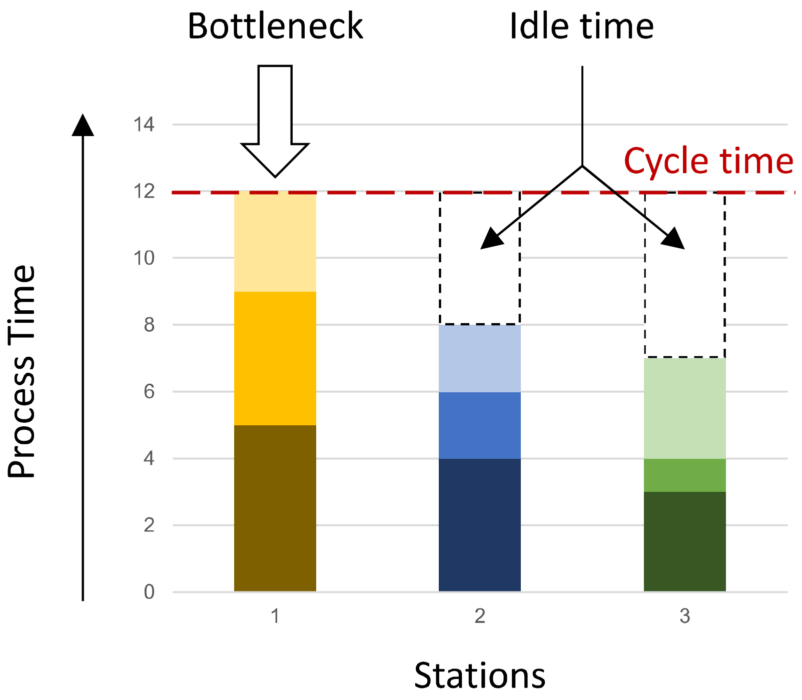

Figure 2.5: Example of an unbalanced production system: the cycle time is defined by the bottleneck station (12 time units), whereas stations 2 and 3 have idle time. Each task is represented by a different colour.

- (SH-3): tasks cannot be processed arbitrary due to technological precedence restrictions;
- (SH-4): all tasks must be processed.

A recent survey regarding ALBPs is done by Battaïa and Dolgui (2013).

2.3 Simple Assembly Line Balancing Problem

The ALBP has been studied by Salveson (1955) for the first time, introducing the problem. However, only after Jackson (1956) the task indivisibility and precedence constraints were solved along with the problem. Bowman (1960) proposed a consistent mixed-integer linear programming (MILP) formulation and it was further enhanced by White (1961).

Along with the mathematical programming models' development, SH (5-10) were added to the previous SH (1-4) that were stated in Section 2.2 to define a new class of

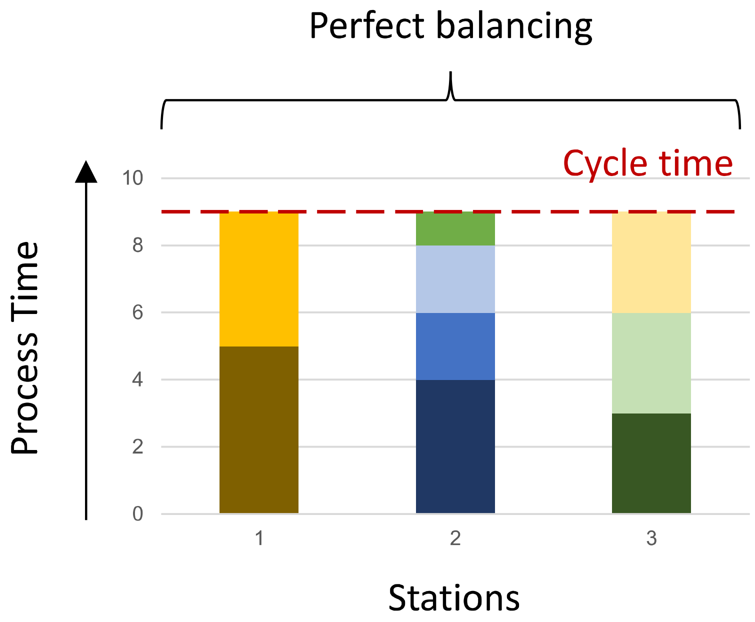

Figure 2.6: Example of a perfect balancing: all three stations have the same total process time, which defines the cycle time (9 time units) and guarantees a perfectly balanced system. Each task is represented by a different colour, notice the reallocation of tasks.

problem. More precisely, there is an addition of six SH in order to strictly define the Simple Assembly Line Balancing Problem (SALBP), as follow:

- (SH-5): all stations are equipped and manned to process any task;
- (SH-6): task process times are not sequence dependent;
- (SH-7): any task can be processed at any station;
- (SH-8): the assembly line is strictly serial;
- (SH-9): the assembly line is designed for a single product;
- (SH-10): the parameters are deterministic.

With this set of assumptions (SH-1)–(SH-10), most of the literature papers focus on achieving the best assignment of tasks among several stations arranged in a serial line, considering these many restricting assumptions. However, the developed research on

SALBP do not always describe and solve more realistic problems (Battaïa and Dolgui, 2013).

According to Baybars (1986), these differences in describing realistic features divide the classification in two categories and make the problems fall into one of the them: the Simple Assembly Line Balancing Problem (SALBP), and the General Assembly Line Balancing Problem (GALBP), further discussed in Section 2.4.

Furthermore, an extensive review on SALBP is done by Scholl and Becker (2006), in which four versions of the problem are described as they arise in the literature. Table 2.1 divides the SALBP versions according to their optimisation objective and given parameter.

Table 2.1: SALBP versions based on their optimisation objective and given parameter.

	Cycle time (CT)	
	Given	Minimise
Number of stations (NS)		
Given	SALBP-F	SALBP-2
Minimise	SALBP-1	SALBP-E

The simplest problem is SALBP-F, which only establish whether or not a feasible distribution can be achieved in a given combination of NS and CT.

In addition to (SH-1)–(SH-10), the optimisation problem in SALBP-1 receives another assumption regarding the line's CT (SH-11):

- (SH-11): the cycle time (CT) is given and fixed;

Therefore, the goal of the SALBP-1 variation is to minimise the number of stations along the line. Analogously, SALBP-2 has an additional assumption to fix NS as a parameter:

- (SH-12): the number of stations (NS) is given and fixed;

This SALBP-2 variation's goal is to minimise the cycle time or, in other words, maximise the production rate.

Lastly, SALBP-E is the most general problem version in which the line's efficiency is maximise by simultaneously minimising the CT and NS, generally considering their interrelationship in a non-linear approach ($CT \cdot NS$).

2.4 General Assembly Line Balancing Problem

As mentioned in Section 2.3, SALBPs do not describe as many practical features as those found in the General Assembly Line Balancing Problem (GALBP). In SALBPs, the system is basically restricted by precedence relations, task distribution, cycle time and/or number of stations constraints (SH-1)–(SH-12), whereas GALBPs regard further specifications, such as task incompatibility, space constraints, parallel station, multiple and capable workers, equipment selection, or unproductive time, among others, which implies in the relaxation of one or any combination of the previously stated simplification hypotheses. A particularly focused review on GALBP is done by Becker and Scholl (2006).

This section is intended to present a concise review on some applications of GALBPs that are related to the problem that is proposed by this master's thesis and, thereby, included in its features, as is summarised in Section 2.6. For instance, Rubinovitz and Bukchin (1991); Rubinovitz and Bukchin (1993) introduced the Robotic Assembly Line Balancing (RALB) problem, a method based on branch and bound algorithm that aimed to allocate robots into a (balanced) robotic assembly line. This model, however, does not analyse how workers (or robots, in the said case) behave in the production system when multi-manned stations, equipment selection, and parallel stations are considered. Applicable variations of the GALBP might be incorporated into the robotic models.

This section also lists and discusses which SH are being relaxed by the reviewed work since many of these ALBP extensions are explored individually. By all means, (SH-1)–(SH-4) are kept in GALBPs and so is (SH-5) for now. This last exception is later discussed in Section 2.5.

Task process times are not sequence dependent (SH-6), i.e. they are independent of the station at which they are performed and of the preceding or following tasks. The most common example of a practical extension that requires the relaxation of this hypothesis is the presence of set-up times: this can be seen as a preparation time needed between tasks (Scholl et al., 2008; Scholl et al., 2013) or a fixed unproductive time (dead time) that occurs before and/or after the product processing in each station (Bard, 1989) or between tasks (Lopes, Sikora, Molina, Schibelbain, Rodrigues and Magatão, 2017), generally related to the robots, workers or conveyors movement time. Moreover, the RALB problem introduced by Rubinovitz and Bukchin (1993) assumes different robots allocated in each station, creating a station dependence for task process times. This problem had its solution methods further studied in the past years (Kim and Park, 1995; Levitin et al., 2006; Daoud et al., 2014).

Any task can be processed at any station (SH-7), which means there are no positional, layout or zoning constraints. In practical cases, this hardly is true: tasks cannot always be assigned to any station (Deckro, 1989), some set of specific tasks frequently must be allocated either at the same station (inclusion constraints), or are incompatible and must be placed at different stations (exclusion constraints). Incompatibility or zoning restriction problems might happen due to position or accessibility problems (Essafi et al., 2010), fixed machinery and minimal/maximal distance between assignments (Scholl et al., 2010), among others (Scholl et al., 2010; Sikora, Lopes and Magatão, 2017).

The assembly line is strictly serial (SH-8) and, therefore, process times are additive at any station, since there are no feeder or parallel sub-assembly lines and any possible interaction of this type is neglected. The balancing of U-shaped lines is frequently approached in the literature. U-lines allow a different grouping of tasks in the precedence diagram, since workers might cross the line to perform tasks at the beginning and at the end of the line, providing better or at least equal results to serial lines (Miltenburg and Wijngaard, 1994). However, the U-line formulation still keeps the simplification of no movement time between the line's sides, an extension that was lately developed by Sikora, Lopes and Magatão (2017), considering any given layout with deterministic operator's movement times. Both Kim et al. (2000) and Guney and Ahiska (2014) consider a two-sided assembly line, the first one is dealt with using a genetic algorithms and the other is solved by mixed-integer programming and is applied on an automotive industry in order to decide the optimal automation level. Station paralleling variations also demand the relaxation of SH-8, and they are further discussed in Section 2.5.1.

The assembly line is designed for a single product (SH-9), mixed-model and multi-model features, balancing or sequencing problems are ignored. Thomopoulos (1967) firstly presented a formulation for serial lines in which multiple products were taken into account. Later, Miltenburg (2002) applied this relaxation on U-shaped lines and more recent studies approached the balancing problem together with the sequencing one (Sawik, 2012; Öztürk et al., 2015; Lopes, Sikora and Magatão, 2016). A further extension of the combined problem is allowing the model to decide the balancing, sequencing and the buffer allocation, Lopes, Sikora, Michels and Magatão (2017a) integrated the problems in a cyclical formulation.

Lastly, the parameters are deterministic (SH-10), no stochastic considerations are made. It means that task process times are determined parameters and do not vary at a probabilist distribution. Relaxing this hypothesis implies on a line balancing that does not guarantee a fixed optimal cycle time, but just secure it with a certain probability (Kottas and Lau, 1973), a cost-oriented approach has been adopted in the said case.

2.5 Assembly Line Design Problem

Thus far, we can declare that GALBPs are generalisations of SALBPs, since they consider relaxing one or any combination of the SALBP stated assumptions. However, GALBPs still omit some designing sub-problems, such as selecting the equipment depending on the set of candidate solutions for each manufacturing operation, costs of dimensioning the production area, assigning workers, and the layout itself. Whenever these fixed and variable costs associated with the production system (facilities, technology, operation) are taken into account, SH-5 (all stations are equipped and manned to process any task) is relaxed and we have a further generalisation of the GALBP (Baybars, 1986), namely Assembly Line Design Problem (ALDP).

Although the literature on ALBP is quite extensive, a gap between the practical and theoretical cases still exists, as may be noticed in the recent surveys (Becker and Scholl, 2006; Boysen et al., 2008; Battaïa and Dolgui, 2013). In real-world lines, flexible assembly systems are adopted and must be designed properly in order to reach the desired production rate at the minimum cost, which includes facilities, programmable robots and equipment selection. Therefore, the global decisions made by the company necessitate and depend on the optimal solution for the line layout that comes along with the balancing objective.

For industrial robotic lines, these designing decisions ought to be combined with cost minimisation procedures as to determine the production system layout. Amen (2000); Amen (2006) provide a survey on heuristic procedures, model formulation and methods to solve cost-oriented ALBPs. Moreover, such designing decisions must take into account the possibility of parallel stations (Section 2.5.1), selecting equipment (Section 2.5.2), assigning multiple workers (Section 2.5.3) and specific characteristics of the resistance spot welding task, which are studied in the problem (Section 2.5.4).

