TUM School of Management
Technische Universität München

Smart Factory in the Automotive Industry: Design of Novel Flexible Layouts and Data-Driven Sequencing for Traditional Assembly Lines

Andreas Michael Hottenrott

Vollständiger Abdruck der von der TUM School of Management der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)

genehmigten Dissertation.

Vorsitzender:

Prof. Dr. Rainer Kolisch

Prüfende der Dissertation:

1. Prof. Dr. Martin Grunow
2. Prof. Dr. Alena Otto,
   Universität Passau

Die Dissertation wurde am 30.12.2020 bei der Technischen Universität München eingereicht und durch die TUM School of Management am 15.10.2021 angenommen.

Acknowledgments

After five years, my PhD studies come to an end. My time at the Technical University of Munich was full of great experiences and joy. At this point, I would like to take the opportunity to thank all persons that supported me during this PhD project.

First of all, I would like to express my deepest gratitude to my supervisor Professor Martin Grunow. The intense paper discussions always led to great progress of my research. I am thankful for his encouragement and generous support during my PhD project. With his advice and ideas as well as his knowledge, he decisively contributed to the success of this research. Furthermore, I want to thank Professor Alena Otto and Professor Rainer Kolisch for being on the assessment committee of my thesis.

I owe a big thank you to Professor Maximilian Schiffer for his co-authorship in one of my research papers. His great expertise improved both the paper and my OR knowledge. Also, I want to thank him for bringing me into contact with Professor Gilbert Laporte and enabling my research stay at CIRRELT in Montréal, Canada.

Over the years, colleagues have become close friends. Thank you Bryndís, Verena, Phillip, Alexander, Sina, Paul, Jishna, Gabor, Daniel, Radu, Thiam, Alex, Frank, Mirko, Alexandre, Lena, and Florian. I am grateful for fruitful discussions, supportive teamwork, delicious birthday cakes, competitive curling tournaments, and fun free-time activities. A special thanks also to our secretary Monika. Her organizing abilities and invaluable support made my PhD life much easier. Moreover, I want to thank Professor Renzo Akkerman for being the mentor of my PhD project.

Also, I want to thank my colleagues and friends at CIRRELT in Montréal for their support during my research stay in Canada. Above all, a big thank you to Professor Gilbert Laporte for inviting me to Montréal and the helpful discussions.

Finally, I want to thank my family and friends for their endless support in all situations of my life.

Andreas Hottenrott
Munich, November 2020

Abstract

The Industry 4.0 revolution is both a major challenge and a great opportunity for automotive manufacturers. By transforming their final assembly into a smart factory, manufacturers can meet the demand for increasing vehicle heterogeneity arising from the diffusion of alternative drivetrain technologies. To remain competitive in a dynamic, uncertain market environment, an automotive smart factory has to achieve an optimal balance between efficiency, flexibility, and robustness. Data-driven advanced planning algorithms are a key enabler in such a smart factory. On top of that, major automotive players recently started to consider a precedent break with the concept of assembly line production, which has been the status quo in this industry for the past century. They envision novel flexible assembly layouts, in which automated guided vehicles transport bodyworks on individual routes between assembly stations. The greater flexibility in this reinvented final assembly allows to cope better with high levels of vehicle heterogeneity.

In this thesis, we study the design and configuration of flexible assembly layouts and compare them to conventional assembly lines. We find that flexible assembly layouts have efficiency advantages of up to 30% compared to assembly lines. These advantages come at the price of an increased work in progress and a greater complexity when planning and controlling operations. We show that flexible assembly layouts are especially beneficial when facing high vehicle heterogeneity or changing demand mixes, e.g., during ramp-ups.

Furthermore, we develop a data-driven robust sequencing approach for conventional assembly lines, targeted at improving sequence stability. In light of decreasing in-house production, stable supplier signals become of utmost importance for a reliable just-in-sequence part supply. We show that our robust sequencing approach outperforms best practice approaches from industry and literature regarding this objective.

This thesis aims at supporting automotive manufacturers in the vital transformation to a smart factory. We seek to build bridges between academic research and industrial practice. By providing quantifiable scientific evidence on future production design, the insights from this thesis constitute a valuable guidance for automotive practitioners. For academics, the presented problems raise challenging methodological questions that open new fields for scientific research.

Contents

List of figures

List of tables

1 Introduction

The Industry 4.0 revolution is reshaping various industries. Core element of this revolution is the *smart factory*, a cyber-physical production system in which intelligent, computer-based algorithms plan, control, and execute the physical operations on the shop floor. The smart factory is enabled by recent advances in key technolgies, such as robotics, big data processing, artificial intelligence, and the Internet of Things (IoT) (Olsen & Tomlin, 2020). By transforming their production facility into a smart factory, manufacturers aim to increase efficiency, flexibility, speed, and quality while reducing cost[1]. Even more intriguingly, they expect to alleviate the tensions between these traditionally contradicting performance targets.

Several distinguishing characteristics affect the smart factory transformation in the automotive industry. For years, automotive manufacturers have been facing increasing vehicle heterogeneity. The demand for customized vehicles rises continuously, especially in the premium market. Original equiment manufacturers (OEMs) have reacted to this trend by offering a large number of models, engines, and selectable options (Meyr, 2004; Pil & Holweg, 2004). Major OEMs produce more than 30 models, each with countless configuration possibilities. For the Audi A3 alone, 10^{38} theoretical configuration possiblities exist[2]. Recently, the diffusion of alternative drivetrain technologies in the product mix adds a new complexity to this vehicle heterogeneity. Battery-powered electric engines seem to be the prevalent technology in the future, yet internal combustion engines are still most common. In addition, OEMs develop hybrids and hydrogen-powered vehicles. All these technologies require significantly different assembly tasks, tools, and worker qualifications. For example, the battery assembly for an electric vehicle differs considerably from the assembly of an internal combustion engine, and different safety standards apply. Even though, the demand for electric vehicles is ramping up, the transition is slow and uncertain. A future market shift towards yet another technology, e.g., from battery-powered to hydrogen-powered, may occur. Therefore, OEMs have to

[1] https://www.mckinsey.de/~/media/McKinsey/Locations/Europe and Middle East/Deutschland/News/Presse/2017/2017-03-31/dcc_brochure_may_2017.pdf (published: 31/03/2017, retrieved: 09/12/2020)
[2] https://www.automotivelogistics.media/12070.article (published: 15/12/2014, retrieved: 09/12/2020)

plan carefully and prepare for a long phase in which vehicles with different drivetrain technologies are produced in parallel.

Another distinguishing characteristic of the automotive industry is the tough market environment. Increased environmental awareness and tight regulations put pressure on the internal combustion engine. Due to overcapacities and low margins, the industry is currently going through a phase of consolidation. News about mergers and cooperations are omnipresent[3,4]. At the same time, new players enter the market for electric vehicles, e.g., Tesla and Google[5]. Especially in Germany, there is fear that the local OEMs have missed a timely adaption and now face competitive disadvantages compared to the tech giants from the U.S and new players from China.

In order to offer a high degree of vehicle heterogeneity without loosing their competitive advantage, OEMs have outsourced the production of standard parts while only retaining competencies in key technologies, e.g., engines. The proportion of in-house production has declined for many years and reached levels of merely 30% for major OEMs[6]. However, since space is a notoriously scarce resource in an automotive plant, OEMs seek to avoid inventories. They rely on just-in-time (JIT) or even just-in-sequence (JIS) supply of the required parts instead. For this, a close alignment with the suppliers and robust planning approaches are crucial, because otherwise any disruption in material supply immediately impairs production at the OEM.

Up to now, improving efficiency has been the eminent objective of automotive OEMs. Recently, they realize that flexibility and robustness are as important to ensure future competitiveness, because both enable adapting to changing, uncertain market environments and to cope with disruptions. Thus, an automotive smart factory has to be designed to achieve the optimum for all three objectives. From an operations management point of view, developing intelligent, data-driven planning algorithms is the primary success factor in this smart factory. On top of that, major players, such as Audi and Volkswagen, consider to break with the precedent concept of assembly line production[7], which has been the status quo in the automotive industry for the past century. They envision new, innovative layouts, so called *flexible assembly layouts (FALs)*, to replace

[3] https://www.ft.com/content/92ff16ec-2162-11ea-92da-f0c92e957a96 (published: 18/12/2019, retrieved: 09/12/2020)

[4] https://www.handelsblatt.com/english/companies/autonomous-plans-vw-bmw-and-daimler-hold-talks-on-cooperation-in-self-driving-cars/23909322.html (published: 25/01/2019, retrieved: 09/12/2020)

[5] https://www.spiegel.de/international/business/will-tesla-and-google-kill-the-german-car-a-1293415.html (published: 04/11/2019, retrieved: 09/12/2020)

[6] https://www.manager-magazin.de/unternehmen/autoindustrie/die-groessten-autozulieferer-a-1108918.html (published: 22/08/2016, retrieved: 09/12/2020)

[7] https://www.audi-mediacenter.com/en/audi-techday-smart-factory-7076/modular-assembly-7078 (published: 17/11/2016, retrieved: 09/12/2020)

parts of the assembly line. In these layouts, automated guided vehicles (AGVs) transport bodyworks on individual routes between assembly stations[8]. The greater flexibility allows to cope better with highly heterogeneous vehicles compared to conventional line assembly layouts (LALs).

The research presented in this thesis aims at supporting OEMs in the transformation to a smart factory. We design and configure FALs for the assembly of heterogeneous vehicles, and we compare them to conventional LALs. Moreover, we develop a data-driven planning approach to increase the robustness of sequence planning for LALs. To embed our research questions into daily operations, we first detail the automotive manufacturing process in Section 1.1, and introduce the planning problems for the automotive assembly in Section 1.2.

1.1 Automotive manufacturing process

The automotive manufacturing process consists of four consecutive production stages: the *press shop*, the *body shop*, the *paint shop*, and the *final assembly* (cf. Figure 1.1). In the press shop, sheet metal parts are produced. In the body shop, these sheet metal parts are jointed together to form the vehicles' bodyworks, which are painted in the subsequent paint shop. The final assembly is the last production stage, where engine, seats, and all other components are installed.

Usually, the press shop is physically decoupled from the other stages, because the sheet metal parts are produced in batches for several shifts. The remaining stages operate in a mixed-model flow fashion. They are sequentially arranged, only separated by resequencing buffers. Especially in-between paint shop and final assembly, large resequencing buffers exist, e.g., automated storage and retrieval systems. While the processes in the press shop, body shop, and paint shop are highly automated and standardized, manual labor is still predominant in the final assembly. The complexity of the vehicles' heterogeneity mainly affects this last production stage, because most configuration options occur there. Thus, we focus our attention on the final assembly in this thesis.

1.1.1 Assembly in line assembly layouts

Traditionally, most OEMs operate LALs in their final assemblies. Due to the serial arrangement, the vehicles have to run through all stations in the same sequence and pace, and workers have to perform tasks in a predefined order. The line is usually

[8] https://www.bcg.com/de-de/publications/2018/flexible-cell-manufacturing-revolutionize-carmaking.aspx (published: 08/10/2018, retrieved: 09/12/2020)

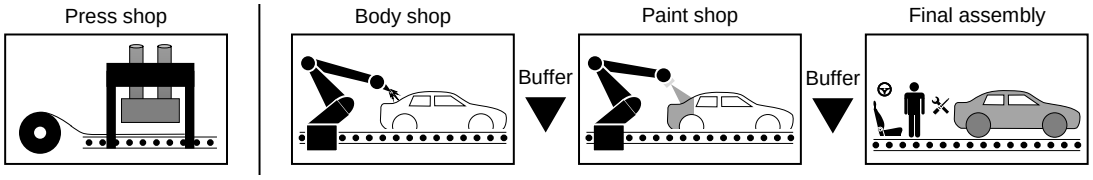


Figure 1.1: Production stages of the automotive manufacturing process.