For exact methods, heuristics and meta-heuristics that deal with the ALDP, Rekiek et al. (2002) provide a comprehensive review.

2.5.1 Parallel Stations

When an additional identical station is placed in parallel with the original station in a serial line, the concept of parallel stations is introduced in the assembly line. In this situation, each station has its cycle time doubled for its operations. Therefore, another particularity applied on line design problem is the inclusion of parallelism as an option for better balancing solutions when there are tasks with longer duration times than the desirable cycle time or when the system's efficiency requires improvement.

Station and line paralleling extensions are approached by several authors. They research some potential advantages and drawbacks in paralleling, including equipment selection features, labour costs, positional constraints, and task assignment restrictions. Parallel stations also requires space, hence additional constraints should also be considered. Lusa (2008) supplies a survey of parallelism applications in ALBPs, summarising the state of the art for the combined problem.

Bard (1989) presents a dynamic programming algorithm which takes into account equipment and task costs, as well as unproductive time between cycles (dead time), which is mostly affiliated with transportation time between stations. The transportation time of work-pieces is attached to the movement of a conveyor or the time a robot takes to move a work-piece in (set-up) and out (tear-down) of a station, this usually is a fixed and unproductive handling time. In order to diminish these negative effects, Bard (1989) included the possibility of paralleling station in the formulation.

Askin and Zhou (1997) use a heuristic procedure for the parallel station problem combined with a mixed-model line. In addition, they proposed a parameter sensitivity analysis by using cost ratio principles that is also applied in the computational experiments conducted in this master's thesis in Section 5.1.

An extensive computational study regarding the effects of station paralleling on

ALBPs is developed by Ege et al. (2009). In this paper, several parameters' combination (number of tasks, task process times, different precedence diagrams, level of paralleling and cost ratios) compose datasets to be solved.

Tuncel and Topaloglu (2013) present a real-world electronic product assembly line, in which the line's rebalancing is a constant problem. Besides considering parallel station to increase the line efficiency, work-piece positioning and task incompatibly due to technological restrictions are also included in the problem, relaxing several SALBP hypotheses.

2.5.2 Equipment Selection

The equipment selection problem is strongly related to the station paralleling one. Whenever a station is paralleled, so is the equipment (labour, tools, machinery).

A weighted approach was conducted by Bukchin and Rubinovitz (2003) in order to integrate station paralleling and equipment selection into the assembly line design problem. They adapted an existent branch and bound algorithm previously used to solve equipment selection problem into a parallel station problem solving one. Furthermore, an integer linear programming formulation to solve the combined problem is developed. However, only an example problem is presented along with the optimal solution.

An ant algorithm was formulated to solve a new class of problem, which was named time and space constrained assembly line balancing problem, by (Bautista and Pereira, 2007). In this problem, the amount of space in the workstation is limited and different equipment must be allocated along the stations to perform each task.

Finally, Dolgui et al. (2012) present a mixed-integer programming model for a transfer line design problem, which is composed of several multi-spindle workstations. They classify their problem as an extension of the assembly line balancing and equipment selection combinatorial problem, seeking for the optimal design or reconfiguration of the studied lines. The model has been developed in order to solve more realistic industrial problems by minimising the line cost whilst satisfying technological restrictions.

2.5.3 Worker Assignment

The inclusion of any worker assignment formulation normally requires the relaxation of SH-5 and, therefore, such problems are powerfully related to ALDPs.

The first generalisation of assignment restrictions is proposed by (Scholl et al., 2010). In this paper, a generic resource is constrained and must be balanced, either process time, space, or operators should be summed as a finite resource.

For the problem proposed in this master's thesis, the most related one is the concept of multi-manned station, in which stations are occupied by multiple workers assigned to the same workstation simultaneously (Fattahi and Roshani, 2011; Yazgan et al., 2011). This feature is commonly applied in the assemblage of large vehicles, and the interference between workers and task scheduling in each station should be considered (Boysen et al., 2008). However, as is stated in Section 2.5.4, the resistance spot welding tasks generally do not have precedence relations between welding spots.

Other variations of the worker assignment problem are presented by Araújo et al. (2015) and Sikora, Lopes and Magatão (2017). The first paper models the inclusion of disabled workers in the assembly line. The other approaches travelling workers that might be able to perform tasks in more than one station, their movement time are measured and integrated into the worker's cycle time. Likewise, the inclusion of workers with different capabilities or tools may be studied.

2.5.4 Resistance Spot Welding Tasks

The ALBP literature is strongly related to the automotive industry, mainly to the final stage assembly. Figure 1.1, on page 14, illustrates how an automated welding line looks like. The arrangement of body shop stages also consist of assembly or manufacturing lines that can be treated by ALBP formulations. The body-in-white stage transforms sheets of metal into the vehicle's body by using resistance spot welding (RSW) procedures. This stage is composed of welding assembly lines that usually are highly automated (Michalos et al., 2010). Welding procedures might present several spot welding tasks with similar characteristics. Therefore, these similar tasks can be gathered together into a single task with a given number of copies of the same task, since their processing times are virtually identical (Sikora, Lopes, Schibelbain and Magatão, 2017). Moreover, there are three main categories in which the welding tasks found in the body shop fall: geometry, stud and finishing tasks. Geometry welding tasks assemble the metal sheet pieces together, stud welding tasks add screws on the metal sheets' surface, and finishing welding tasks are used to reinforce the vehicle's structure and, therefore, are executed after geometry welding tasks. Spot welding tasks do not present precedence relations between welding spots other than some accessibility constraints caused by geometry tasks assembling new metal sheets.

In a robotic welding assembly line, station paralleling possibilities might be included in the system in order to increase the line's efficiency, whereas equipment selection is primordial because geometry and finishing tasks can be performed with identical tools, but a different tool is required in order to execute stud tasks. Multiple robots working in the same station can be used to shorten the line's length.

2.6 Proposed Problem

In this master's thesis, we develop a mathematical model that minimises the designing cost of an assembly line considering several realistic aspects found in the automotive industry: (i) different robots, (ii) their space and accessibility constraints, (iii) equipment selection for each of them, (iv) task assignment restrictions (incompatibility and special precedence), (v) parallel stations, and (vi) unproductive time due to work-pieces movement between stations (dead time). The combination of these realistic aspects have the (SH-5)–(SH-8) relaxed and are not present in previous ALDP models, thus a new MILP model is herein proposed and solved.

Chapter 3

Problem Statement

This chapter presents the Robotic Assembly Line Design (RALD) problem definition. The elements considered in the problem are presented and explained, advantages of parallel stations are discussed and the body-in-white stage tasks' specificities are described. These practical characteristics are further included in the modelling formulation.

3.1 The Robotic Assembly Line Design Problem

The Robotic Assembly Line Design (RALD) problem is proposed and studied in this master's thesis. For this formulation, a robotic assembly line composed of an odd number of stations in an alternated platform-fixed and transporter robots pattern is considered, as illustrated in Figure 3.1.

The first transporter robot is on a conveyor (S1), the second station (S2) is a parallel platform station composed of two cells with four robots each (two welding robots and two stud robots). The third station (S3) is also paralleled and contains two cells of transporter robots. The forth station (S4) has only one cell and holds three robots (two welding robots and one stud robot). The fifth and last station (S5) is a single transporter cell without any task performing tool placed sideways.

The robots are assigned to platform or transporter stations composed of one cell (single stations) or two identical parallel cells (double stations). The transportation of work-pieces between the stations can be done manually by humans or can be automated

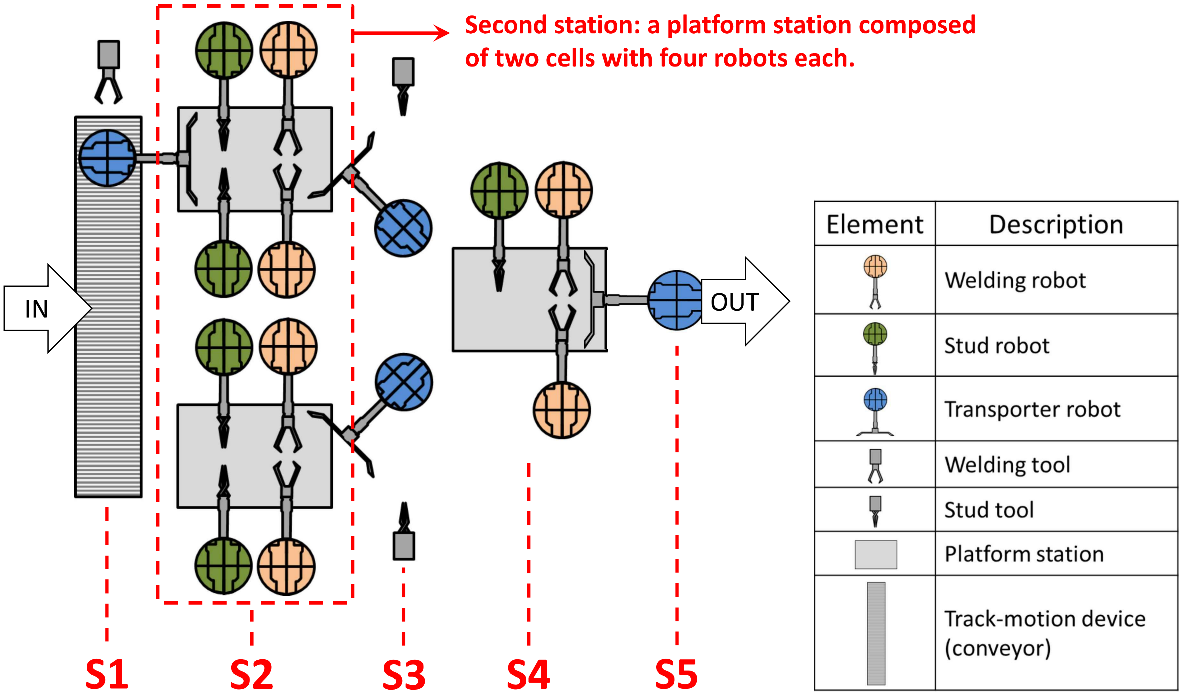

Figure 3.1: Example of elements in the proposed RALD problem. Parallelism in platform and transporter stations, multiple platform and transporter robots holding different tools or placed sideways, and transporter robots on conveyors.

by installing conveyors or manipulator robots: this last one is applied on the proposed problem. Commonly, these movement times (dead time) are neglected in the problem formulation, a simplification that contributes to unreliable solutions.

Platform robots holding welding tools are dedicated exclusively to perform assembly tasks, while transporter robots are mainly used for moving products in and out of the stations, but they also can use the remaining time of their cycle to perform tasks as long as they have a static tool placed on the line's sideways. These transporter robots that are capable of performing tasks are named capacitated transporter robots and are under the effect of an increased time penalisation on task time duration, since they take longer to perform the same tasks robots in platform stations do. This increment in the task time duration happens because, whereas in platform stations the work-pieces are steady for the robot to access the spot welding points, in transporter stations the robots have to manipulate the entire work-piece in order to make the points accessible to the tool.

Figure 3.2 is a picture of platform robots assembling a work-piece by performing welding tasks. The yellow welding guns (highlighted in the figure) are hold by the robots and the product stays fixed in the station. The movement between welding spots is reduced and task perform times as well.

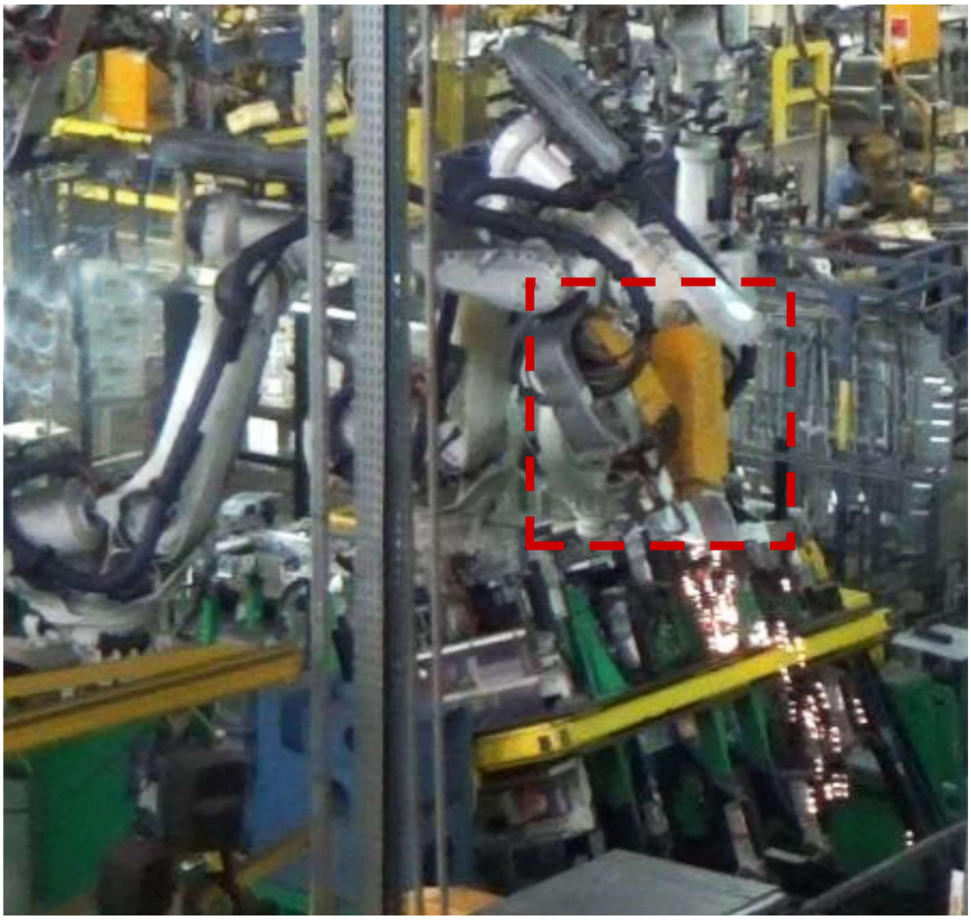

Figure 3.2: Platform robots performing welding tasks at the same time on the same work-piece. The yellow welding guns that each robot holds are highlighted in the picture.