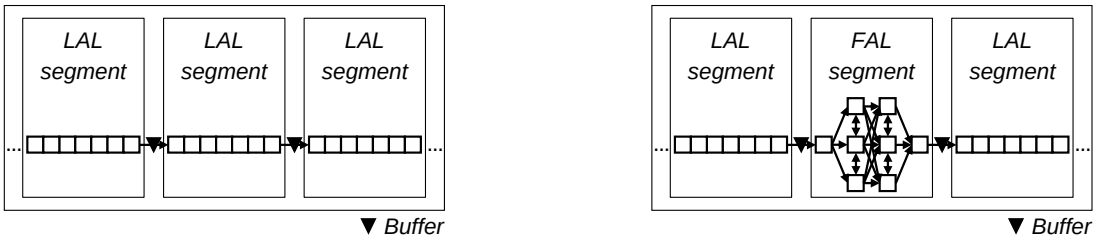
divided into multiple segments, which are decoupled by small buffers. A visualization of this status quo is given in Figure 1.2a. Since OEMs assemble multiple models in a mixed-model fashion, such LALs are also referred to as *mixed-model assembly lines (MMALs)*.

The uniform, paced workflow is both an advantage and a disadvantage of LALs. As an advantage, it allows for JIS stocking at stations and simplifies planning and control. The transportation effort is minimal, because stations are positioned right next to each other. The uniform workflow permits employing a highly efficient but inflexible transportation system, e.g., a conveyor. Consequently, LALs are best suited when producing homogeneous vehicles in high volumes. This is reflected in the famous quote by Henry Ford, the inventor of the automotive assembly line in 1913, who once said: "any customer can have a car painted any color that he wants so long as it is black"[9]. Indeed, Ford's Model T was highly standardized at that time, and therefore was a perfect fit for assembly line production. When producing heterogeneous vehicles, in contrast, the uniform, paced workflow turns into a disadvantage. Heterogeneous vehicles require different workloads at the stations which, in combination with the definite cycle time, causes work overloads at some stations and idle times at others. As an example, electric vehicles cause high workloads at the battery pack assembly station, whereas conventional vehicles have idle times. These imbalances result in a low efficiency and may deteriorate the output rate. OEMs exert a great deal of effort in alleviating these imbalances, e.g., by sequencing vehicles with alternating workloads consecutively, allowing workers to drift into subsequent stations, and allocating utility workers whenever work overloads occur.

1.1.2 Assembly in flexible assembly layouts

FALs break with the concept of a uniform, paced workflow. Here, the stations are neither arranged serially nor interconnected by a paced transportation system. OEMs employ AGVs instead. The AGVs fulfill two purposes. First, they transport the vehicles between stations, and second, they serve as assembly platforms. Due to the flexibility of

[9] Ford H. (1922). *My life and work.*



(a) Assembly in LALs. **(b)** Assembly with an FAL.

Figure 1.2: Alternative layouts for the automotive assembly.

the AGVs, each vehicle can take a unique route through the layout visiting only stations that are required for its assembly. All other stations are bypassed, e.g., a vehicle with an internal combustion engine bypasses the battery pack assembly station. Since FALs are not paced by a cycle time, the AGVs may enter and exit the stations at any point in time. For instance, a four-door vehicle spends more time at the door assembly station than a two-door vehicle. However, since the stations are not serially connected, transportation between stations requires time. Consequently, the transportation effort is higher than in an LAL.

In line with the smart factory vision, the AGVs and the stations are integrated in a digital network and continuously report their statuses to a central control that steers the movements of the AGVs in real time. This central control can exploit two types of flexibility. Tasks that do not require a fixed order allow for *operation flexibility*, i.e., modifying the task sequence (process plan) in real time. For example, the installations of sunroof and headlights are independent. In case the sunroof station is occupied, the vehicle can proceed with the headlights assembly first. *Routing flexibility* exists whenever the same task can be performed at multiple stations and the routing decision is made in real time. Because manual labor is predominant in the automotive assembly, the duplication of tasks to multiple stations requires limited investments.

LALs and FALs manifest a fundamental difference in the worker-to-vehicle relation. While workers have to wait for vehicles to arrive at their station in LALs, vehicles wait at stations for workers to become available in FALs. Consequently, OEMs expect that FALs achieve higher efficiency and output levels, especially when producing heterogeneous vehicles from changing demand mixes. The waiting vehicles, however, are likely to induce higher work in progress (WIP). Moreover, operation flexibility can cause worker confusion. Since the task sequence is not predefined, a worker might be confronted with different assembly states when performing the same task on different vehicles. Routing flexibility complicates material supply. JIS stocking is impossible, because it may only

be decided in real time at which station a task is performed. Instead, part kits have to be prepared in advance and are transported together with the vehicles on the AGVs. Another disadvantage of FALs is the increased complexity when planning and controlling operations.

In light of these advantages and disadvantages, OEMs seek to exploit the benefits of both layout types by combining FALs and LALs in the final assembly. They plan to replace the LAL by an FAL only in those segments in which the heterogeneity of the vehicles is particularly high, e.g., the assembly of the power train. In segments with low vehicle heterogeneity, e.g., the windshield assembly, the LAL is not altered. As an example, consider Figure 1.2b. The left and right segments are realized in an LAL, whereas the middle segment is realized in an FAL.

1.2 Planning problems for the automotive assembly

Planning for the automotive assembly is carried out in a hierarchical fashion. On the strategic level, the OEM decides on the layout design, whereas the assembly sequence of vehicles is optimized on the operational level.

1.2.1 Strategic layout design for the automotive assembly

Strategic layout design is a long-term decision problem with a planning horizon of several months or even years. Herein, the OEM decides on the number of stations, their positions on the shop floor, and the assignment of tasks to stations. OEMs seek to anticipate lower-level operations when optimizing the layout design, because any subsequent modification of the layout is expensive. Due to the long planning horizon, however, input data is unreliable, e.g., demand forecasts. Hence, OEMs aim for a flexible and robust layout.

OEMs differentiate between green-field and brown-field design. In green-field design, a new assembly layout is formed from scratch, whereas an existing layout is modified in brown-field design. Both scenarios are common for the automotive assembly.

1.2.2 Operational sequence planning for the automotive assembly

On the operational level, the OEM defines the assembly sequence of the vehicles produced in a shift. The goal is to balance the workload over time and to avoid work overloads. This can be achieved by sequencing vehicles with alternating workloads consecutively. Work overloads are costly, because they cause production delays and/or require employ-

ing utility workers. Moreover, the risk for quality defects rises when a worker faces high workloads over a period of time.

Sequence planning is also crucial for the alignment between an OEM and its suppliers. In order to enable a JIT or even JIS supply of the required parts, OEMs commit to a sequence several days prior to production. This sequence is forwarded to the suppliers (*supplier signal*) that produce the required parts and are responsible for a timely delivery to the final assembly. This supplier alignment is referred to as *pearl necklace concept* (Boysen, Scholl, & Wopperer, 2012; Meissner, 2010; Meyr, 2004), *pearl chain concept* (Wagner & Silveira-Camargos, 2012), *in-line vehicle sequencing* (Inman, 2003), or *stabilized production* (Müller, Lehmann, & Kuhn, 2020). The committed sequence is essential for efficient operations both at the OEM and its suppliers. Hence, sequence stability is of utmost importance.

1.3 Research objectives

The aim of this thesis is to design FALs for segments of the automotive assembly and to develop a robust sequencing approach for LAL segments. We contribute to the research on automotive smart factory concepts by developing data-driven optimization approaches for planning problems affected by vehicle heterogeneity. Although the problems investigated in this thesis are typical for the automotive assembly, they are also related to other industrial sectors in which heterogeneous products are manufactured, e.g., aerospace. Specifically, we study the following research questions:

RQ 1: How to design and configure FALs for the automotive assembly?
Inspired by the smart factory concept and pressured by increasing vehicle heterogeneity arising from the diffusion of alternative drivetrain technologies, major OEMs recently envision FALs for the automotive assembly. Since this is a novel layout concept, verified planning approaches neither exist in academia nor in industry. For a successful introduction of FALs, however, OEMs require reliable design and configuration approaches. Therefore, our first research question addresses the appropriate design and configuration of an FAL for a segment of the automotive assembly. We deduce two subquestions:

RQ 1.A: How to strategically design FALs for the automotive assembly? A fundamental question is obviously how to design an FAL, i.e., how many stations are required, where should the stations be placed on the shop floor, and which tasks should be assigned to which station. Because the application of FALs to the automotive assembly is novel, OEMs are most interested in a green-field design approach.

RQ 1.B: How to tactically configure FALs for the automotive assembly? On a tactical planning level, OEMs face a flexibility configuration problem in an FAL. The OEM has to decide on the exploitation of operation and routing flexibility as well as on an appropriate WIP target for the FAL. Hereby, the OEM has to find the optimal balance between a low WIP on the one hand and sufficient flexibility to deal with various unknown vehicle sequences on the operational level on the other hand. By exploiting operation and routing flexibility, the OEM can reduce the WIP without compromising operational performance. However, operation flexibility may cause worker confusion and routing flexibility complicates material supply.

RQ 2: What are the advantages and disadvantages of FALs compared to LALs? For which application scenarios are FALs superior to LALs?
FALs are vividly discussed in the automotive industry. However, it remains an open question to which extend FALs improve the efficiency of automotive manufacturing compared to conventional LALs. On the downside, FALs are expected to require a higher WIP. It is crucial that OEMs are aware of this trade-off and have quantifiable insights on it before deciding on the introduction of FALs in practice. Additionally, research on appropriate application scenarios is required. It is yet unknown which drivers affect the attractiveness of FALs. Automotive experts assume that FALs are especially beneficial when facing high vehicle heterogeneity and changing demand mixes, e.g., during ramp-ups. Ramp-ups become increasingly frequent due to shorter product life cycles, faster technological innovations, and continuous market launches of new models (Michalos, Makris, Papakostas, Mourtzis, & Chryssolouris, 2010). It is presumed that FALs enable smoother, faster, and cheaper ramp-ups than LALs. Nevertheless, quantitative research is required to confirm this hypothesis.

RQ 3: How to increase the robustness of sequence planning for conventional LALs?
Sequence planning is pivotal for OEMs that operate LALs or combinations of LALs and FALs, especially when producing heterogeneous vehicles. For a reliable supplier signal, stable sequences are important. In practice, though, sequence stability is non-satisfying (Inman, 2003; Meissner, 2010). Often, the committed sequence has to be changed in order to react to short-term disruptions (Lehmann & Kuhn, 2020; Müller et al., 2020). Quality problems or missing parts may delay the arrival of certain vehicles at the final assembly, such that they miss their scheduled sequence position. To maintain a high efficiency, the resulting gap is filled by bringing succeeding vehicles forward. These sequence alterations, however, may cause workload changes and potentially work

overloads at the assembly stations in LAL segments. As a remedial measure, additional sequence alterations are necessary, which further disturb material supply. Consequently, OEMs require a robust sequencing approach that anticipates disruptions based on past experience.