Figure 3.3, on the other hand, is a picture of a transporter robot that uses a static welding tool. The yellow welding gun, in this case, is placed in the line's sideways and the robot moves the entire work-piece in order to make the welding spots accessible to the welding gun and perform the required tasks in a diminished rate.

Furthermore, transporter robots might be placed on track-motion devices in order to allow them to reach adjacent parallel stations and avoid unnecessary transporter station paralleling. However, transporter robots on track-motion devices suffer an additional penalisation on their useful time due to the extra conveyor's movement. Figure 3.1 depicts this alternated pattern between platform and transport stations, as well as equipment disposition and several other characteristics of the RALD problem that are further explained separately in this chapter.

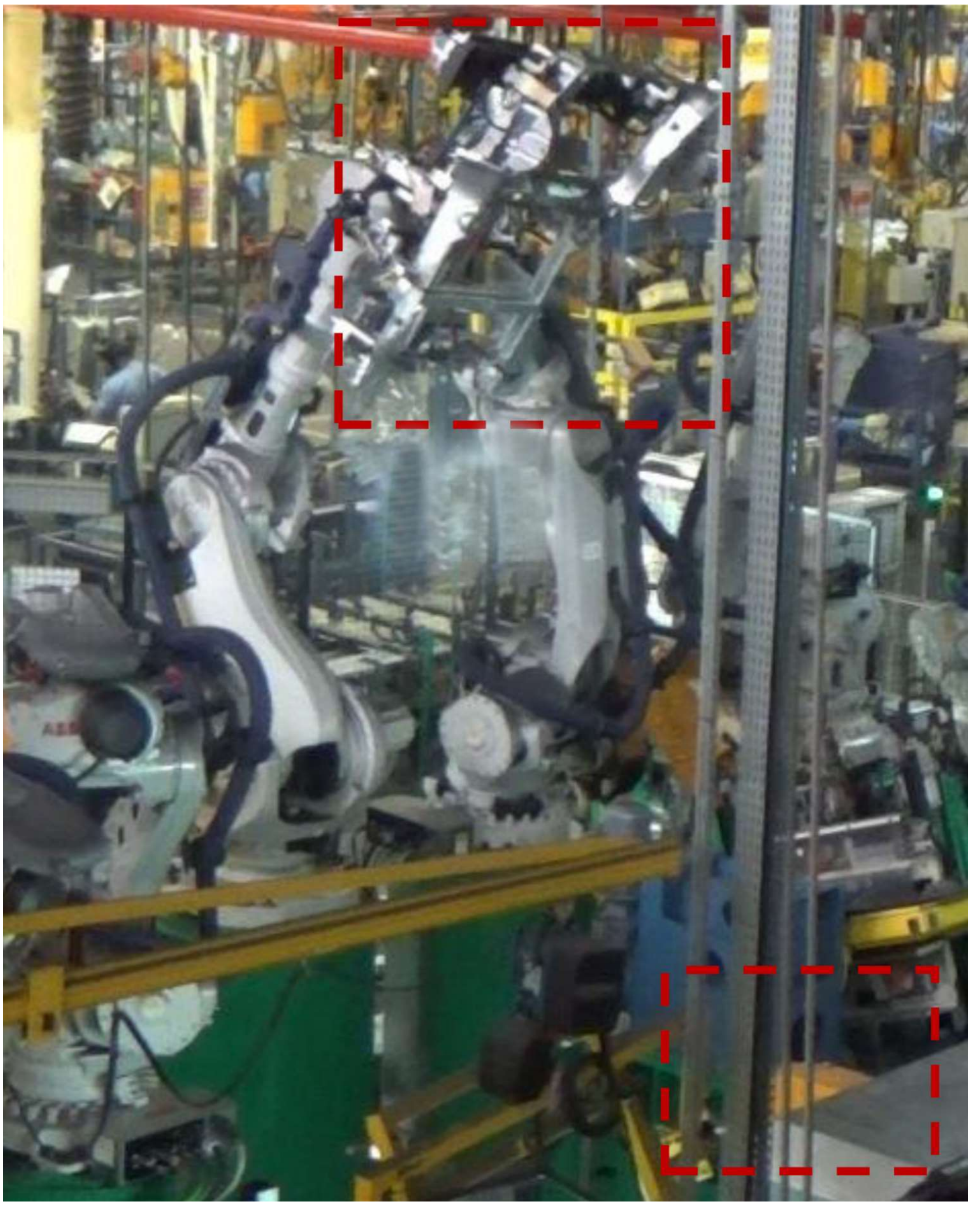

Figure 3.3: Transporter robot holding the entire work-piece and manipulating it to the welding gun placed on the line's sideways. The work-piece that the robot holds and the static yellow welding gun are highlighted in the picture.

3.2 Problem's Assumption

Firstly formulated by Rubinovitz and Bukchin (1991) and later dealt with a genetic algorithm (GA) by Levitin et al. (2006), the robotic assembly line balancing (RALB) problem is incorporated into the problem we have at hand. Some assumptions can be kept from their work since the balancing core characteristic for the single product problem is present in our formulation. The differences of the herein proposed formulation are highlighted in bold in the following assumptions:

1. The cycle time to meet the demand is known.
2. The assembly tasks precedence diagram is known.
3. There are no precedence relations within a station.
4. The duration of a task is deterministic and cannot be subdivided.
5. The duration of a task depends on **which robot and tool is assigned** to perform it.
6. **Parallel stations** are allowed.
7. For the general case, any task can be performed at any station when the precedence relations and **equipment requirements** are attended.
8. **Multiple robots** may be assigned to each station in line.
9. Transportation time for loading (set-up) and unloading (tear-down) are considered. Therefore, **dead time is considered** (Bard, 1989).
10. The goal is to **minimise design cost**, both robots and equipment have their prices as parameters.
11. Robots and equipment are available at any quantity.

These highlighted aspects distinguish the classical RALB problem from our RALD problem: lines are not strictly serial, i.e. either platform or transporter workstations can be doubled (see second and third stations in Figure 3.1) in order to improve efficiency at a reduced cost, unproductive transportation time is considered in the mathematical formulation, and multiple robots holding different tools are allowed at the same station (see second and forth stations in Figure 3.1).

3.3 Advantages of Parallel Station

Station paralleling might lead to some advantages over single lines. The cycle time increase at the doubled station is one of them, mainly when the dead time is considered, since the relative importance of movement times (set-up and tear-down) is reduced. This effect is depicted in Figure 3.4 and Figure 3.5. These figures show how parallel stations affect the line efficiency positively.

In Figure 3.4, two cycles of a serial station and one cycle of a parallel station is presented on a Gantt diagram. In the first case, the processing time (useful time) represents 50% of the cycle time (24 time units) and set-up and tear-down (dead time) the other 50%. When this same station is doubled, each copy of it benefits from paralleling and increases its useful time to 75% of the cycle time, an improvement of 50% in efficiency.

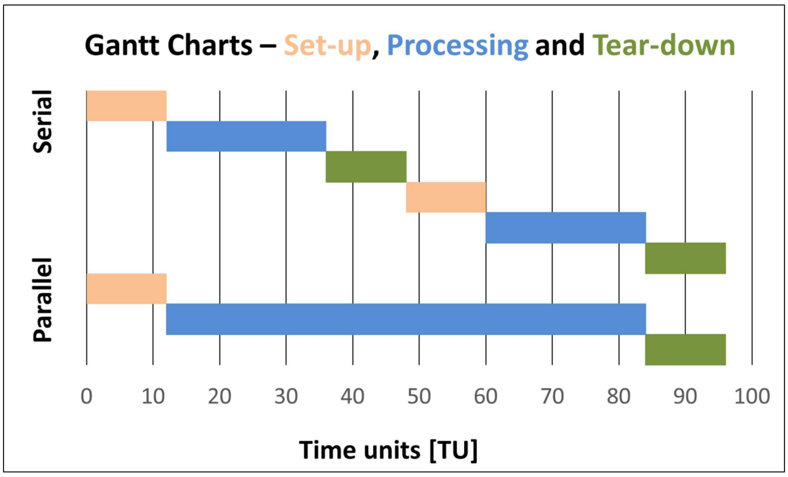

Figure 3.4: Gantt chart comparative between serial and parallel stations. The benefits of paralleling a station increase as the dead time represents a higher proportion of the cycle time: each cell of parallel stations can be 50% more efficient than a serial station for a CT composed of 12 time units of each set-up and tear-down, and 24 time units of processing/useful time.

In Figure 3.5, a schematic representation of two cells in a serial and parallel disposition is presented. Machines S1 and S2 only have 50% (24 time units represented by the larger block placed on its top) of their cycle time (48 time units represented by the sum of blocks placed around the machine) to perform task, whereas machines P1 and P2 have 75% (72 time units) of their cycle time (96 time units) dedicated for task performing activities, since there is more available time for that. Work-piece handling time (dead time) is the same for both configurations when loading (set-up) or unloading (tear-down) stations. However, the relative importance of this manipulation time is reduced because such work-piece handling process is conducted fewer times.

Moreover, secondary advantages of parallel stations are the improvement in productivity as a consequence of better balancing (Boysen et al., 2007) and the reduction on failure sensitivity, albeit the reduced production rate (Rekiek et al., 2002). These reasons make parallel stations a potentially profitable feature to be allowed into the line for prac-

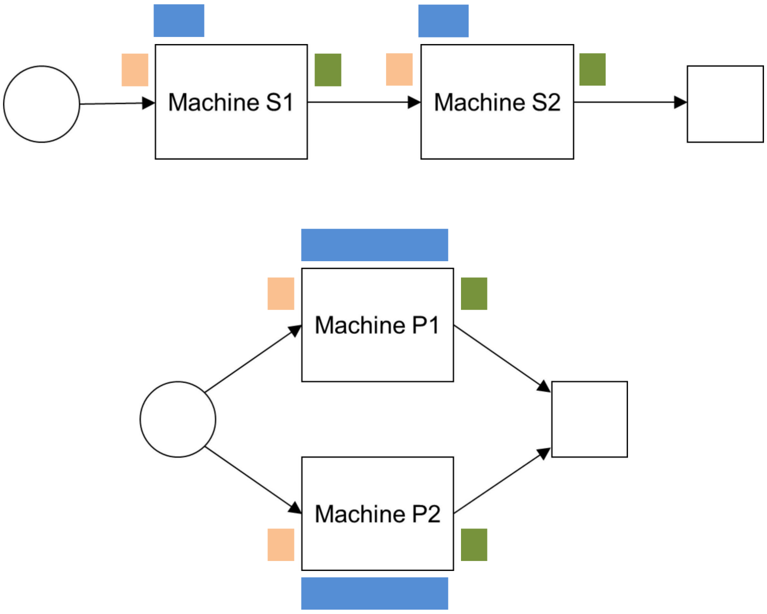

Figure 3.5: Schematic comparative between serial and parallel cells. The serial cells dedicate 50% of their cycle time to manipulate work-pieces (dead time), whereas parallel cells only use 25% for the same activity.

tical applications. Nonetheless, there are some drawbacks in the practical use: greater investment might be required for equipment and operation costs, making it necessary to calculate a trade-off and larger space is required for the stations, robots and equipment, which is incorporated into the model as a constraint limiting the number of robots per workstation.

3.4 Body-in-White Stage

Even disregarding the aforementioned drawbacks, paralleling stations indefinitely is not always possible depending on which tasks one is performing. For instance, the automotive manufacture widely uses the Resistance Spot Welding (RSW) technique in order to perform welding tasks in the body-in-white stage. This process unites metal sheets by using welding guns, which, in turn, require accessibility on both sides of the piece and, therefore, multiple types of this tool may be necessary as the welding steps proceed. In addition, metal sheet joining tasks must respect geometric tolerances, demanding actuators to bind the metal sheets to be united in the proper position. Such tasks are called **geometry** tasks. These geometry tasks must be performed at platform stations and the stations in which they are processed must not be paralleled for quality reasons.