1.4 Thesis outline and contributions

This thesis is organized as a collection of three research papers that address the research questions outlined in Section 1.3. We propose data-driven optimization approaches for three planning problems on different hierarchical levels, i.e., at strategic design level in Chapter 2, at tactical configuration level in Chapter 3, and at operational sequencing level in Chapter 4. For each problem, the related literature is outlined in the respective chapter. From a methodological perspective, we focus on mixed-integer (non-)linear programming. We develop solution algorithms tailored to each individual problem, and we contribute to a wide range of algorithmic concepts, both exact and heuristic. More specifically, the organization and contributions of this thesis are as follows:

Chapter 2 discusses the strategic design of FALs. We investigate the optimal number of stations, their locations, and the assignment of tasks to stations. In addition, we compare the efficiency of FALs to LALs. The chapter thus targets research questions 1.A and 2. The contributions are fourfold. First, we provide a classification of the decision problems related to designing and operating FALs in the automotive assembly. Second, we provide a mathematical representation of the FAL design problem, which comprises an integrated station formation, station location, and flow allocation problem. Third, we develop an exact decomposition-based solution algorithm and an iterative fix-optimize matheuristic to solve problem instances. Fourth, we evaluate the effect of vehicle heterogeneity on the efficiency of FALs and LALs in our computational study.

Chapter 3 studies the tactical configuration of FALs. We quantify the inherent benefits of flexibility in FALs and compare worker utilization, output levels, and WIP to conventional LALs. Thus, this chapter addresses research questions 1.B and 2. There are five key contributions. First, we show analytically that operation and routing flexibility can have a significant impact on the operational performance of FALs. Second, we present a chance-constrained integer program that formalizes the flexibility configuration problem in FALs. Third, we show how this problem can be decomposed into deterministic subproblems, and we develop a branch-and-price (B&P) framework to solve the subproblems optimally. Fourth, we apply this framework to an extensive computational study in order to evaluate the impact of FALs in automotive manufacturing. Fifth,

we provide managerial insights on configuration options for different flexibility levers in FALs by quantifying their effect on operational performance. Moreover, we compare the performance of FALs to conventional LALs for both a stationary demand mix and the ramp-up of electric vehicles.

In Chapter 4, we answer research question 3 by developing an operational robust car-sequencing algorithm for conventional LALs. Our contributions are fourfold. First, we formulate the robust car-sequencing problem as a mixed-integer non-linear program. Second, we develop a branch-and-bound (B&B) algorithm that solves small-sized instances optimally, and we derive tailored lower bounds that significantly improve the algorithmic performance. Third, we propose a sampling-based adaptive large neighborhood search (ALNS) metaheuristic, which builds on observations we extract from optimal B&B solutions. Fourth, we solve the robust car-sequencing problem for a major European OEM, using extensive real-world data. We validate the superiority of our approach compared to the industry solution and an approach from literature.

Chapter 5 summarizes our findings with respect to the defined research questions, presents a synthesis, and gives an outlook.

1.5 Included publications

The research presented in this thesis is based on three different papers that all have been published in or submitted to selected A journals in the field of production management. Each of the following chapters is based on one of these papers. Accordingly, this thesis provides a comprehensive summary on designing FALs and on robust sequencing for conventional LALs.

Chapter 2: Hottenrott, A., & Grunow, M. (2019). Flexible layouts for the mixed-model assembly of heterogeneous vehicles. *OR Spectrum*, 41(4), 943-979. https://doi.org/10.1007/s00291-019-00556-x

Chapter 3: Hottenrott, A., Schiffer, M., & Grunow, M. (2020). IoT-driven manufacturing in the automotive industry: An impact assessment of flexible assembly layouts. *Submitted for publication.*

Chapter 4: Hottenrott, A., Waidner, L., & Grunow, M. (2020). Robust car sequencing for automotive assembly. *European Journal of Operational Research.* https://doi.org/10.1016/j.ejor.2020.10.004

2 Design of flexible assembly layouts for the automotive assembly

This chapter is based on an article published as:

Hottenrott, A., & Grunow, M. (2019). Flexible layouts for the mixed-model assembly of heterogeneous vehicles. *OR Spectrum*, 41(4), 943-979. https://doi.org/10.1007/s00291-019-00556-x

Abstract

The increasing vehicle heterogeneity is pushing the widespread MMAL to its limit. The paced, serial design is incapable of coping with the diversity in workloads and task requirements. As an alternative, the automotive industry has started to introduce FALs for segments of the assembly. In FALs, the stations are no longer arranged serially and no longer linked by a paced transportation system but by AGVs. This chapter investigates the initial design of such systems.

The FAL design problem (FALDP) is the problem of designing an FAL for a segment of the assembly of heterogeneous vehicles. It comprises an integrated station formation and station location problem. Moreover, the FALDP anticipates the operational flow allocation of the AGVs. We formalize the FALDP in a mixed-integer linear program (MILP) and develop a decomposition-based solution approach that can optimally solve small- to mid-sized instances. In addition, we transform this solution approach to a matheuristic that generates high-quality solutions in acceptable time for large-sized instances. We compare the efficiency of FALs to LALs and quantify the benefits of FALs which increase with vehicle heterogeneity.

2.1 Introduction

Inspired by recent technological advances in factory digitalization, automotive OEMs have started to investigate FALs as alternative to the widespread MMALs[10]. They have realized that the efficiency of conventional LALs deteriorates with high vehicle heterogeneity. Especially in the premium market, customers wish to configure their cars individually. Therefore, OEMs offer a large number of models, engines, and selectable options (Meyr, 2004; Pil & Holweg, 2004). This number is further increased by new technologies, like electric drives, which require very different assembly tasks, tools, and worker qualifications. One of the pioneers implementing FALs is Audi, which uses them in the assembly of the model R8 in Neckarsulm, Germany. By converting segments of the final assembly from LALs to FALs, Audi estimates efficiency gains of around twenty percent[11].

Assembling heterogeneous vehicles on paced, serial LALs is challenging. Heterogeneous vehicles require different workloads at the stations which, in combination with the definite cycle time, causes work overloads at some stations and idle times at others. For example, electric vehicles cause high workloads at the battery pack assembly station, whereas conventional vehicles have idle times. OEMs exert a great deal of effort in addressing these imbalances, e.g., by consecutively sequencing vehicles with alternating workloads, allowing workers to drift out of stations, and allocating utility workers whenever work overloads occur.

In alternative FALs, the stations are neither arranged serially nor linked by a paced transportation system. Instead, AGVs are used to transport the vehicles between the stations. The AGVs stay with the assigned vehicles throughout the assembly, because the AGVs are also used as assembly platforms at the stations. Stations that are not needed by a vehicle can be bypassed. For instance, a vehicle for a customer from a hot climate region can bypass the auxiliary heating assembly station. FALs are not paced by a cycle time, meaning that the AGVs can enter and exit the stations at any point in time. As an example, a four-door vehicle spends more time at the door assembly station than a two-door vehicle. In line with the Industry 4.0 and the smart factory vision, the AGVs and the stations are integrated in a digital network and continuously report their statuses to a central control that optimizes the movements of the AGVs in real time.

[10] https://www.audi-mediacenter.com/en/audi-techday-smart-factory-7076/modular-assembly-7078 (published: 17/11/2016, retrieved: 09/12/2020)

[11] https://www.handelsblatt.com/unternehmen/industrie/keine-fliessbaender-mehr-audi-plant-eine-revolution/14894190.html (published: 27/11/2016, retrieved: 09/12/2020)

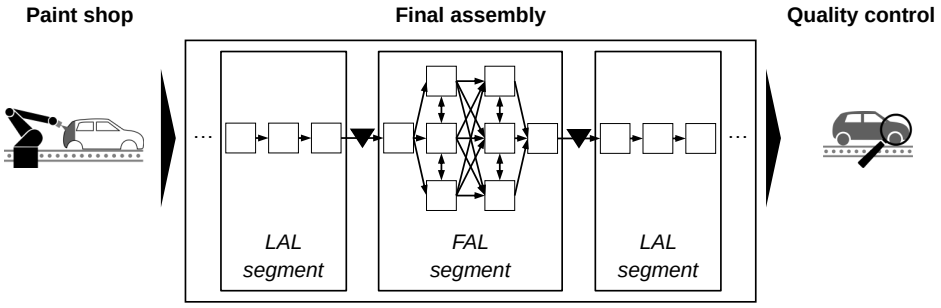


Figure 2.1: Integration of FALs in the automotive final assembly.

The control can exploit two types of flexibility, i.e., routing and operation flexibility (Browne, Dubois, Rathmill, Sethi, & Stecke, 1984). Routing flexibility is given whenever the same task can be performed at multiple stations. FALs are used for assembly segments in which manual labor is predominant. Since the duplication of manual tasks requires limited investments, assigning the same task to multiple stations is possible. Operation flexibility is given whenever there is no precedence relation between a pair of tasks. For example, it does not matter if the sunroof is installed before or after the headlights. In case the sunroof station is occupied, the vehicle can proceed with the headlights assembly first.

Given current vehicle architectures, combining LALs and FALs in the final assembly appears to be most beneficial. LALs are suitable for segments in which standardized, automated tasks needed by all vehicles are performed, whereas FALs are promising for segments in which highly variant, manual tasks are performed (cf. Figure 2.1).

Several decision problems need to be addressed when designing and operating FALs in the automotive assembly. Following the classification for LALs proposed by Boysen, Fliedner, and Scholl (2009), Figure 2.2 illustrates the hierarchy of these decision problems.