Furthermore, the RSW technique is also used for reinforcement welding and screw adding points, namely **finishing** and **stud** tasks, respectively. Differently from geometry tasks, these ones do not require actuators to assure geometric tolerances and, theoretically, there are no precedence relations among any welding point. However, these spot welding points might become inaccessible after geometry tasks are performed, since the newly added sheets block the access of inner layers, creating station-wise accessibility windows that can be represented as a precedence diagram (see Figure 5.1 on page 58). Moreover, this welding assembly line property creates an incompatibility restriction between piece joining tasks (geometry tasks) and reinforcement tasks (finishing and stud tasks): all reinforcement tasks must be completed a station before any successor geometry task is allocated.

Consequently, this incompatibility restriction and the non-existence of precedence relations between two welding points enable the use of multiples robots not considering the task scheduling within a station by automotive industries. This is further discussed in Chapter 4.

3.5 Main Decision on a RALD Scenario

Therefore, after the presented considerations, the configuration of an entire robotic line can be defined, conceiving the Robotic Assembly Line Design (RALD) Problem: how many robots per station should be installed, how many tools of each type to use, which tasks are assigned to each robot considering equipment availability, and which stations would need to be paralleled in order to meet the demand at minimum cost. These decisions must all be made simultaneously to assure the optimal solution. The summary of elements in the optimisation model is shown in Figure 3.1 and are hereafter detailed as:

- **Parallelism possibilities:** both platform and transporter stations are allowed to be single or double stations;
- **Platform and transporter robots:** multiple robots per station are allowed, and both platform and transporter robots are capable of performing tasks as long as they have a tool assigned to them;

- **Equipment selection:** different types of tools may be assigned to both types of robots, even at the same station;
- **Conveyor possibility:** transporter robots might be placed on track-motion devices; this feature is required for single transporters adjacent to double platforms due to the size of the products to be assembled.

Ultimately, if one knows the productivity rate to meet the demand (desired cycle time of the line), the best solution will be the one that accomplishes this need and satisfies the precedence and space constraints at the lowest cost.

Chapter 4

Problem's Formulation

This chapter presents the Mixed-Integer Linear Programming (MILP) formulation for the Robotic Assembly Line Design (RALD) problem. Variables, parameters and sets are described and the mathematical model is presented and discussed in detail.

4.1 The Robotic Assembly Line Design Model

This chapter contains the Mixed-Integer Linear Programming (MILP) formulation for the Robotic Assembly Line Design (RALD) considering the problem definition and its characteristics described in Chapter 3.

4.1.1 Terminology

In order to ease the model's understanding, a concept based on the initial letter orientation will be employed for the parameters, sets, and variables definition as follows: all parameters and sets are written with an initial capital letter, an initial "b" indicates a binary variable (domain in $\{0, 1\}$) and an initial "v" indicates a non-negative continuous (domain in $\mathbb{R}_+$) or integer variable (domain in $\mathbb{Z}_+$).

Table 4.1 contains the applied terminology, describing the parameters and the sets used in the formulation. The parameters are empirically collected based on industrial conditions, such as available physical space and suppliers' prices. Note that the maximum number of stations (NS) must always be an odd number due to: (i) work-piece initial handling and final manipulation out of the line, and (ii) the transporter-platform sequential stations characteristic of the problem. Figure 3.1 on page 36 illustrates a line with an odd number of stations planned in an alternated pattern. The variables are detailed in Table 4.2, they are created by the model depending on the sets.

Table 4.1: Terminology: names of parameters and sets, their meaning, and [dimensional units].

Parameter	Meaning
NT	Number of tasks
NS	Maximum number of stations (an odd number)
$Nmax$	Maximum number of robots per cell
CT	Cycle time [time units]
DT	Dead time [time units]
TM	Time penalisation for track-motion [time units]
$D_{t,e}$	Duration [time units] of task t performed by equipment e
N_t	Number of copies of task t
$Prec$	List of precedence relations between two tasks t_i and t_j: (t_i, t_j)
$PCost$	Platform cost [$]
$TMCost$	Track-motion cost [$]
$RTCost_e$	Transporter robot cost [$] holding equipment e
$RPCost_e$	Platform robot cost [$] holding equipment e
SS	List of tasks that require single stations
Inc	List of incompatible tasks t_i and t_j: (t_i, t_j)
Set	**Meaning**
T	Set of tasks t
S	Set of stations s
St	Set of transporter stations s, odd stations in S
Sp	Set of platform stations s, even stations in S
TS	Set of feasible Task-Station elements
SE	Set of feasible Station-Equipment elements
TSE	Set of feasible Task-Station-Equipment elements

4.1.2 Objective Function

The problem's objective function is to minimise the purchase cost of the line. As it is described in Expression 4.1, the line's cost is composed of the cost of the platform and transporter robots along with their tools, platforms, and track-motions.

Table 4.2: Terminology: definition of the model's variables.

Variable	Set	Domain	Meaning
$vTd_{t,s,e}$	$(t,s,e) \in TSE$	$\mathbb{Z}_+$	Designation decision: set to the number of copies N_t of the task t assigned at station s using equipment e
$bTe_{t,s}$	$t \in T, s \in S$	$\{0,1\}$	Task ending: set to 1 if the task t is finished up to station s
$bTo_{t,s}$	$(t,s) \in TS$	$\{0,1\}$	Task occurrence: set to 1 if task t is performed at station s
bSo_s	$s \in S$	$\{0,1\}$	Station opened: set to 1 if station s needs to be used
bSd_s	$s \in S$	$\{0,1\}$	Station doubled: set to 1 if station s needs to be parallel
bTM_s	$s \in St$	$\{0,1\}$	Transporter station with track-motion: set to 1 if the robot in transporter station s is on a track-motion device
$vNR_{s,e}$	$(s,e) \in SE$	$\mathbb{Z}_+$	Number of robots per cell in the station s holding equipment e
$vTNR_{s,e}$	$(s,e) \in SE$	$\mathbb{Z}_+$	Total number of robots per station s holding equipment e
$vUT_{s,e}$	$(s,e) \in SE$	$\mathbb{R}_+$	Useful time at the station s using equipment e [time units]

$$\text{Minimise: } \underbrace{\sum_{\substack{(s,e)\in SE\\ s\in Sp}} vTNR_{s,e} \cdot RPCost_e}_{\text{platform robots' cost}} + \underbrace{\sum_{\substack{(s,e)\in SE\\ s\in St}} vTNR_{s,e} \cdot RTCost_e}_{\text{transporter robots' cost}} + $$

$$+ \underbrace{PCost \cdot \sum_{s\in Sp}(bSo_s + bSd_s)}_{\text{platform cost}} + \underbrace{TMCost \cdot \sum_{s\in St} bTM_s}_{\text{track-motion cost}} \tag{4.1}$$

4.1.3 Balancing Core

The model consists in balancing tasks among workstations. In each of the workstations, the number of robots, the paralleling, and the presence of track-motion devices delimit the available time for the operations. The inequalities for the balancing core of the model (Inequalities 4.2 to 4.5) are based on the formulation for the Integer SALBP, presented by Sikora, Lopes, Schibelbain and Magatão (2017) and applied on real-world instances from an automotive body welding assembly line.

Equation 4.2 is the occurrence restriction. It states that all N_t copies of task t must be assigned to stations. The precedence restriction is given by Inequality 4.3. Each task can only be assigned to a station (by variable vTd) if its predecessors have already been completed (measured by variable bTe) in or before a station s. The link between the vTd and bTe variables for the same task is given by Inequalities 4.4 and 4.5. Inequality 4.4 assures bTe can only assume 1 if all N_t copies of the task are already performed up to station s. Complementary, Inequality 4.5 forces $bTe = 1$ when the task is finished.

$$\sum_{(t,s,e)\in TSE} vTd_{t,s,e} = N_t \qquad \forall\ t\ \in\ T \tag{4.2}$$

$$bTe_{t_i,s} \cdot N_{t_j} \geq vTd_{t_j,s,e_j} \qquad \forall\ (t_i,t_j)\ \in\ Prec,\ (t_j,s,e_j)\ \in\ TSE \tag{4.3}$$

$$bTe_{t,s} \leq \sum_{\substack{(t,sa,e)\in TSE\\ sa\leq s}} \frac{vTd_{t,sa,e}}{N_t} \qquad \forall\ t\ \in\ T,\ s\ \in\ S \tag{4.4}$$

$$bTe_{t,s} + N_t - 1 \geq \sum_{\substack{(t,sa,e)\in TSE\\ sa\leq s}} vTd_{t,sa,e} \qquad \forall\ t\ \in\ T,\ s\ \in\ S \tag{4.5}$$

4.1.4 Paralleling and Problem's Linearisation

Bukchin and Rubinovitz (2003) state there is a diminishing return in paralleling stations. The first station in parallel improves the efficiency, whereas the contribution of additional parallel stations is quite small. Moreover, having many parallel station is only needed due to long task times, which do not happen in spot welding assembly lines (Sikora, Lopes, Schibelbain and Magatão, 2017). Besides, the cost of adding another copy of a robotic cell and the transporter accessibility to the stations would result in unaffordable or infeasible production layouts due to the size of vehicles. Therefore, we only consider possibilities of single or double stations.

The variable vUT is responsible for the measurement of the useful time used to perform tasks, and is calculated by summing the performed tasks (Equation 4.6). The available time depends on whether the station is open, single or double, and the number of robots. Inequality 4.7 presents a limit for the variable vUT based on the available time. The bold terms are variables. Note that the equation is not linear: the useful time depends on the product of three variables, posing a linearisation challenge.

$$vUT_{s,e} = \sum_{(t,s,e)\in TSE} vTd_{t,s,e} \cdot D_{t,e} \qquad \forall\ (s,e)\ \in\ SE \tag{4.6}$$

$$vUT_{s,e} \leq \boldsymbol{bSo}_s \cdot [(1 + \boldsymbol{bSd}_s) \cdot CT - DT] \cdot \boldsymbol{vNR}_{s,e} \qquad \forall\ (s,e)\ \in\ SE \tag{4.7}$$

This non-linear expression can be decomposed into the linear expressions 4.8, 4.9, and 4.11. Inequality 4.8 is used to determine whether the station is open: if bSo is 0, there is no useful time in the station; otherwise, the value $(2 \cdot CT - DT) \cdot Nmax$ is an upper bound for the useful time in a station. If the station is open and single, Inequality 4.9 is dominant. A non-doubled station ($bSd = 0$) results in restricting the useful time to $(CT - DT) \cdot vNR$. Inequality 4.10 is only applied to the transport stations (St), also considering a time penalisation for the use of track-motion devices. Once a robot using track-motion has to move between two workstations, there is less available time for the performance of tasks. Finally, Inequality 4.11 restricts the useful time when a station is doubled.

$$vUT_{s,e} \leq \boldsymbol{bSo}_s \cdot (2 \cdot CT - DT) \cdot Nmax \qquad \forall\ (s,e)\ \in\ SE \tag{4.8}$$

$$vUT_{s,e} \leq (CT - DT) \cdot \boldsymbol{vNR}_{s,e} + \boldsymbol{bSd}_s \cdot Nmax \cdot CT \qquad \forall\ (s,e)\ \in\ SE \tag{4.9}$$

$$\sum_{(s,e)\in SE} vUT_{s,e} \leq \sum_{(s,e)\in SE} (CT - DT) \cdot \boldsymbol{vNR}_{s,e} + \boldsymbol{bSd}_s \cdot Nmax \cdot CT - \boldsymbol{bTM}_s \cdot TM \qquad \forall\ s\ \in\ St \tag{4.10}$$

$$vUT_{s,e} \leq (2 \cdot CT - DT) \cdot \boldsymbol{vNR}_{s,e} \qquad \forall\ (s,e)\ \in\ SE \tag{4.11}$$

4.1.5 Multiple Robots

Due to space and accessibility constraints, the number of robots per cell (vNR) must be limited (Inequality 4.12). The total number of robots per station ($vTNR$) depends on whether the station is doubled. For example, in Figure 3.1, the second station contains four robots per cell ($vNR = 4$), however, as that station has been doubled, the total number of robots in the station is eight ($vTNR = 8$). Inequality 4.13 measures the number of robots for single stations, while Inequality 4.14 is active for double stations. Notice that this multi-robots aspect is only possible because there are no precedence relations within a station. Otherwise, task scheduling would be necessary in order to assure feasible answers. Nonetheless, if one sets the maximum number of robots per cell to one ($Nmax = 1$) in Inequality 4.12, the model is still valid for problems with a single robot per station. It is also important to notice that $vTNR$ is minimised in the objective function and, therefore, the variable $vTNR$ is set to receive the value of the number of robots per cell (vNR) depending on the parallelism applied to the station in Inequalities 4.13 and 4.14.

$$\sum_{(s,e)\in SE} vNR_{s,e} \leq Nmax \qquad \forall\ s\ \in\ Sp \tag{4.12}$$

$$vTNR_{s,e} \geq vNR_{s,e} \qquad \forall\ (s,e)\ \in\ SE \tag{4.13}$$

$$vTNR_{s,e} \geq 2 \cdot vNR_{s,e} - 2 \cdot Nmax \cdot (1 - bSd_s) \qquad \forall\ (s,e)\ \in\ SE \tag{4.14}$$

4.1.6 Work-pieces Flow

The next restrictions control the shape and flow of the line. Firstly, adjacent stations can only be opened if a previous one has already been opened (Inequality 4.15). Inequality 4.16 assures that a station can only be doubled if it is open. As the product flows across the line using transporter robots, it is necessary to have at least one of them at the starting point and after all platform stations. The transport cells are considered to contain only one robot. These restrictions are represented by Equality 4.17 for the first station and Equality 4.18 for the remainder stations. Furthermore, either when a work-piece has to be transported from a single transporter robot into a doubled station or vice-versa, the transporter robot requires to be on a track-motion device, unless this transporter station has also been paralleled (Inequalities 4.19 and 4.20).

$$bSo_s \leq bSo_{s-1} \qquad \forall\, s \in S \mid s > 1 \tag{4.15}$$

$$bSd_s \leq bSo_s \qquad \forall\, s \in S \tag{4.16}$$

$$\sum_{(s,e) \in SE} vNR_{s,e} = 1 \qquad \forall\, s \in S \mid s = 1 \tag{4.17}$$

$$\sum_{(s,e) \in SE} vNR_{s,e} = bSo_{s-1} \qquad \forall\, s \in St \mid s > 1 \tag{4.18}$$

$$bTM_s \geq (1 - bSd_s) + bSd_{s+1} - 1 \qquad \forall\, s \in St \mid s < NS \tag{4.19}$$

$$bTM_s \geq (1 - bSd_s) + bSd_{s-1} - 1 \qquad \forall\, s \in St \mid s > 1 \tag{4.20}$$

4.1.7 Practical Extension: Resistance Spot Welding Lines

Up to this point, the model is sufficient to describe the basic Robotic Assembly Line Design presented in Chapter 3. There are, however, some extra practical restrictions in the case study of Chapter 5 that require more expressions. As it is stated in Chapter 3, some tasks may require a single station, and some tasks cannot be performed in the same station.

The modelling of extra restrictions require an auxiliary variable (bTo) that controls whether any copy of task t is performed at station s. The link between bTo and the number of copies of tasks allocated to a station (vTd) is given by Inequalities 4.21 and 4.22.

$$bTo_{t,s} \geq \sum_{(t,s,e) \in TSE} \frac{vTd_{t,s,e}}{N_t} \qquad \forall\, (t,s) \in TS \tag{4.21}$$

$$bTo_{t,s} \leq \sum_{(t,s,e) \in TSE} vTd_{t,s,e} \qquad \forall\ (t,s)\ \in\ TS \tag{4.22}$$

Due to technological restrictions, geometry welding tasks are required to be performed on single platform stations, and all the precedent tasks must be completed one station before the geometry tasks start. Therefore, Inequality 4.23 is needed to assure that if a geometry task is performed in station s, the station cannot be doubled ($bSd = 0$). The necessity of finishing all precedence tasks before a geometry task can be modelled with a precedence relation (Inequality 4.3) added to the effect of an exclusion constraint due to incompatibility (Inequality 4.24).

$$1 - bTo_{t,s} \geq bSd_s \qquad \forall\ t\ \in\ SS,\ (t,s)\ \in\ TS \tag{4.23}$$

$$bTo_{t_i,s} + bTo_{t_j,s} \leq 1 \qquad \forall\ (t_i,t_j)\ \in\ Inc,\ (t_i,s)\ \in\ TS,\ (t_j,s)\ \in\ TS \tag{4.24}$$

Finally, the proposed MILP formulation for the studied practical RALD problem involves the objective function defined in Expression 4.1 and constraints from Equation 4.2 to Equation 4.6 and from Inequality 4.8 to Inequality 4.24. This MILP model is hereafter used for the computational and practical case studies presented in Chapter 5.

Chapter 5

Results and Discussion

> This chapter presents the computational and industrial study cases' results of the Robotic Assembly Line Design Problem described in Chapter 3 and formulated in Chapter 4. The parameters' influence over the optimal line design is discussed and the validated MILP model is applied to real-world data collected from three vehicle models.

Computational tests were performed in order to examine the influence of several model's parameters and validate the model due to expected outcomes. Two datasets were developed based on real-world data and the complete mathematical formulation, including extensions (Constraints 4.21 and 4.22) and extra restrictions (Constraints 4.23 and 4.24), was applied to them. The first dataset is composed of basic robots and tools, whilst the second one was elaborated with the same data in an enlarged equipment pool. These results are presented in Section 5.1.

Real-world data was collected from an automotive industry located on the outskirts of Curitiba-PR (Brazil) in order to analyse three practical case studies for different vehicle models produced in the company. Each vehicle model requires different amount of copies of each task and the duration of each copy might also be different depending on the vehicle model. The last model is the most complex one, presenting more tasks to be performed. The production rate to meet the demand is known and the assembly welding line ought to be designed aiming to achieve the desired cycle time at the lowest cost. The results are compared and discussed in Section 5.2.

To all instances, a 64 bit Intel™ i7 CPU (2.9 GHz) with 8 GB of RAM was employed using eight threads and the IBM ILOG CPLEX Optimization Studio 12.6. Optimal solutions were found for all instances in the first set (Section 5.1, Table 5.1) of the computational experiments and practical cases (Section 5.2, Table 5.5) within 3600 seconds, not exceeding the solving time limit, but for just 18 out of 32 instances in the enlarged set (Section 5.1, Table 5.2).

5.1 Parameters' Influence Computational Study

As the relative importance of the dead time (DT) in regard of the cycle time (CT) increases, it is expected that the line design will converge towards solutions with more parallel stations, so as to reduce the negative effects of unproductive movements and product loading. To observe this behaviour, computational tests were performed varying the DT from 0 to 70% of the CT. Larger values of DT were neglected for functional reasons: no line would operate with such inefficiency and CPU processing time is much higher when the number of maximum stations is increased. In addition, consequences of this fluctuation on DT can be detected on the number of robots, use of track-motions, and the final cost.

Another computational experiment has been conducted in order to analyse the effects of changing cost structures (Askin and Zhou, 1997), i.e. setting the cost ratio between the robot cost (R) and equipment cost (E) to the practical rate (approximately $R/E = 2$), $R/E = 1$ (Equal: robots and equipment have comparable costs), $R/E = 30$ (High: robots are much more expensive than equipment), and $R/E = 1/30$ (Low: robots are much cheaper than equipment). Moreover, tools that are able to perform the same tasks in a reduced time are included in the equipment pool in order to evaluate computational complexity and a possible trade-off. Faster robots and tools combination are capable of executing the same tasks slower robots and tools do in 60% of the time, and cost twice as much. For instance, if a welding robot that performs a copy of task t in 10 time units and costs 10 monetary units (\$) is considered, an additional welding robot that performs the same copy of task t in 6 time units and costs 20 monetary units (\$) is also considered in the enlarged set.

Thus, the combination of both experiments (DT and R/E variations) resulted in an amount of 64 instances that were summarised in Table 5.1 and Table 5.2, containing the total number of robots in system ($\#vTNR$), the number of opened and doubled stations ($\#bSo$ and $\#bSd$), the number of robots on track-motion devices ($\#bTM$), and the computational time in seconds.

Fixed parameters were defined based on practical characteristics of robotic welding assembly lines found in automotive industries. The desired CT is set to 1000 time units, the DT ranged from 0 to 70% of it, the number of maximum stations (NS) is gradually increased depending on the DT proportion and varies from 13 to 19, the necessary time to use the track-motion to 10% of the CT, the number of tasks is set to 40 (13 geometry tasks, 4 stud tasks and 23 finishing tasks), the number of copies of tasks varies from 1 to 20 replicas, the duration time of each copy ranges from 21 to 77 time units, and this value is increased by 50% if the task is performed at a transporter station.

Table 5.1: Results for different relative dead times (DT) and cost ratios (R/E) with a reduced equipment pool. $\#vTNR$, $\#bSo$, $\#bSd$ and $\#bTM$ stand for total number of robots in system, the number of opened, the number of doubled stations and the number of robots on track-motion devices, respectively.

DT (%)	Cost ratios (R/E): Practical | Equal | High | Low				
	$\#vTNR$	$\#bSo$	$\#bSd$	$\#bTM$	CPU Time (s)
0	22 | 22 | 22 | 22	11 | 11 | 11 | 11	1 | 1 | 1 | 1	0 | 0 | 0 | 0	29 | 20 | 23 | 28
10	25 | 25 | 25 | 26	13 | 13 | 13 | 13	0 | 0 | 0 | 0	0 | 0 | 0 | 0	52 | 39 | 76 | 67
20	28 | 26 | 26 | 28	13 | 13 | 13 | 13	0 | 2 | 3 | 1	0 | 4 | 3 | 2	21 | 43 | 22 | 22
30	30 | 29 | 29 | 32	13 | 13 | 13 | 15	4 | 3 | 5 | 3	0 | 4 | 2 | 6	39 | 71 | 127 | 82
40	32 | 33 | 32 | 33	13 | 15 | 13 | 15	4 | 3 | 5 | 3	3 | 6 | 2 | 6	13 | 31 | 41 | 22
50	36 | 36 | 36 | 37	15 | 15 | 15 | 15	9 | 4 | 6 | 3	0 | 5 | 1 | 6	55 | 183 | 729 | 387
60	39 | 39 | 39 | 43	15 | 15 | 15 | 15	7 | 7 | 8 | 3	2 | 2 | 1 | 6	36 | 30 | 33 | 30
70	46 | 48 | 46 | 51	19 | 19 | 19 | 19	9 | 5 | 9 | 3	1 | 4 | 1 | 6	18 | 18 | 27 | 22

Table 5.2: Results for different relative dead times (DT) and cost ratios (R/E) with an enlarged equipment pool. $\#vTNR$, $\#bSo$, $\#bSd$ and $\#bTM$ stand for total number of robots in system, the number of opened, the number of doubled stations and the number of robots on track-motion devices, respectively.

DT (%)	Cost ratios (R/E): Practical | Equal | High | Low				
	$\#vTNR$	$\#bSo$	$\#bSd$	$\#bTM$	CPU Time (s)
0	19 | 19 | 19 | 19	9 | 9 | 9 | 9	0 | 0 | 0 | 0	0 | 0 | 0 | 0	72 | 190 | 150 | 302
10	23 | 20 | 23 | 21	11 | 9 | 11 | 9	0 | 0 | 0 | 0	0 | 0 | 0 | 0	487 | 207 | 3600 | 450
20	26 | 26 | 26 | 28	11 | 13 | 13 | 13	1 | 2 | 3 | 1	0 | 4 | 3 | 2	3600 | 211 | 183 | 135
30	29 | 30 | 29 | 31	13 | 13 | 13 | 13	3 | 2 | 3 | 2	0 | 4 | 1 | 4	3600 | 2058 | 3600 | 129
40	32 | 33 | 32 | 33	13 | 15 | 13 | 15	4 | 3 | 5 | 3	3 | 6 | 2 | 6	3600 | 386 | 3600 | 220
50	36 | 35 | 36 | 35	15 | 15 | 15 | 15	9 | 3 | 7 | 3	0 | 6 | 1 | 6	3600 | 944 | 3600 | 1227
60	39 | 39 | 39 | 43	15 | 15 | 15 | 15	7 | 7 | 10 | 3	2 | 2 | 0 | 6	3600 | 3600 | 3600 | 518
70	44 | 45 | 41 | 43	17 | 17 | 15 | 15	8 | 5 | 8 | 3	2 | 4 | 1 | 6	3600 | 3600 | 3600 | 1899

Out of the 32 cases from Table 5.1, all of them were solved to optimality, whilst only 18 out of 32 cases from Table 5.2 would result in optimal solutions within the time limit, the enlarged equipment pool dataset is further discussed in Section 5.1.1. On average, the cost is increased in 9.92% whenever there is an increase of 10% in the DT, except for the pace from 60% to 70%. In this last case, the cost is impacted with a 16.69% raise and it clearly attests that such relative unproductive times would result in infeasible production systems.

The cost ratio experiment turned out as expected, validating the model for the practical cases: the line layout is completely changed depending on robot and equipment relative costs. For the cases in which the robots are much more expensive ($R/E = 30$), the line applies parallel stations more frequently in order to take advantage on productive time enlargement. On the other hand, the use of track-motion devices was more intensive for the opposite cases ($R/E = 1/30$), since the robots are much cheaper than the tools, the model decided to allocate the equipment mainly on platform stations, where there is no penalisation on task performing time.