Starting point of this process is the segmentation of the final assembly into LAL and FAL segments, after which, the design of these segments as well as their intermittent buffers are determined. Initial design can be distinguished from reconfigurations. While initial design refers to the first design of new assembly systems, reconfigurations are adaptations of an existing assembly system during its lifetime. The design of the LAL segments involves the balancing, in which the stations are formed, and the design of the part feeding system. Two options exist for part feeding of LAL segments, i.e., stocking and kitting. For FAL segments, the design of the FAL, the design of the AGV system, and the design of the kitting system have to be created. In FAL design, tasks are assigned to stations (station formation) and the stations are arranged on the shop floor

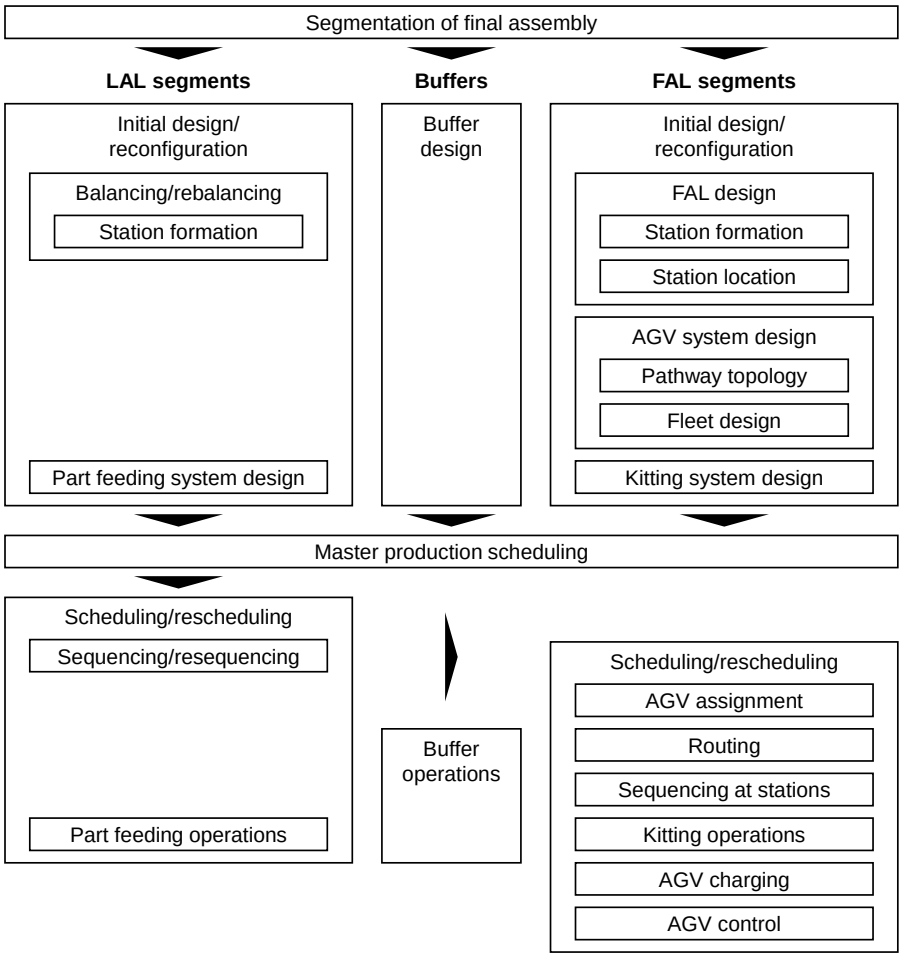


Figure 2.2: Hierarchy of decision problems when designing and operating FALs in the automotive assembly.

(station location). In AGV system design, the AGV pathways and the AGV fleet are determined. One challenge in designing FAL segments is part feeding. Limited station accessibility makes stocking difficult. Moreover, JIT or JIS stocking is impossible, because the task and station sequence for each vehicle may only be decided in real time. Therefore, kitting is used for part feeding. The kits of the required parts are prepared in advance and transported together with the vehicles on the AGVs. Because the movements of the AGVs are not continuous, buffers are needed within the FAL segments as well as at their entry and exit points. The design of these buffers involves decisions on their sizes and layouts. The buffers fulfill several objectives. They compensate for the stochasticity of the system and attenuate blocking and starving of the stations. The primary role of FAL segments is not to resequence the vehicles. Although FAL segments

could theoretically be used for resequencing, changing the vehicle sequence comes along with several challenges in practice. For example, doors are usually dismounted in the beginning of the final assembly and remounted in the end. If the vehicle sequence was changed, the door sequence would need to be changed accordingly. Also, a synchronized stocking of the LAL segments becomes challenging when the vehicle sequence is changed in the FAL segments. The outgoing sequence of an FAL segment needs to comply with the requirements of the succeeding LAL segment. The buffers at the exit points are used to reestablish the desired sequences. These sequences are not arbitrarily decided in real time, but planned in advance. Thus, simple FIFO buffers cannot be used, but more sophisticated and costly buffers that allow for resequencing are required.

Concerning the initial design, FALs have advantages and disadvantages compared to LALs. Because FALs are not paced, vehicles only occupy the stations while tasks are being performed. Wasting station capacity due to a smaller workload than the cycle time is avoided. Moreover, vehicles only visit stations that they need to visit. This improves the stations' utilization and therefore the efficiency. On the other hand, routing and operation flexibility lead to irregular flows that require more space and increase flow complexity.

Reconfiguring both LAL and FAL segments may become necessary during the lifetime of the assembly system in order to react to demand shifts, capacity changes, or new model introductions. FALs can more easily be reconfigured than LALs. Existing stations can be adjusted and new stations can be installed with limited effort and possibly even without suspending production. In contrast, in LALs, production usually needs to be suspended while the line is being rebalanced.

Master production scheduling is carried out on a mid-term planning level. It repetitively assigns vehicles to production periods, e.g., days or shifts. Because the LAL segments are characterized by lower flexibility than the FAL segments, master production scheduling is mainly restricted by the capacity of the stations in the LAL segments.

On a short-term planning level, vehicles are sequenced. Because of their high flexibility, FAL segments pose fewer limitations on the feasibility of sequences than LAL segments. Compared to pure LALs, the car-sequencing problem thus generally becomes easier. After the sequences are fixed, the part feeding operations for the stations in the LAL segments can be planned. The output of the LAL sequencing determines the arrival times and the required completion times of the vehicles in the FAL segments. The first step in scheduling of the FAL segments is to assign each vehicle to an AGV. Afterwards, the sequence of performed tasks and visited stations of each vehicle (routing) as well as the sequence of the vehicles at the stations are decided such that the arrival times and

required completion times are satisfied and the limited buffer capacities are respected. Moreover, the kitting operations are planned, AGV charging is scheduled, and the AGV flow is coordinated (AGV control). In parallel to assembly scheduling, buffer operations are planned. This involves planning the positioning, retrieval, and possibly resorting of the vehicles in the buffers.

FALs have advantages and disadvantages in terms of scheduling. The flows between the stations are neither paced nor coupled. Routing and operation flexibility allow a balanced distribution of the workload among the stations. One disadvantage is that routing and operation flexibility complicate AGV control. The higher flow complexity is, the more effort must be exerted in avoiding AGV collisions.

Rescheduling is required when unforeseen disruptions, such as tools becoming defective, occur. FALs allow vehicles to be rerouted to other stations. Conversely, in LALs, disruptions have more severe impacts and may even stop the entire line.

In this chapter, we investigate the initial design of FALs. More specifically, we look into the FAL design for a given segment of the automotive assembly (cf. Figure 2.2). We concentrate on the initial design as FALs are not yet common in the industry. We seek to answer two questions. First, we study the optimal design of FALs. We focus our analysis on the efficiency of FALs in order to demonstrate their capabilities. We implicitly consider space requirements and flow complexity by imposing restrictions on the AGV pathways and by minimizing the flow intensity. Second, we compare the efficiency of FALs to LALs. We study the effect of vehicle heterogeneity on the efficiency of both types of layout since this is the main driver that motivates OEMs to implement FALs.

Our contributions are fourfold. First, we propose a classification scheme for the decision problems connected to the design and operation of FALs in the automotive assembly as shown in Figure 2.2. Second, we provide a formal representation of the FALDP. The FALDP is the problem of designing an FAL for an assembly segment at an automotive OEM. It comprises an integrated station formation and station location problem. Moreover, the FALDP anticipates the operational AGV flow allocation. The flow allocation, however, is only used to evaluate the quality of the generated layouts. Specific characteristics of the FALDP are the consideration of routing and operation flexibility. The problem is modeled as a lexicographic multi-objective MILP. The objectives are to maximize efficiency and to minimize flow intensity. Third, we develop a solution approach to generate optimal layout designs. In our solution approach, we iteratively increase the number of stations (reduce the efficiency). For any fixed number of stations, we minimize the flow intensity. We show how this solution approach can be transformed to

a matheuristic to solve large-sized instances. Fourth, we conduct a performance comparison between FALs and LALs based on an adapted standard benchmark test bed. We show that FALs generally are more efficient than LALs. The gain in efficiency for FALs, however, is influenced by the heterogeneity of the vehicles. When producing homogeneous vehicles, the efficiency gain is low, whereas it increases with greater vehicle heterogeneity.

The remainder of this chapter is structured as follows: In Section 2.2, we review the related literature. In Section 2.3, we present an MILP for the FALDP. Our exact and matheuristic solution approaches are described in Section 2.4. In Section 2.5, we assess the performance of the solution approaches based on adapted instances from the literature. Also, we compare the efficiency of FALs to LALs and examine the effect of vehicle heterogeneity. In Section 2.6, we summarize our findings and discuss future research directions.

2.2 Literature review

The FALDP has not yet been addressed in the literature. In the following section, we review research papers that treat design problems related to the FALDP. The discussion is arranged by increasing flexibility of the underlying layout. We start our discussion with assembly line design, followed by row layout design, cellular manufacturing design, and finally job shop design. Due to the characteristics of the FALDP, we limit our review to papers that investigate static problems and that assume a discrete shop floor representation as well as equally-sized resources to be positioned.

Assembly line balancing is the problem of assigning tasks to stations in an LAL such that the precedence relations are satisfied and the limited capacity of the stations is respected. Assembly line balancing is related to the station formation problem of the FALDP. It has received considerable attention in the scientific literature. For comprehensive reviews, we refer to Becker and Scholl (2006) and the classification scheme of Boysen, Fliedner, and Scholl (2007). In the literature, mixed-model problems are usually transferred to single-model problems by balancing the average model mix along the line. Common tasks for multiple models are required to be assigned to the same stations. Capacity constraints are modeled so that the average processing time is within the cycle time at each station. With increasing vehicle heterogeneity, however, the representativity of the average model mix declines. We therefore focus our review on papers that treat MMAL balancing by explicitly considering capacity constraints for each model and that, similar to the FALDP, permit task duplication. Roberts and Villa (1970) were among

the first to consider the assignment of common tasks to multiple stations. J. Bukchin, Dar-El, and Rubinovitz (2002) divide the assembly tasks into two groups. The tasks in the first group require costly equipment and are therefore not allowed to be duplicated. The second group consists of manual tasks which can be duplicated. Y. Bukchin and Rabinowitch (2006) assign cost parameters for duplicated tasks. Their objective is to minimize the sum of station opening and task duplication cost. None of the reviewed papers allows for splitting the workload of one model among task duplicates at different stations (routing flexibility). Also, we did not find any paper that allows for multiple process plans to be used in parallel (operation flexibility). Routing and operation flexibility are thus not considered within MMAL balancing.

The row layout design problem deals with the arrangement of machines in a flexible manufacturing system. It can be differentiated between single-row, double-row, and cluster (multi-row) layouts (Heragu & Kusiak, 1988). Depending on the material handling device used, the rows are arranged linearly on a straight line, in a U- or serpentine shape, or in a loop. The flow direction can be uni- or bidirectional. A summary on row layout arrangements, relevant objectives, and possible solution methods is available in Keller and Buscher (2015). We limit our review to papers that consider routing flexibility by allowing the flow to be allocated among multiple machine duplicates. The reviewed row layout design problems are comparable to the integrated station location and flow allocation problem of the FALDP. Ho and Moodie (1998) investigate uni- and bidirectional loop and non-loop flow paths in single-row layouts. In their approach, the flow volumes between the machine duplicates are defined a priori. In contrast, Chen, Wang, and Chen (2001) as well as Aneke and Carrie (1986) include flow allocation decisions in their problem definition. Chen et al. (2001) seek to minimize the flow distance in unidirectional, linear single-row layouts. The number of machine duplicates is limited and backtracking is not permitted. Aneke and Carrie (1986) evaluate the trade-off between backtracking and the utilization of duplicated machines in a similar problem setting. To the best of our knowledge, there is no paper published in this field of research that considers process plan alternatives and therefore operation flexibility.