The maximum number of stations (NS) for the herein reported tests were estimated based on the previous knowledge of the problem and the proposed dataset, this NS was generally higher than necessary, i.e. a pessimistic estimation. Observing the non-linear fluctuation on computational times in Tables 5.1 and 5.2, further analysis on the influence of NS on computational effort appeared to be required. Section 5.1.2 shows how setting NS to different values affects the computational processing time.

5.1.1 Computational Effort: Enlarged Equipment Pool Analysis

Comparing Table 5.1 with Table 5.2, it is possible to state that computational times were highly affected by allowing more equipment options in the pool. However, the increase on the DT does not influence the solving time directly, and the instances in which the robot cost was much lower than the equipment cost (Low R/E) expressed that this class of parameters presents a reduced computational time to be solved. Moreover, the potential gains in saving design costs were analysed based on the cases that reached optimality, i.e. the 18 out of 32 instances presented in Table 5.2. On average, a potential economy of 1.11% was obtained and the larger difference was found to be a 7.57% (Low R/E and 0%

of DT) cost reduction.

The optimal answer was proved for all the instances in Table 5.1. Still, for the enlarged equipment pool instances (Table 5.2), there was a gap between the best found answer (UB) and the best possible answer (LB) in 14 out of 32 cases. The gap for each instance is shown in Table 5.3. For the instances that did not prove optimality in one hour of computational processing time, the average gap was 7.97%.

Table 5.3: Gaps for different relative dead times (DT) and cost ratios (R/E) for the enlarged equipment pool instances.

DT (%)	Gap: $(UB - LB)/UB$			
	Practical R/E	Equal R/E	High R/E	Low R/E
0	0%	0%	0%	0%
10	0%	0%	5.06%	0%
20	1.87%	0%	0%	0%
30	1.37%	0%	9.90%	0%
40	1.66%	0%	5.00%	0%
50	11.95%	0%	20.54%	0%
60	2.93%	3.26%	8.71%	0%
70	10.86%	7.32%	21.18%	0%

5.1.2 Computational Effort: Maximum Number of Stations Analysis

In order to investigate the differences on computational times in each experiment set, an analysis on the maximum number of stations (NS) parameter was conducted. When the model is applied to these datasets, it is observed that the solution time is strongly dependent on the parameter NS. If NS is set as more stations than necessary for the optimal solution, the resulting problem has a greater search space and, consequently, tends to be harder to solve. Therefore, it might be reasonable to attribute the increased computational effort to a wrongly overestimated parameter.

The tendency of longer computational processing times is observed whenever the maximum number of stations (NS) parameter is unnecessarily set to higher values than the optimal number of opened stations ($\#bSo$) in the answer. For this analysis, the 32 first instances presented in Table 5.1 were solved with NS set to the sum of the opened stations ($\#bSo$) found in the optimal solution, to $\#bSo + 2$ and to $\#bSo - 2$. Hereafter, the optimal number of opened stations ($\#bSo$) is referred to as NS. The addition and subtraction of two units in the parameter related to the maximum number of stations is because it must be an odd number (see Section 4.1.1, on page 44).

As expected, the instances in which the maximum number of stations was set to $NS + 2$ took much more CPU time to be solved, varying from the increase of 224% (Practical R/E and 40% of DT) up to 5577% (Equal R/E and 20% of DT) in the worst case, or, in other words, the increase of 30.6 seconds up to 1149.4 seconds. It was also identified that the primal results (UB) in the instances with more NS than necessary are found as easily as in those with the optimal NS previously set, but the dual answer (LB) is more hardly raised. Therefore, it is possible to state that increasing NS more than required causes difficulties in finding better lower bounds.

On the other hand, setting NS to two station less than the optimal (number of opened stations) answer always resulted in infeasible solutions in a very reduced computational processing time. On average, the model took 1.3 second to prove the infeasibility, whereas the worst case to do so took 2.4 seconds. These results lead to an insight on how to estimate this parameter (NS) reasonably: it is possible to develop a heuristic that estimates the maximum number of stations optimistically, i.e. an estimation in which NS is always lower or equal to the optimal solution, and run the model increasing NS in two units (in order to keep the odd number characteristic) for each run until a feasible (and optimal) answer is found. The processing times are then summed up and this global processing time will be the CPU time. This approach only is attainable, however, because there is not an enlarged equipment pool and a trade-off between the tools does not occur. The proposed estimation application is not presented in this master's thesis, since the emphasis is on the new proposed problem and on its mathematical formulation. However, this heuristic method to define NS is listed as a future work direction (Section 6.2).

Lastly, the problem may be very hard depending on its parameters: mainly for the practical and high R/E, the problems were unsolved and presented a big gap (see Table 5.3), whilst the low R/E instances were solved in a reduced computational time. Another evidence to notice that was the unpredictable increase on computational time when two stations were added on the parameter NS (CPU time increase varying from 224% to 5577%). This indeterminable behaviour on processing times requests further investigation on the computational complexity of the RALD problem in order to classify it in one of the categories described by Garey and Johnson (1979).

5.2 Practical Case Study

Currently, three different vehicle models are being produced in the company. The validated model explored in Section 5.1 has been employed to the data of each vehicle model in order to conduct the practical tests. Figure 5.1 shows the adapted real-world automotive industry precedence diagram presented by Sikora, Lopes, Schibelbain and Magatão (2017), geometry and stud tasks are indicated in the diagram. Table 5.4 presents the task times for each vehicle model as long as they are performed in platform stations. These time durations are 50% longer if the task is performed in a transporter station. Geometry tasks are **bold-faced**, stud tasks are *italicised*, and the remaining ones are finishing tasks. Notice that less complex vehicles do not have all the assembly tasks that Model 3 does.

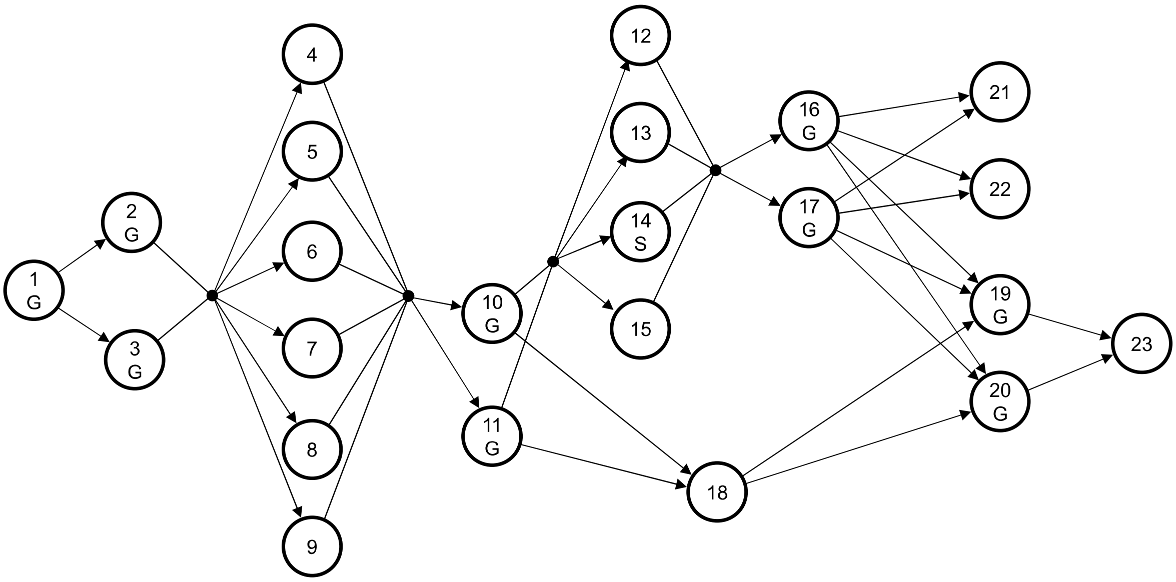

Figure 5.1: Precedence diagram for all vehicle models. Geometry (G) and Stud (S) tasks are indicated in the diagram. The remaining ones are finishing tasks. Adapted from (Sikora, Lopes, Schibelbain and Magatão, 2017)

As for the other parameters, the operating line had been observed and movement time analysed to properly determine DT and TM (respectively 50% and 10% of the CT for the practical case), and the desired CT has also been informed by the company (1168 time units). The maximum number of stations is empirically estimated by measuring the

Table 5.4: Task times in platform stations for each model. Geometry tasks are **boldfaced**, stud tasks are in *italics*, and the remaining ones are finishing tasks. Adapted from (Sikora, Lopes, Schibelbain and Magatão, 2017)

Task	Model 1		Model 2		Model 3	
	Copies	Duration	Copies	Duration	Copies	Duration
1	**8**	**57**	**8**	**57**	**8**	**57**
2	**6**	**38**	**6**	**38**	**6**	**38**
3	**6**	**50**	**6**	**50**	**6**	**50**
4	6	47	4	49	10	42
5	10	29	10	29	20	27
6	14	58	10	57	6	55
7	18	40	18	40	22	43
8	4	47	4	47	8	43
9	4	63	4	63	2	64
10	**15**	**63**	**15**	**63**	**15**	**63**
11	**13**	**39**	**13**	**39**	**13**	**39**
12	7	42	11	38	11	38
13	34	35	46	34	40	37
14	*21*	*70*	*18*	*71*	*37*	*77*
15	11	28	7	26	13	21
16	**15**	**69**	**15**	**69**	**16**	**71**
17	**5**	**44**	**5**	**44**	**8**	**45**
18	0	-	0	-	12	37
19	**20**	**34**	**6**	**35**	**18**	**32**
20	**0**	**-**	**12**	**50**	**4**	**52**
21	12	35	12	35	14	33
22	12	55	12	55	12	51
23	0	-	6	56	11	52

possible maximum length of the line and was set to $NS = 15$ for the practical case study. Industrial economic parameters have been collected, namely robot, equipment, track-motion, and platform costs. These are not always the same for any project, they often depend on numeric studies and labour cost for installing the line. The price parameters are average normalised values ($) taken from the last recent projects and updates: platform cost ($PCost = 4.2$), track-motion cost ($TMCost = 10.3$), transporter robot cost holding no equipment other than the work-piece manipulation system ($RTCost_e = 19.8$), with a static welding tool placed in the sideways ($RTCost_e = 29.9$), with a static stud tool placed in the sideways ($RTCost_e = 25.8$), and platform robot cost holding a welding tool ($RPCost_e = 20.7$), or a stud tool ($RPCost_e = 18.4$).

5.2.1 Line Design for Vehicle Models

Table 5.5 presents the results for the given parameters applied to each vehicle model. Naturally, the line cost is higher for Model 3, since it has more assembly tasks than the other vehicle models.

Table 5.5: Results for the three vehicle models produced by the company nowadays.

	Model 1	Model 2	Model 3
Cost ($)	557.5	561.5	628.6
$\#vTNR$	24	23	26
$\#bSo$	13	11	13
$\#bSd$	0	2	2
$\#bTM$	0	2	1
CPU Time (s)	40.1	31.2	32.7

The first and simplest vehicle model's layout configuration could be designed as an exclusively serial line in the optimal solution. Figure 5.2 shows the distribution of the robots and their tools' allocation through the stations.

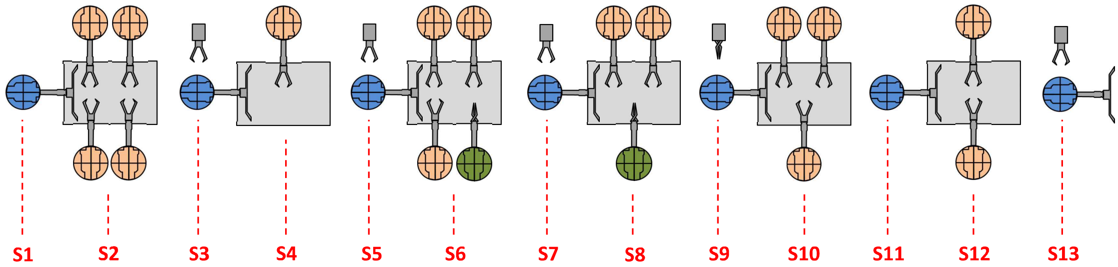

Figure 5.2: Optimal line design for Model 1. There are 13 serial stations (S1 to S13), no double stations or track-motion were employed on the configuration. There are 24 robots in total, composed of 17 platform robots (15 performing geometry and finishing welding tasks and 2 performing stud tasks) and 7 transporter robots (4 performing finishing welding tasks, 1 performing stud tasks (S9) and 2 for work-pieces handling, in the entrance and S11).