Cell formation is the key design problem in a cellular manufacturing system. It optimizes the grouping of machines into cells and parts into families such that the majority of operations of a part family take place in a single cell (Goldengorin, Krushinsky, & Pardalos, 2013). Cell formation shares similarities with the station formation problem of the FALDP. However, it is different as the machines assigned to a cell can usually be operated simultaneously. In case of the FALDP, the tasks assigned to a station cannot be performed at the same time. Another difference is that the assignments of tasks to

stations in the FALDP is not aimed at a complete elimination of flows between stations. Papaioannou and Wilson (2010) published an overview on the cell formation literature. In our review, we concentrate on papers that consider machine duplicates (routing flexibility) and/or multiple process plans (operation flexibility). Rajamani, Singh, and Aneja (1990) show that the presence of alternative process plans improves cell formation when part families and machine groups are identified simultaneously. In their model, machine duplicates exist, but a unique machine has to be chosen for each operation of a part. Sofianopoulou (1999) considers the same problem. She develops two binary programs, i.e., a machine allocation and a part allocation model, which are solved sequentially. Caux, Bruniaux, and Pierreval (2000) decompose the cell formation problem into a machine partitioning and a route selection problem, which are solved iteratively. Solimanpur, Vrat, and Shankar (2004) develop a multi-objective binary program in which they consider part similarity within cells, processing cost, processing time, and investment cost as objectives. All research articles discussed so far assume that for each part a single process plan is to be chosen out of a given set of alternatives. Conversely, Heragu and Chen (1998) allow multiple process plans to be used simultaneously. However, they assume that the proportion of parts that follow a certain process plan is a given input parameter. The paper of Nagi, Harhalakis, and Proth (1990) is the only reference we have found that optimizes the proportions of parts produced along different process plans and thereby takes full advantage of operation flexibility. Our literature review reveals that routing flexibility is frequently considered in cell formation. In contrast, operation flexibility is only considered in Nagi et al. (1990).

Job shop design problems are typically addressed in the context of facility layout design. Facility layout design considers the positioning of facilities, e.g., machines, cells, or departments, on the plant floor. A good overview on problem characteristics and solution methods is provided by Drira, Pierreval, and Hajri-Gabouj (2007). In a classical job shop, facility duplicates are positioned in adjacent locations. Since long flow distances are inherent drawbacks of such an arrangement, the literature also discusses *distributed layouts* in which facility duplicates are allowed to be placed in non-adjacent locations (Benjaafar & Sheikhzadeh, 2000). We focus our review on distributed layout design problems that consider machine duplicates (routing flexibility) and/or multiple process plans (operation flexibility). The distributed layout design problem is comparable to the integrated station location and flow allocation problem of the FALDP. Benjaafar and Sheikhzadeh (2000) analyze the design of distributed layouts considering routing flexibility in stochastic environments. Their research contributes two major results. First, they show that it is beneficial to place duplicates in non-adjacent locations to hedge against

uncertainties. Second, they prove that the marginal benefits of duplication are decreasing, i.e., that the first duplicate contributes the highest benefit. These results are in line with the findings of Montreuil (1999) in the context of fractal layouts. Urban, Chiang, and Russell (2000) as well as Jaramillo and McKendall (2010) study the deterministic version of the same problem. While Urban et al. (2000) consider the number of facility duplicates as given, Jaramillo and McKendall (2010) view them as a decision. There is only limited literature on facility layout design that takes operation flexibility into account. Askin and Mitwasi (1992) investigate the integrated facility layout design, process selection, and capacity planning problem. They assume a cellular setting in which the production volume of each product can be allocated among multiple process plans. Defersha and Hodiya (2017) study integrated distributed layout and cellular manufacturing systems design. In contrast to the model of Askin and Mitwasi (1992), they do not only model the inter-cell layout but also the intra-cell layout. Similar to our findings in cell formation, also the literature on job shop design considers routing flexibility more often than operation flexibility.

The FALDP shares features with the discussed design problems. However, there is no approach in the literature that covers the combination of station formation, station location, and flow allocation considering routing and operation flexibility as in the FALDP.

2.3 Flexible assembly layout design problem (FALDP)

In this section, we formulate an MILP for the FALDP. We investigate a segment of the final assembly at an automotive OEM in which highly variant, manual tasks are performed. The purpose is to derive layouts that allow for the efficient assembly of a given model mix. We consider $m \in M$ models with $t \in T_m$ required tasks. The shop floor is represented by $l \in L$ locations at which stations could be opened. The set of routes $r \in R$ includes all possible flow paths through the shop floor. We decide whether a station is opened at a location (variables X_l) and which tasks are assigned (variables $Y_{t,l}$). Also, we anticipate the models' flow allocations along the positions $i \in I_r$ of the routes (variables $Z_{m,r,t,i}$). The objectives are to minimize the number of opened stations as well as to minimize the flow intensity.

Figure 2.3 illustrates an FALDP solution. The shop floor has 16 locations $L1$ - $L16$ at which stations could be opened. The locations are arranged in four rows and four levels. The shown solution consists of nine stations that are positioned at locations $L2$, $L3$, $L5$, $L6$, $L7$, $L9$, $L10$, $L11$, and $L12$. The assigned tasks to the stations are indicated in the

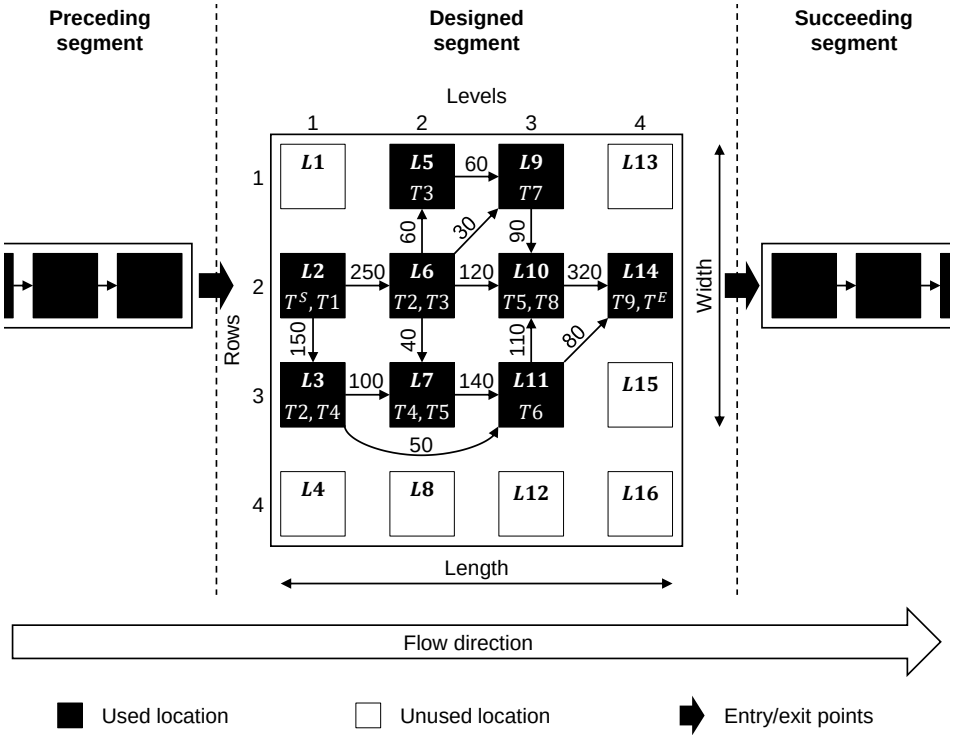


Figure 2.3: FALDP solution sketch.

figure, e.g., tasks $T2$ and $T3$ are assigned to the station at location $L6$. The numbers next to the arrows indicate the anticipated flow volumes.

Common entry and exit points are required in order to enable the interconnection to the preceding and succeeding segments. The exact locations of the entry and exit points are determined in the FALDP. In the example in Figure 2.3, locations $L2$ and $L14$ are chosen. Within the segment, we aim for a directed flow. This has numerous benefits such as a simplification of the AGV system design and control. Therefore, the created layout should be longer than it is wide. In Figure 2.3, the layout has a length of four stations and a width of three stations. In addition, the AGVs are only allowed to travel downstream and between stations on the same level. As can be seen in the example, no arcs point to the left. For all vehicles, the flow paths should be predominantly directed in the flow direction. Also, the same station should only be visited once. Otherwise, workers might get confused or perform tasks earlier or later than planned.

The layout is designed for a given model mix and volume that should be produced in a given production time. In the example in Figure 2.3, the total production volume is 400 vehicles. Each model requires that a set of specific tasks be performed. The task sequence is not predefined but mutable within the precedence relations, resulting

in multiple potential process plans. Different units of the same model can be assembled along different process plans (operation flexibility). The assembly tasks are assigned to stations so that the workload at each station does not exceed the production time. The workloads of the different tasks assigned to the same station are added. These tasks cannot be performed simultaneously, because only one employee works at each station. However, the same task can be performed at multiple stations (routing flexibility). In the example, task $T2$ is assigned to locations $L3$ and $L6$. To limit the adverse effects of disruptions, caused for example by equipment malfunctioning, we enforce full routing flexibility. This means that whenever there is a precedence relation between two tasks for any model, none of the assignments of the successor task is positioned upstream of any assignment of the predecessor task. In this way, the AGVs can always access all task assignments and the robustness of the obtained layout is increased. As an example, assume there is a precedence relation between tasks $T3$ and $T5$ in the instance plotted in Figure 2.3. Full routing flexibility means that no assignment of task $T5$ is positioned to the left of any assignment of task $T3$.

We make five assumptions. First, we assume that the shop floor is represented by a discrete grid of potential station locations. The locations on the shop floor are arranged in a grid of rows and levels as illustrated in Figure 2.3. The discrete representation is sufficiently detailed and common practice for strategic layout design problems. Second, we neglect the AGV flows from the exit point back to the entry point, because we assume that the best return path is designed subsequently based on the results of the FALDP. Third, we assume that the stations are of equal size. Since manual labor is predominant in segments suitable for FALs, the stations do not need to accommodate large machinery and equipment. Also, stocking areas are not needed at the stations, because the parts are supplied in kits that are delivered on the AGVs. Fourth, we neglect task assignment cost. Duplicating manual tasks is relatively cheap, because no major equipment redundancies are required. However, we limit the maximum number of duplicates for each task since the marginal benefits of duplication are decreasing (Benjaafar & Sheikhzadeh, 2000). Finally, we assume that all input data, especially demand and task times, are deterministic as is common practice in industry.

We use the notation as summarized in Table 2.1. Given a shop floor with a set of potential station locations L, the FALDP is the problem of determining the minimum number of stations to be opened as well as their positions and assigned tasks in order to minimize the flow intensity. The static, deterministic demand for a set of models M needs to be satisfied. The set of tasks T includes two dummy tasks T^S and T^E, which represent common start and end activities for all models and are needed for implemen-

Table 2.1: Problem notation.

Index sets	
$m \in M$	Models
$t \in T$	Tasks
$[T^S, T^E] \subset T$	Dummy start and end tasks
$T \backslash [T^S, T^E]$	Real tasks
$T_m \subseteq T$	Tasks for model m
$t_2 \in V_{m,t}$	Successor tasks of task t for model m (precedence relations): t to be finished before t_2 starts
$l \in L$	Locations
$r \in R$	Routes: potential AGV flow paths
$i \in I_r$	Position index on route r: $i = 1, ..., \|r\|$
Parameters	
w_r	Distance on route r
d_m	Demand for model m
$q_{m,t}$	Task time of task t for model m
τ	Production time
b_t	Maximum number of duplicates of task t ($b_{T^S} = b_{T^E} = 0$)
e_l	Level index of location l
f_l	Row index of location l
$p_{r,l}$	Position index of location l on route r
Decision variables	
X_l	1 if station at location l is opened, else 0
$Y_{t,l}$	1 if task t is assigned to location l, else 0
$Z_{m,r,t,i}$	Units of model m that receive task t at ith location on route r

tation purposes. Both T^S and T^E have a duration of zero time units. All other tasks are real tasks. The set T_m comprises all tasks needed for model m. The feasible process plans are represented through the precedence relations of the models' tasks. The set of successor tasks $V_{m,t}$ indicates all tasks for model m that can only start after task t has been finished.

The set of routes R comprises all potential flow paths of the AGVs from the entry to the exit point. A route is defined as a unique sequence of visited locations along which an AGV can navigate. Considering that the locations used as well as the positions of the entry and exit points are decisions and not defined a priori, the cardinality of R is fairly large. However, we will show in Section 2.4 how our solution approaches overcome this obstacle. The flow restrictions of the AGVs can be mapped in the route definition. We only generate routes that consist of up-, down-, and rightward moves, but no leftward (backward) moves. Also, we only generate routes that are predominantly directed in the flow direction and that visit a location only once. The traveled distance along each route is known.

There are three types of decision variables. First, the binary variable X_l shows whether a station is opened at location l. Second, $Y_{t,l}$ is a binary variable that states whether

task t is assigned to location l. Third, the continuous variable $Z_{m,r,t,i}$ indicates the units of model m that receive task t at the ith position on route r. The FALDP can then be formulated as follows:

$$\min \ Z_1 = \sum_{l \in L} X_l \tag{2.1a}$$

$$\min \ Z_2 = \sum_{r \in R} \sum_{m \in M} w_r \cdot Z_{m,r,T^S,1} \tag{2.1b}$$

s.t.

$$\sum_{l \in L} Y_{t,l} \leq 1 + b_t \qquad \forall t \in T \tag{2.1c}$$

$$\sum_{t \in T} Y_{t,l} \leq |T| \cdot X_l \qquad \forall l \in L \tag{2.1d}$$

$$\sum_{m \in M} \sum_{t \in T_m} \sum_{r \in R | l \in r} q_{m,t} \cdot Z_{m,r,t,p_{r,l}} \leq \tau \qquad \forall l \in L \tag{2.1e}$$

$$\sum_{r \in R} Z_{m,r,T^S,1} = d_m \qquad \forall m \in M \tag{2.1f}$$

$$\sum_{i \in I_r} Z_{m,r,t,i} = Z_{m,r,T^S,1} \qquad \forall m \in M, r \in R, t \in T_m \backslash T^S \tag{2.1g}$$

$$Z_{m,r,t_2,i} \leq \sum_{j \in I_r | j \leq i} Z_{m,r,t,j} \qquad \forall m \in M, r \in R, t \in T_m, t_2 \in V_{m,t}, i \in I_r \tag{2.1h}$$

$$\sum_{m \in M | t \in T_m} \sum_{r \in R | l \in r} Z_{m,r,t,p_{r,l}} \leq Y_{t,l} \cdot \sum_{m \in M} d_m \qquad \forall t \in T, l \in L \tag{2.1i}$$

$$\sum_{l_2 \in L | e_{l_2} < e_l} Y_{t_2,l_2} \leq |L| \cdot (1 - Y_{t,l}) \qquad \forall m \in M, t \in T_m, t_2 \in V_{m,t}, l \in L \tag{2.1j}$$

$$\sum_{l \in L | f_l = \bar{e}} X_l \leq |L| \cdot \sum_{l \in L | e_l > \bar{e}} X_l \qquad \forall \bar{e} \in 1, \ldots, \max_{l \in L} e_l \tag{2.1k}$$

$$X_l \in \{0, 1\} \qquad \forall l \in L \tag{2.1l}$$

$$Y_{t,l} \in \{0, 1\} \qquad \forall l \in L, t \in T \tag{2.1m}$$

$$Z_{m,r,t,i} \geq 0 \qquad \forall m \in M, r \in R, t \in T_m, i \in I_r \tag{2.1n}$$

We employ a lexicographic multi-objective formulation as shown in Equations (2.1a) and (2.1b). The Primary Objective (2.1a) is to minimize the number of opened stations. Because demand and production time are fixed, minimizing the number of opened stations is equivalent to maximizing the efficiency of the layout, which is defined as the total workload divided by the installed capacity as shown in Equation (2.2).

$$\text{efficiency} = \frac{\text{total workload}}{\text{installed capacity}} = \frac{\sum\limits_{m \in M} d_m \cdot \sum\limits_{t \in T_m} q_{m,t}}{\tau \cdot \sum\limits_{l \in L} X_l} \tag{2.2}$$

As Subordinate Objective (2.1b), we minimize the flow intensity for the minimum number of stations. The flow intensity is an indicator for the transportation effort and is defined as the sumproduct of route distance and number of vehicles assembled along a route.

Constraints (2.1c) limit the maximum number of task duplicates. To achieve common entry and exit points, the dummy tasks T^S and T^E are not allowed to be duplicated, i.e., $b_{T^S} = b_{T^E} = 0$. Constraints (2.1d) ensure that tasks can only be assigned to opened stations. Constraints (2.1e) guarantee that the workload assigned to a station is less than the production time. Constraints (2.1f) guarantee demand fulfillment and Constraints (2.1g) ensure flow balance. Constraints (2.1h) are used to satisfy the precedence relations. They ensure for all precedence relations that the number of vehicles receiving the successor task at a location cannot be higher than the number of vehicles receiving the predecessor task at all preceding locations along a specific route. Constraints (2.1i) link the binary assignment variables $Y_{t,l}$ to the continuous flow variables $Z_{m,r,t,i}$. A positive flow is only allowed when the corresponding task is assigned to the corresponding location. Constraints (2.1j) enforce full routing flexibility. They forbid any assignment of a task t_2 that is a successor of task t for any model m to be positioned to the left of any assignment of task t. In other words, they make sure that all assignments of the successor task t_2 are accessible from any assignment of the predecessor task t by the AGVs, which are not allowed to travel backwards. This increase in flexibility is especially beneficial in the event of disruptions. We therefore refer to Constraints (2.1j) as *robustness constraints*. Constraints (2.1k) enforce the layout shape to be longer than it is wide. The constraints only allow locations to be used if their row index is lower than the level index of the last (right-most) used location. Finally, in Constraints (2.1l) - (2.1n), we restrict the domains of the decision variables.

Structurally, the FALDP is an integrated quadratic assignment and multi-commodity network flow problem with linear side constraints. Askin and Mitwasi (1992) have shown that this type of problem is NP-hard. In order to solve real-world instances, heuristic solution approaches are likely to be inevitable.

2.4 Solution approaches

In this section, we design approaches to derive solutions to the FALDP. We start by discussing layout properties in an optimal solution, which will be exploited in our solution approaches. We then propose an exact approach that is capable of finding an optimal solution for small- and mid-sized instances. Finally, we show how the exact approach can be transformed to a matheuristic approach that is capable of solving large-sized instances.

2.4.1 Preliminary considerations on layout properties in an optimal solution

Before designing an exact approach to solve the FALDP, we first investigate layout properties in an optimal solution. We define the layout as the assignment of stations to locations on the shop floor as well as the positioning of the entry and exit points. For a given number of stations, not all layouts can possibly lead to a solution with minimum flow intensity, because many layouts are dominated by other layouts.

We state the following dominance rule: given two layouts A and B with an equal number of stations, layout A is dominated by layout B whenever there is a mapping between the used locations in both layouts such that each route in A can be mapped to a feasible route in B which is shorter or equally long. For a route to be feasible, it cannot include any backward, i.e., leftward, transfer and it has to be predominantly directed in the flow direction. To reduce symmetry, we also eliminate weakly dominated layouts.

For illustration purposes, let us consider the two layouts with four stations shown in Figure 2.4. Keeping in mind that the assembly of each vehicle starts at entry point T^S and ends at exit point T^E, and that the AGVs are neither allowed to travel backwards nor in cycles, there exist four feasible routes in layout A. These routes are illustrated by the solid, dashed, dotted, and dashed-dotted arrows. Let us now consider the mapping in Table 2.2: we map locations $L1$, $L4$, $L8$, and $L10$ in layout A to locations $L1$, $L4$, $L7$, and $L8$ in layout B respectively. All four routes in A can then be mapped to feasible routes in B as indicated by the corresponding arrows. When comparing the route distances, we see that all mapped routes in B are shorter or equally long (cf. Table 2.3). We conclude that layout A is dominated by layout B. For any FALDP instance requiring four stations, the best solution in A cannot have a lower flow intensity than the best solution in B. In order to find an optimal solution, we do not need to consider layout A or any other dominated layout.

In the discussed example, we have set the distance between two neighboring locations to one distance unit and employed the Manhattan metric. The dominance rule, however,

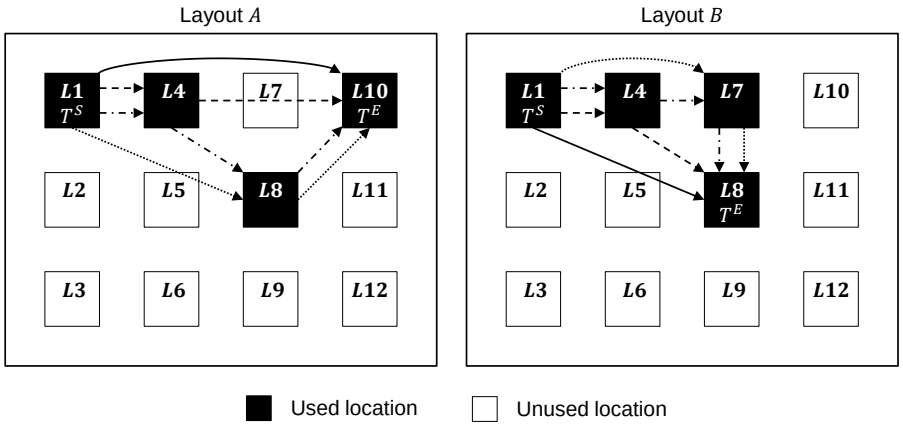


Figure 2.4: Example for dominance relations.

Table 2.2: Example for dominance relations: station mapping.

Station	Mapping	
	A	B
1	$L1$	$L1$
2	$L4$	$L4$
3	$L8$	$L7$
4	$L10$	$L8$

Table 2.3: Example for dominance relations: route distances.

Route	Station sequence	Location sequence		Distance	
		A	B	A	B
Solid	1-4	$L1$-$L10$	$L1$-$L8$	3.0	3.0
Dashed	1-2-4	$L1$-$L4$-$L10$	$L1$-$L4$-$L8$	3.0	3.0
Dotted	1-3-4	$L1$-$L8$-$L10$	$L1$-$L7$-$L8$	5.0	3.0
Dashed-dotted	1-2-3-4	$L1$-$L4$-$L8$-$L10$	$L1$-$L4$-$L7$-$L8$	5.0	3.0

can be applied to all other metrics as well. The sets of non-dominated layouts can be derived for any number of stations and are independent of the instance to be solved. We generate these sets up front and use them throughout our solution approaches.