Analogously to the representation of the first model, Figure 5.3 depicts the optimal layout configuration for Model 2. In this case, station paralleling has been employed in order to reduce costs for the design project and has also shorten the line's length. Moreover, track-motion devices are used to reach and move work-pieces in and out of parallel stations.

The optimal line design of the last and most complex vehicle model is shown in Figure 5.4. In this configuration, the production layout requires 11 serial stations, 2 parallel stations and a track-motion device. The Model 3's line employs more features

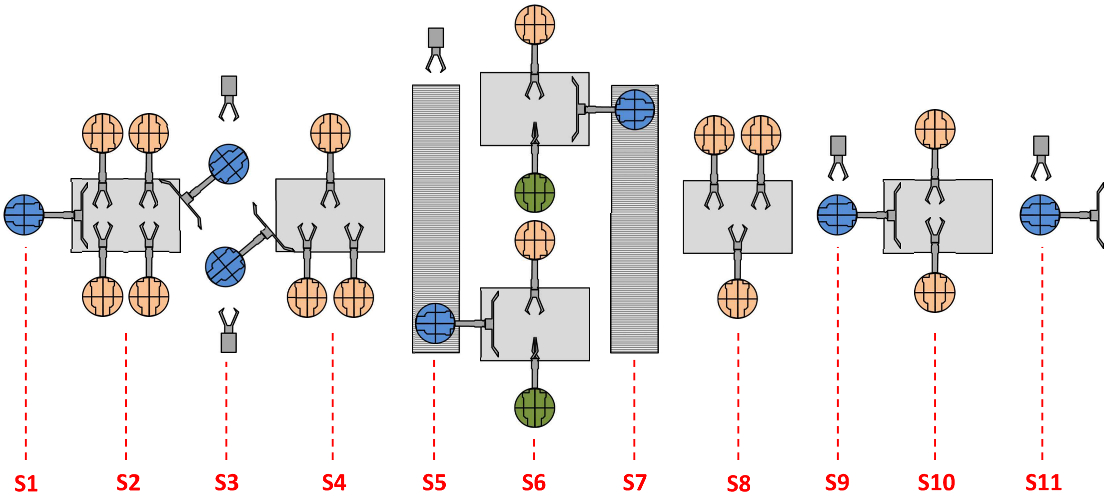

Figure 5.3: Optimal line design for Model 2. There are 11 stations (S1 to S11), 2 of them are doubled (S3 and S6) and 2 use a track-motion device (S5 and S7). There are 23 robots in total, composed of 16 platform robots (14 performing geometry and finishing welding tasks and 2 performing stud tasks) and 7 transporter robots (5 performing finishing welding tasks, none performing stud tasks and 2 for work-pieces handling, in the entrance and S7).

than the first one and is longer than Model 2's line. This fact is expected due to the larger amount of tasks, copies and longer task duration parameters.

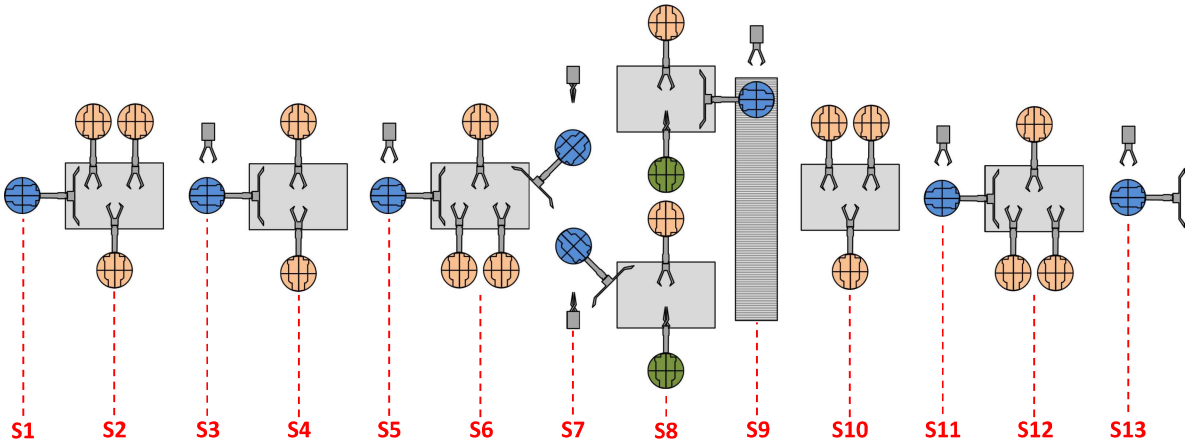

Figure 5.4: Optimal line design for Model 3. There are 13 stations (S1 to S13), 2 of them are doubled (S7 and S8) and 1 uses a track-motion device (S9). There are 26 robots in total, composed of 18 platform robots (16 performing geometry and finishing welding tasks and 2 performing stud tasks) and 8 transporter robots (5 performing finishing welding tasks, 2 performing stud tasks (S7) and 1 in the entrance for work-pieces handling).

5.2.2 Results Comparison

Figures 5.2, 5.3 and 5.4 represent robotic welding assembly lines for single products, as stated in the problem's hypotheses (Chapter 3). However, in the automotive industry, production systems are frequently built to process multiple models of vehicles, giving them the property of mixed-model assembly lines. Therefore, in order to analyse the applicability of any of these layouts, they must be feasible for all the vehicle models, otherwise, extra robots would have to be included in the faulty segments. Note that this approach can be seen as designing the line for the worst case. In this situation, the layout proposed by the mathematical model for vehicle Model 3 is the most probable candidate to assume such position and is a natural candidate to be tested for the remaining vehicle models.

The adopted procedure was setting the variables in the optimisation model for the last vehicle model's design, apply it to the data of vehicle models 1 and 2 and verify its feasibility for each case. The obtained results indicate that the configuration presented in Figure 5.4 was able to support the production of the three vehicle models and, thus, allowing the cost comparison with the current as-built line, presented in Figure 5.5. Alternatively, (i) some robots could be disable depending on the vehicle model that is to be processed in order to avoiding idle times or (ii) the cycle time for the less complex models could be even reduced in specific situations. Nonetheless, it is important to state and remind that the most costly design for a single product is not necessarily fit to produce all the products in a mixed-model assembly line due to the task distribution possibilities and idle times caused by relative demands of the products. A more general approach for mixed-model lines is a future research goal (Section 6.2).

Hence, the optimal solution for Model 3 has been compared to the current as-built design and proven coherent, reinforcing the reliability of the mathematical formulation, previously stated by the computational results of Section 5.1. Furthermore, Table 5.6 presents a comparative between the model's solution for the optimal line design, the configuration proposed by the engineering team and the strictly straight line for the Model 3. Similarly to the last procedure to test the Model 3's layout to Models 1 and 2, the strictly serial line was simulated for the vehicle Model 3 by setting decision variables (bSd) to the desired values (always equal to zero). In other words, parallel stations had been forbidden in the model and it was applied to the vehicle Model 3's data. A similarity between the optimal solution and the as-built configuration might be noticed in Table 5.6. However, if Figure 5.4 is compared to Figure 5.5, one can realise that the optimal design did not just reduce the amount of robots in the line, but also gave a different configuration

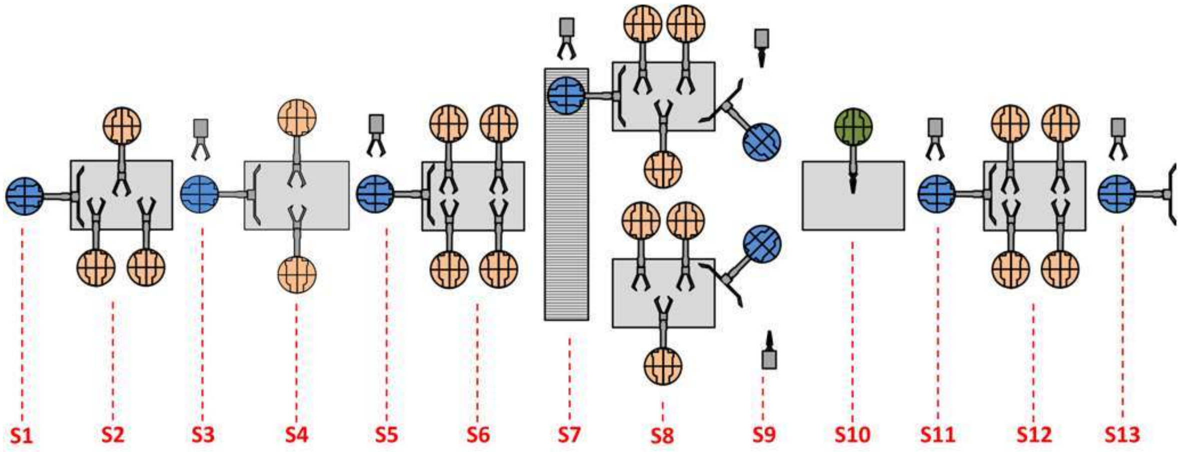

Figure 5.5: Current as-built line. There are 13 stations (S1 to S13), 2 of them are doubled (S8 and S9) and 1 uses a track-motion device (S7). There are 28 robots in total, composed of 20 platform robots (19 performing geometry and finishing welding tasks and 1 performing stud tasks) and 8 transporter robots (5 performing finishing welding tasks, 2 performing stud tasks (S9) and 1 in the entrance for work-pieces handling).

from the current operating line as solution.

Table 5.6: Comparative between the optimal line design, the configuration proposed by the engineering team (as-built), and the strictly straight line for Vehicle Model 3.

	Optimal	As-built	Serial
Cost ($)	628.6	667.7	692.4
$\#vTNR$	26	28	30
$\#bSo$	13	13	15
$\#bSd$	2	2	0
$\#bTM$	1	1	0

The costs on Table 5.6 are normalised due to industrial reasons and do not appear to be so large in absolute values. The obtained relative values indicate a potential economy of approximately 5.9%, when comparing the as-built with the obtained optimal solution. Nonetheless, taking into consideration the purchase cost of industrial welding robots and the potential of applying the model to all robotic lines found in an automotive industry, the cost reduction on the production layout can reach several hundred thousand dollars to be saved by the company.

Chapter 6

Conclusions

This chapter presents the conclusions and main contributions of the presented master's thesis. Furthermore, future research directions are provided.

6.1 Conclusions & Contributions

Providing the best solution to real-world problems has been the greatest interest of optimisation practical applications. Nonetheless, a gap between the academy research and industrial applications still exists in the literature, one of them was observed on robotic welding assembly lines, which are frequently found in the automotive industry: defining their production layout design is an important global and strategic decision. In this master's thesis, the Robotic Assembly Line Design (RALD) problem was defined and an MILP formulation was proposed for it, taking into account several practical considerations of a RALD scenario.

An extensive literature review based on publications and surveys definitions regarding the Assembly Line Balancing Problem (ALBP) was conducted in Chapter 2. This chapter covered the definition and uses of different production layouts (Section 2.1), more specifically flow-shop layouts, which are employed in automotive assembly lines and arises the ALBP (Section 2.2). The simplification hypotheses (SH) presented in Section 2.3 define the most explored problem in the literature, the Simple Assembly Line Balancing Problem (SALBP). Several variations of SALBPs were discussed and how the SH are relaxed in order to generalise the problem into a General Assembly Line Balancing Problem

(GALBP) (Section 2.4). A further generalisation was explored (Section 2.5) to define the Assembly Line Design Problem (ALDP) and to present variations that are applicable in the automotive industry, describing its distinctive characteristics.

> ***Contribution 1: The development of a literature review concerning the ALBP and its general variations to evidence the gap for an automotive industry assembly line design model.***

In Chapter 3, the RALD problem is described in detail and an overview of the optimisation elements is done. Section 3.2 states the problem's assumption and distinguishes the proposed problem from the previous Robotic Assembly Line Balancing (RALB) problem. Sections 3.3 and 3.4 described practical advantages of parallel station and welding tasks characteristics found in automotive industries.

> ***Contribution 2: The proposal and definition of the Robotic Assembly Line Design (RALD) problem considering real-world practical extensions.***

An MILP formulation to solve the proposed RALD problem was exposed in Chapter 4. The modelling challenges were due to an integer linearisation of a cubic constraint and the possibility to allow parallel stations in order to weight the costs and benefits of this feature in an exact manner (Section 4.1.4). Moreover, the general problem is added to extra practical restriction caused by the welding line characteristic (Section 4.1.7).

> ***Contribution 3: The development of an MILP formulation to solve the proposed problem and the modelling of extra restriction for resistance spot welding tasks.***

The model described in Chapter 4 was applied on computational tests and case studies in Chapter 5, in which the results were presented and discussed.