2.4.2 An exact solution approach

In order to derive optimal solutions to the FALDP, we propose an iteratively solved problem decomposition as illustrated in Figure 2.5. To accommodate the lexicographic objective, we iterate in increasing order over the potential solution values of the primary objective, which is the minimization of the number of opened stations. In each iteration, we fix the value of the primary objective and search for the solution that minimizes the subordinate objective. That is, we solve the subproblem of finding a solution that

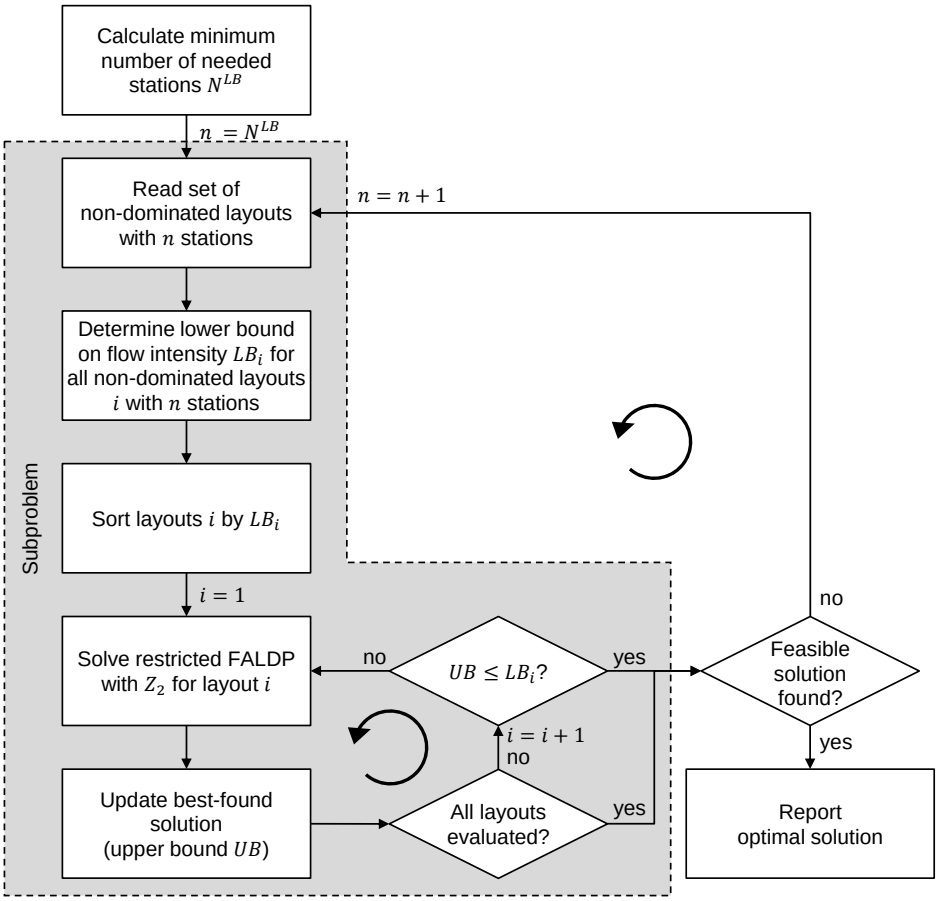


Figure 2.5: Flow chart of exact solution approach.

minimizes flow intensity for a fixed number of opened stations. As long as the subproblem is infeasible, we increase the number of opened stations by one and reiterate. An optimal solution to the FALDP is found as soon as the subproblem is feasible and yields an optimal solution. As the starting point for the iterations, we derive a lower bound on the number of required stations N^{LB}. The lower bound can be determined by rounding up the ratio of total workload to production time to the next integer as shown in Equation (2.3).

$$N^{LB} = \left\lceil \frac{\text{total workload}}{\text{production time}} \right\rceil = \left\lceil \frac{\sum_{m \in M} d_m \sum_{t \in T_m} q_{m,t}}{\tau} \right\rceil \tag{2.3}$$

When solving the subproblems, we make use of our observation that only non-dominated layouts need to be considered. For any number of opened stations n, we iterate over all corresponding non-dominated layouts to find a solution that minimizes flow intensity. By only considering the non-dominated layouts, we reduce the solution space of the subproblems significantly without excluding the optimal solution.

In order to conduct the evaluation of the non-dominated layouts in an intelligent order, our first step when solving a subproblem is to determine a lower bound on the flow intensity for each non-dominated layout based on the given FALDP instance. A lower bound is obtained by relaxing two sets of constraints in the FALDP, i.e., the constraints that limit the maximum number of task duplicates as well as the robustness constraints. By relaxing these two sets of constraints, all assignment restrictions for real tasks are excluded. We can thus assume that every real task can be performed at all stations. The flow allocation problem then becomes the simple problem of allocating the models' workloads among the available stations in the layout without considering task assignments. The minimum flow intensity for this relaxed problem is obtained by assigning the largest flow to the shortest routes. The largest flow is obtained for the models with the lowest workload. A similar lower bound procedure was proposed by Urban et al. (2000) in the context of the integrated machine allocation and layout problem. For each non-dominated layout, we employ the following three-step procedure:

Step 1: Let $dist_{l_1,l_2}$ be the distance between any pair of used locations l_1 and l_2 in the considered layout. Assuming T^S is assigned to location L^S and T^E is assigned to location L^E, sort all used locations l in increasing order of their combined distances to L^S and L^E: $\widehat{dist_l} = dist_{L^S,l} + dist_{l,L^E}$.

Step 2: Sort all models m in increasing order of their workloads: $v_m = \sum_{t \in T_m} q_{m,t}$.

Step 3: Fill the available capacity of the stations in the layout in increasing order of $\widehat{dist_l}$ with the workloads of the models in increasing order of v_m. Flow intensity is thereby minimized.

The quality of the lower bound is affected by the task assignment restrictions. In the FALDP, task assignments are restricted by the maximum number of task duplicates and the robustness constraints. If the number of task duplicates were unlimited and there were no precedence relations between the tasks (and therefore no robustness constraints), then the lower bound and the optimal solution would coincide. The more restrictive the maximum number of task duplicates and the more precedence relations, the less tight is the lower bound.

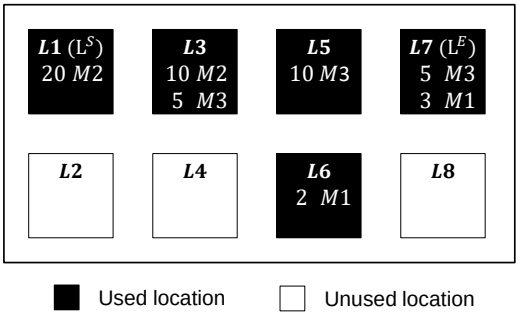


Figure 2.6: Example for lower bound calculation ($\tau = 60$).

Table 2.4: Example for lower bound calculation: model data.

Model	Demand d_m	Workload v_m
$M1$	5	10.0
$M2$	30	3.0
$M3$	20	6.0

For illustrating the lower bound calculation, consider the layout with five stations in Figure 2.6. L^S and L^E correspond to locations $L1$ and $L7$ respectively. We assume Manhattan metric and a distance of one distance unit between neighboring locations. The production time is 60 time units. Exemplary data for three models is given in Table 2.4. First, we sort the used locations in increasing order of their combined distances to L^S and L^E. Locations $L1$, $L3$, $L5$, and $L7$ have a combined distance of 3.0 distance units each, whereas location $L6$ has a combined distance of 5.0 distance units. Next, we sort the models in increasing order of their workloads v_m, i.e., $M2$, $M3$, $M1$. Finally, we fill the models' workloads on the stations. We start by assigning as many vehicles of model $M2$ to location $L1$. Because the production time is 60 time units and the vehicles of model $M2$ have a workload of 3.0 time units, 20 vehicles can be assigned to $L1$. The remaining ten vehicles of model $M2$ are assigned to the next location $L3$. Afterwards, we allocate the vehicles of model $M3$. Location $L3$ has a remaining capacity for five vehicles of model $M3$. Ten vehicles of model $M3$ are assigned to location $L5$ and the remaining five vehicles to location $L7$. Finally, three vehicles of model $M1$ fit on location $L7$. The remaining two vehicles of model $M1$ need to be assigned to location $L6$. By taking the sumproduct of the number of assigned vehicles and the combined distance to L^S and L^E for all used locations, we derive the lower bound on the flow intensity. For the shown example, the lower bound is $20 \cdot 3 + (10 + 5) \cdot 3 + 10 \cdot 3 + (5 + 3) \cdot 3 + 2 \cdot 5 = 169$ distance units.

After obtaining a lower bound on the flow intensity for all non-dominated layouts, we sort the layouts in increasing order of their lower bounds. We start the subproblem iterations by evaluating the layout whose lower bound on flow intensity has the lowest value. We then iterate over all non-dominated layouts in increasing order of their lower bounds. The subproblem iterations consist of two steps. First, we solve the restricted FALDP for the considered layout using the subordinate objective to minimize flow intensity. Because the locations used as well as the positions of the entry and exit points are specified by the layout, the problem is simplified substantially: *i)* all X_l variables are eliminated, *ii)* the $Y_{t,l}$ variables are eliminated for locations which are not used, and *iii)* the $Z_{m,r,t,i}$ variables are eliminated for routes that visit locations which are not used.

The reduced problem size and complexity allow us to solve this restricted problem for small- to mid-sized instances using a generic MILP solver. In the second step, we update the best-found solution whenever we obtain a layout with a lower flow intensity than the incumbent best-found solution.

The best-found solution provides an upper bound on the optimal flow intensity. We can therefore define a stopping criterion for the subproblem iterations. We stop whenever we find a layout with a flow intensity that is lower than or equal to the lower bound of all remaining yet unevaluated layouts. In case no feasible solution has been found after evaluating all non-dominated layouts, we increase the number of stations by one and solve the new subproblem. Otherwise, we terminate and report the solution with the lowest flow intensity, which is an optimal solution for the FALDP.

2.4.3 A matheuristic solution approach

The exact solution approach is capable of finding an optimal solution for small- to mid-sized instances. For large-sized instances, however, prohibitive run times occur.

We are facing two challenges when applying our exact solution approach to large instances. The first challenge is that the number of routes $r \in R$ increases exponentially with the number of opened stations, as depicted on the left-hand side of Figure 2.7. This leads to a substantial increase in the sizes of the restricted FALDPs and slows down the solution time of each iteration in the subproblems. The second challenge is that the number of non-dominated layouts also increases exponentially with the number of opened stations, as shown on the right-hand side of Figure 2.7. Consequently, we need to perform many iterations in the subproblems to prove optimality.

In order to find good solutions to the FALDP in acceptable time for instances of all sizes, we propose to solve the restricted FALDPs heuristically. We design a matheuristic approach as shown in Figure 2.8.