Computational case studies were performed in Section 5.1, combining large instances of real-world inspired cases adapted from Sikora, Lopes, Schibelbain and Magatão (2017) and cost ratio principles proposed by Askin and Zhou (1997). The existence of multiple tool alternatives and with the trade-off between equipment cost and efficiency led to higher computational difficulties. However, 18 out of 32 of such cases were solved to optimality within the time limit (Table 5.2, on page 54). The main conclusions drawn from this experiment are: (i) the model's inclination in paralleling more stations as the dead time increases or the robots are costly compared to the equipment, and (ii) the intense use of track-motion devices when equipment prices are much higher than the robot ones. A deeper analysis on the maximum number of station parameter (NS) was conducted in order to acquire an insight on the solution methodology.

> ***Contribution 4: The development of a validation dataset for the parameters' influence analysis and the model's computational results analyses themselves.***

Practical case studies based on the three vehicle models presented in Sikora, Lopes, Schibelbain and Magatão (2017) reached optimal answers and led to a 5.9% cost reduction in the line design for the most complex model compared to the originally human-designed line (Section 5.2). This was only possible because the third vehicle model line layout was able to assemble both vehicle models 1 and 2. Furthermore, parallel stations evidenced its essential role when unproductive times are considered, though paralleling was not necessarily cost-effective in every condition (e.g. Figure 5.2, on page 61).

> ***Contribution 5: The presentation of an automotive industry case study to apply the RALD model and the possibility to design robotic assembly lines at lower costs in the future.***

6.2 Future Research

This master's thesis exposed how effective the MILP formulation is when it comes to designing a robotic assembly line, including practical extensions that were able to shrink the search-space. Therefore, for future research, we plan to widen our model to incorporate literature variants that might be adapted to represent more realistic problems, being them:

- Multiple models: relaxing the SH-9 to project the design of a mixed-model line can lead to advantages in the final cost, once the line will not always be designed for the worst case, but for the combination of the multiple products demand.

- Task scheduling for each robot in the station: in order to guarantee the multiple robots working at the same station, a scheduling of their tasks must be modelled. Partial useful time to perform tasks might be summed up to weld an extra point, or geometry and finishing tasks assigned to the same station could require an accessibility order that is not considered in the model.

- Set-up time: including set-up times for different models would describe multi-model or mixed-model lines more precisely.

- Stochastic times: relaxing the SH-10 would remove the deterministic characteristic from the line, and if combined with the multiple model feature, the model would withdraw all the simplification hypotheses.

- External handling or feeder robot: it is common for robotic assembly lines to have external handling or feeder devices. The geometry task itself requires an auxiliary outsider robot to fix the work-piece in the proper place for the welding procedure. These details should be integrated into the model.

- Heuristic method to define NS: as shown in the computational results (Section 5.1), any NS lower than the optimal solution results in an infeasible problem for the basic equipment pool. An heuristic procedure to estimate the NS value in an optimistic manner can be developed. Then, the model would be executed whereas the parameter is increased by two units in each iteration.

- Solution method for the enlarged equipment pool: the computational time for problems with an enlarged equipment pool showed difficulties for the real-world based data. Therefore, a solution method for such cases ought to be developed in order to solve the practical cases in a reasonable time.

Bibliography

Amen, M. (2000). Heuristic methods for cost-oriented assembly line balancing: A survey, *International Journal of Production Economics* **68**: 1–14.

Amen, M. (2006). Cost-oriented assembly line balancing: Model formulations, solution difficulty, upper and lower bounds, *European Journal of Operational Research* **168**: 747–770.

Araújo, F. F. B., Costa, A. M. and Miralles, C. (2015). Balancing parallel assembly lines with disabled workers, *European Journal of Industrial Engineering* **9**(3): 344–365.

Askin, R. G. and Zhou, M. (1997). A parallel station heuristic for the mixed-model production line balancing problem, *International Journal of Production Research* **35**(11): 3095–3106.

Bard, J. F. (1989). Assembly line balancing with parallel workstations and dead time, *International Journal of Production Research* **27**(62015): 1005–1018.

Battaïa, O. and Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches, *International Journal of Production Economics* **142**(2): 259–277.

Bautista, J. and Pereira, J. (2007). Ant algorithms for a time and space constrained assembly line balancing problem, *European Journal of Operational Research* **177**: 2016–2032.

Baybars, I. (1986). A survey of exact algorithms for the simple line balancing problem, *Management Science* **32**: 909–932.

Becker, C. and Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing, *European Journal of Operational Research* **168**: 694–715.

Bowman, E. (1960). Assembly-line balancing by linear programming, *Operational Research* **8**: 2016–2032.

Boysen, N., Fliedner, M. and Scholl, A. (2007). A classification of assembly line balancing problems, *European Journal of Operational Research* **183**(2): 674–693.

Boysen, N., Fliedner, M. and Scholl, A. (2008). Assembly line balancing: Which model to use when?, *International Journal of Production Economics* **111**(2): 509–528.

Bukchin, J. and Rubinovitz, J. (2003). A weighted approach for assembly line design with station paralleling and equipment selection, *IIE Transactions* **35**: 73–85.

Daoud, S., Chehade, H., Yalaoui, F. and Lionel, A. (2014). Solving a robotic assembly line balancing problem using efficient hybrid mehtods, *Journal of Heuristics* **20**: 235–259.

Deckro, R. F. (1989). Balancing Cycle Time and Workstations, *IIE Transactions* **21**(2): 106–111.

Dolgui, A., Guschinsky, N. and Levin, G. (2012). Enhanced mixed integer programming model for a transfer line design problem, *Computers & Industrial Engineering* **62**(2): 570–578.

Ege, Y., Azizoglu, M. and Ozdemirel, N. E. (2009). Assembly line balancing with station paralleling, *Computers & Industrial Engineering* **57**(4): 1218–1225.

Essafi, M., Delorme, X., Dolgui, A. and Guschinskaya, O. (2010). A MIP approach for balancing transfer line with complex industrial constraints, *Computers & Industrial Engineering* **58**(3): 393–400.

Fattahi, P. and Roshani, A. (2011). A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem, *International Journal of Advanced Manufacturing Technology* **53**: 363–378.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1st edn, W. H. Freeman & Co., New York, NY, USA.

Guney, Y. and Ahiska, S. S. (2014). Automation Level Optimization for an Automobile Assembly Line, *Proceedings of the 2014 Industrial and Systems Engineering Research Conference*, pp. 2209–2218.

Jackson, J. R. (1956). A computing procedure for a line balancing problem, *Management Science* **2**(3): 261–271.

Kim, H. and Park, S. (1995). Mixed model U-line balancing type-1 problem: A new approach, *International Journal of Production Research* **33**: 2311–2323.

Kim, Y. K., Kim, Y. and Kim, Y. J. (2000). Two-sided assembly line balancing: A genetic algorithm approach, *Production Planning & Control: The Management of Operations* **11**(1): 44–53.

Kottas, J. F. and Lau, H. S. (1973). Cost-oriented approach to stochastic line balancing, *IIE Transactions* **5**(2): 164–171.

Krajewski, L. J., Ritzman, L. P. and Malhotra, M. K. (2013). *Operations Management: Processes and Supply Chains*, 10th edn, Pearson.

Levitin, G., Rubinovitz, J. and Shnits, B. (2006). A genetic algorithm for robotic assembly line balancing, *European Journal of Operational Research* **168**(3): 811–825.

Lopes, T. C., Michels, A. S., Molina, R. G., Sikora, C. G. S. and Magatão, L. (2016). An MILP formulation for robotic assembly line design with parallel stations, dead time and equipment selection, *XLVIII Simpósio Brasileiro de Pesquisa Operacional*, Vitória - ES, pp. 3257–3258.

Lopes, T. C., Sikora, C. G. S. and Magatão, L. (2016). Buffer and Cyclical Product Sequence Aware Assembly Line Balancing Problem: Model and Steady-State Balancing Case Study, *Annals of the XLVIII SBPO*, Vitória - ES, Brazil, pp. 37–48.

Lopes, T. C., Sikora, C. G. S., Michels, A. S. and Magatão, L. (2017a). Balancing, Cyclical Scheduling and Buffer Allocation Mixed-Model Assembly Line Problem: Model and Case Studies, **in review**.

Lopes, T. C., Sikora, C. G. S., Michels, A. S. and Magatão, L. (2017b). Mixed-Model Assembly Lines Balancing with Given Buffers and Product Sequence: Model, Formulation Comparisons and Case Study, *Annals of Operations Research* **in review**.

Lopes, T. C., Sikora, C. G. S., Molina, R. G., Schibelbain, D., Rodrigues, L. C. d. A. and Magatão, L. (2017). Balancing a Robotic Welding Manufacturing Line: Problem, Model and Case Study, *European Journal of Operational Research* **in press**.

Lusa, A. (2008). A survey of the literature on the multiple or parallel assembly line balancing problem, *European Journal of Industrial Engineering* **2**(1): 50–72.

Michalos, G., Makris, S., Papakostas, N., Mourtzis, D. and Chryssolouris, G. (2010). Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach, *CIRP Journal of Manufacturing Science and Technology* **2**(2): 81–91.

Michels, A. S., Dias, F. A., Sikora, C. G. S. and Magatão, L. (2016). Using a LP Model for the Design Analysis of an Integrated Renewable Energy System, *XLVIII Simpósio Brasileiro de Pesquisa Operacional*, Vitória - ES, Brazil, pp. 1029–1040.

Michels, A. S., Lopes, T. C., Sikora, C. G. S. and Magatão, L. (2017). The Robotic Assembly Line Design (RALD) Problem: Model and case studies with practical extensions, **in review**.

Miltenburg, J. (2002). Balancing and Scheduling Mixed-Model U-Shaped Production Lines, *The International Journal of Flexible Manufacturing Systems* **14**: 119–151.

Miltenburg, J. and Wijngaard, J. (1994). U-line line balancing problem, *Management Science* **40**(10): 1378–1388.

Öztürk, C., Tunal, S., Hnich, B. and Örnek, A. (2015). Cyclic scheduling of flexible mixed model assembly lines with parallel stations, *Journal of Manufacturing Systems* **36**: 147–158.

Rekiek, B., Dolgui, A., Delchambre, A. and Bratcu, A. (2002). State of art of optimization methods for assembly line design, *Annual Reviews in Control* **26**: 163–174.

Rubinovitz, J. and Bukchin, J. (1991). Design and balancing of robotic assembly lines, *Proceedings of the fourth world conference on robotics research*, Pittsburgh, PA.

Rubinovitz, J. and Bukchin, J. (1993). RALB - A Heuristic Algorithm for Design and Balancing of Robotic Assembly Lines, *Annals of the CIRP* **42**(1): 497–500.

Salveson, M. E. (1955). The assembly line balancing problem, *The Journal of Industrial Engineering* **6**: 18–25.

Sawik, T. (2012). Batch versus cyclic scheduling of flexible flow shops by mixed-integer programming, *International Journal of Production Research* **50**(18): 5017–5034.

Scholl, A. (1999). *Balancing and sequencing assembly lines*, 2nd edn, ed. Physica, Heidelberg.

Scholl, A. and Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, *European Journal of Operational Research* **168**: 666–693.

Scholl, A., Boysen, N. and Fliedner, M. (2008). The sequence-dependent assembly line balancing problem, *Operations Research Spectrum* **30**: 579–609.

Scholl, A., Boysen, N. and Fliedner, M. (2013). The assembly line balancing and scheduling problem with sequence-dependent setup times: problem extension, model formulation and efficient heuristics, *OR Spectrum* (35): 291–320.

Scholl, A., Fliedner, M. and Boysen, N. (2010). ABSALOM: Balancing assembly lines with assignment restrictions, *European Journal of Operational Research* **200**(3): 688–701.

Sikora, C. G. S., Lopes, T. C. and Magatão, L. (2016). Implementing Changes in the Balancing of Assembly Lines: a Sequencing Problem Addressed by MILP, *XLVIII Simpósio de Brasileiro de Pesquisa Operacional*, Vitória - ES, Brazil, pp. 73–83.

Sikora, C. G. S., Lopes, T. C. and Magatão, L. (2017). Traveling worker assembly line (re)balancing problem: model, reduction techniques, and real case studies, *European Journal of Operational Research* **259**: 949–971.

Sikora, C. G. S., Lopes, T. C., Schibelbain, D. and Magatão, L. (2017). Integer Based Formulation for the Simple Assembly Line Balancing Problem with Multiple Identical Tasks, *Computers & Industrial Engineering* **104**: 134–144.

Thomopoulos, N. T. (1967). Line Balancing-Sequencing for Mixed-Model Assembly, *Management Science* **14**(2): 59–75.

Tuncel, G. and Topaloglu, S. (2013). Assembly line balancing with positional constraints, task assignment restrictions and station paralleling: A case in an electronics company, *Computers & Industrial Engineering* **64**(2): 602–609.

White, W. W. (1961). Comments on a Paper by Bowman, *Operations Research* **9**(2): 274–276.

Yazgan, H. R., Beypinar, I., Boran, S. and Ocak, C. (2011). A new algorithm and multi-response Taguchi method to solve line balancing problem in an automotive industry, *International Journal of Advanced Manufacturing Technology* **57**: 379–392.