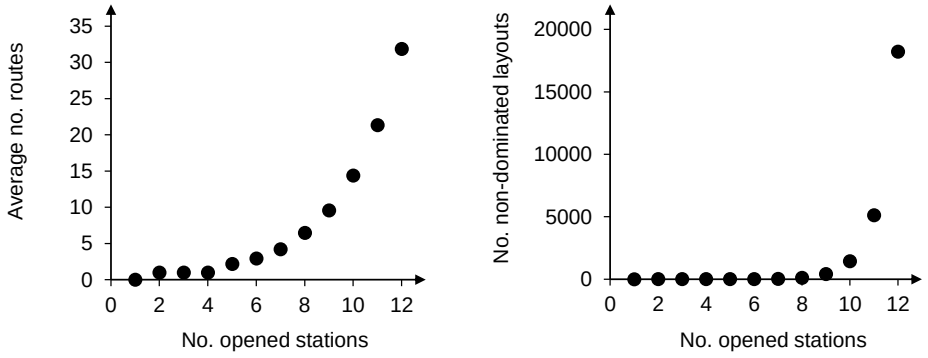


Figure 2.7: Increase in average number of routes in restricted FALDPs (left) and number of non-dominated layouts (right) with number of opened stations.

The matheuristic to solve the restricted FALDPs is composed of two heuristic stages. In the first stage, the route reduction stage, we reduce the set of routes to a subset of promising routes R^P. In an optimal solution, it is unlikely that all potential routes are used. We identify promising routes as all routes that are shorter or equal to a threshold distance. The threshold distance is defined as the minimum distance such that all locations in the layout are covered in at least one promising route.

In the second stage, the fix-optimize stage, we use a fix-optimize approach to iteratively improve an initial solution. We construct the initial solution by solving the restricted FALDP as a feasibility problem that considers the subset of promising routes R^P. Solving the restricted FALDP as a feasibility problem is much faster than solving it as an optimization problem. It allows us to quickly check whether a layout is capable of improving the incumbent best-found solution. If a feasible solution that is not worse than the incumbent best-found solution (UB) exists, we start the fix-optimize iterations. Otherwise, we proceed with the next non-dominated layout. The fix-optimize iterations consist of three steps. First, we randomly fix $\gamma\%$ of the binary task-location assignments in the incumbent solution. In the solution shown in Figure 2.3, there are 15 task-location assignments. For $\gamma = 80\%$, we would fix twelve of these assignments. For example, we might fix all assignments except for $(L9, T7)$, $(L10, T8)$, and $(L11, T6)$. We employ a tabu list in order to avoid analyzing the same fixings multiple times. In the second step, we optimize the restricted FALDP with the objective of minimizing flow intensity while fixing the task-location assignments that have been selected in the first step. Because a large proportion of the binary variables is fixed, a generic MILP solver can solve this optimization problem in short CPU time. Note that the solution value of the incumbent solution can only improve during the iterations since the last incumbent solution is part

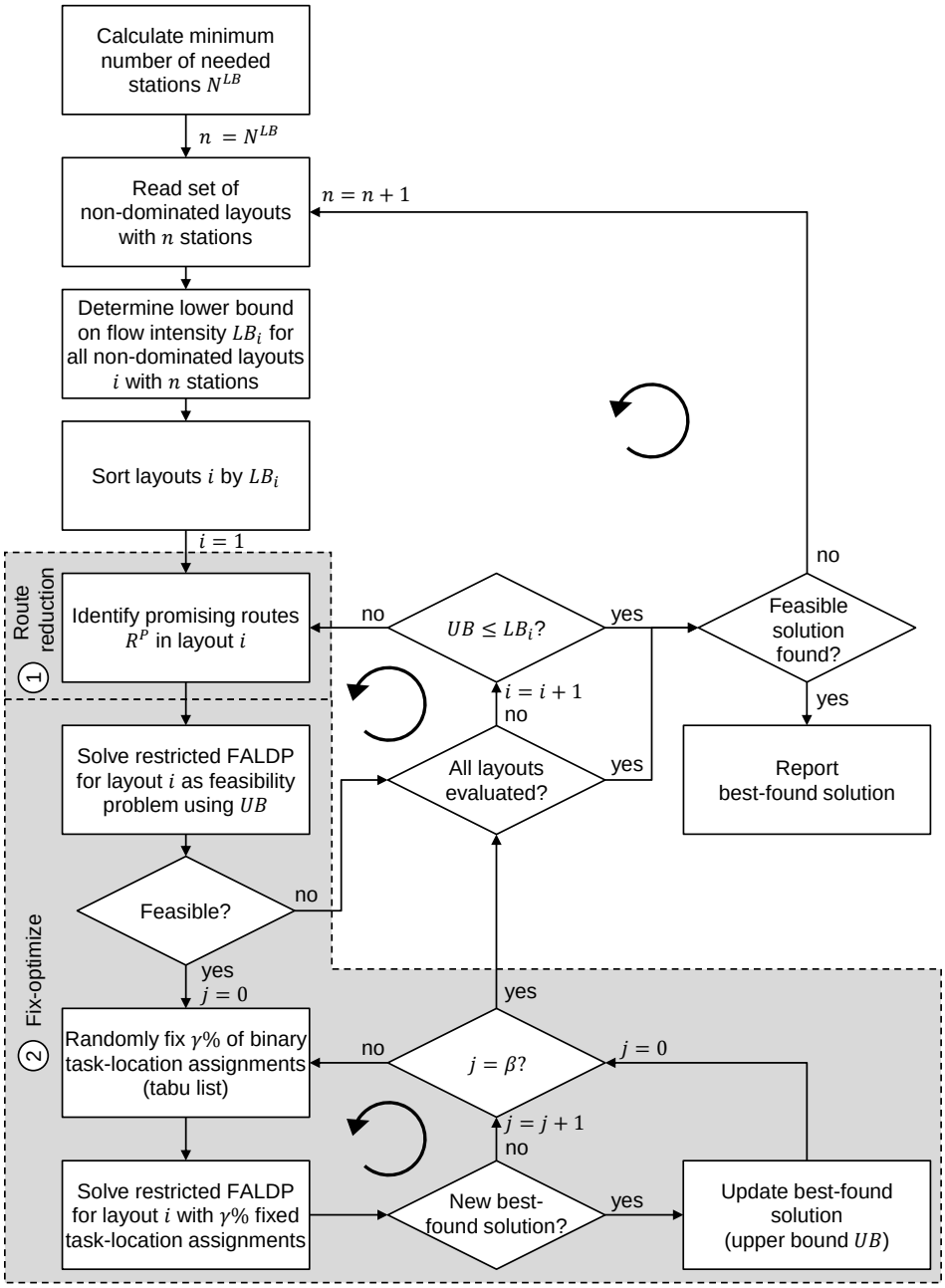


Figure 2.8: Flow chart of matheuristic solution approach.

of the solution space of the next iteration. In the third step, we update the best-found solution. The fix-optimize iterations are stopped whenever we observe β consecutive non-improving iterations. For example, we stop whenever 25 non-improving iterations occur. Afterwards, we evaluate the next non-dominated layout.

2.5 Computational results and empirical analysis

The computational analysis is structured into five parts. First, we describe the generation of our instance set and the design of experiments. In the second part, we evaluate the computational performance of our solution approaches. Next, we summarize the key design characteristics of the obtained solutions. In the fourth part, we compare the efficiency of FALs to LALs. Finally, we investigate the effect of vehicle heterogeneity on the efficiency of both types of layout.

2.5.1 Instance generation and design of experiments

In order to assess the performance of our FALDP solution approaches, we evaluate a total of 528 instances. We base our instances on the data set of Scholl (1993). This data set is commonly used as benchmark for simple assembly line balancing problems. The original data set can be obtained on the website `http://assembly-line-balancing.mansci.de`. The instances that we use for our computational analysis comprise between seven and 53 tasks, which are reasonable numbers. In discussions with OEMs, we learned that they are planning segments with 20 to 25 tasks. Note again that we are only investigating a limited segment of the automotive assembly. Also, we do not consider elementary worker moves as tasks, but bundled worker moves that should be performed together. We allow each task – except for dummy tasks – to have at most one duplicate ($b_t = 1 \; \forall t \in T \backslash [T^S, T^E]$). Allowing more duplicates per tasks is possible. However, we realize that there are decreasing marginal benefits of task duplication. Additional duplicates would also require training more workers to perform a task and possibly purchasing more tools. We fix the production time τ to 100 000 time units. The cycle time c is generated based on the method proposed by Hoffmann (1992). Let t^{sum} be the sum of task times and t^{max} the maximum task time in the simple assembly line balancing instance. Hoffmann (1992) derives the cycle time according to Equations (2.4) and (2.5).

$$n^{max} = \left\lfloor \frac{t^{sum}}{t^{max}} \right\rfloor \tag{2.4}$$

$$c(n) = \left\lceil \frac{t^{sum}}{n} \right\rceil \qquad \forall n = \left\lfloor \frac{n^{max}}{2} \right\rfloor, ..., n^{max} \tag{2.5}$$

We use the highest value of the theoretical minimum number of stations $n = n^{max}$ as the arrangement in an FAL does not make sense for low numbers of stations. The total demand can then be determined by dividing production time through cycle time, i.e.,

$\sum_{m \in M} d_m = \lfloor \tau / c \rfloor$. To ensure feasibility, we round the obtained value to the nearest lower integer.

Since the data set of Scholl (1993) is used for the simple assembly line balancing problem, the instances only consists of a single model. We use a procedure similar to that of Li and Gao (2014) to derive multi-model instances. As in Li and Gao (2014), we consider five models in each instance and randomly assign the total demand among the five models. The number of models might be higher in reality. However, the number of models only needs to reflect the heterogeneity in the considered segment and not the entire assembly. When two models have the same assembly requirements in the considered segment, they can be treated as identical in the FALDP.

In order to generate instances with heterogeneous vehicles, we consider four levels of structure heterogeneity (sh), i.e., 0%, 10%, 25%, and 50%. Structure heterogeneity refers to the degree of dissimilarity in the structures of the models' precedence graphs. It is defined as the average percentage difference between the number of tasks in the minimum supergraph and the precedence graphs of all models. In our base case ($sh = 0\%$), all models need all tasks. Thus, the structures of the models' precedence graphs are identical and match the precedence graphs in the data set of Scholl (1993). For $sh = 25\%$, we randomly remove 25% of the task-model assignments in the base case. We make sure that each task is needed by at least one model and that each model needs at least two real tasks. Similarly, we consider four levels of task time heterogeneity (tth), i.e., the task times of each model $q_{m,t}$ are allowed to deviate from the demand-weighted mean task time $\bar{q}_t$ by $\pm 0\%$, $\pm 10\%$, $\pm 25\%$, and $\pm 50\%$. In our base case ($tth = 0\%$), all models have the same task times $q_{m,t} = \bar{q}_t \ \forall m \in M, t \in T_m$. For $tth = 25\%$, we choose task times such that $q_{m,t} \in [0.75\bar{q}_t, 1.25\bar{q}_t] \ \forall m \in M, t \in T_m$ and such that the heterogeneity of task times is maximized. In order to have comparable instances, the demand for each model as well as the overall workload for each task are kept constant across all structure and task time heterogeneity levels. A detailed explanation of our instance generation scheme is provided in Appendix A.1.

The scheme to generate instances with different levels of vehicle heterogeneity is illustrated in Figure 2.9. By way of example, we show how to derive instances with 0% and 25% structure and task time heterogeneity from a fictitious base case. In the precedence graphs shown, the values above the task nodes represent the task times and the values below the model nodes represent the demands. In the upper left corner, the base case is plotted, in which the precedence graphs and the task times of all models are identical. Consequently, structure and task time heterogeneity are both 0%. The instance with 25% structure and 0% task time heterogeneity is shown in the upper right corner.

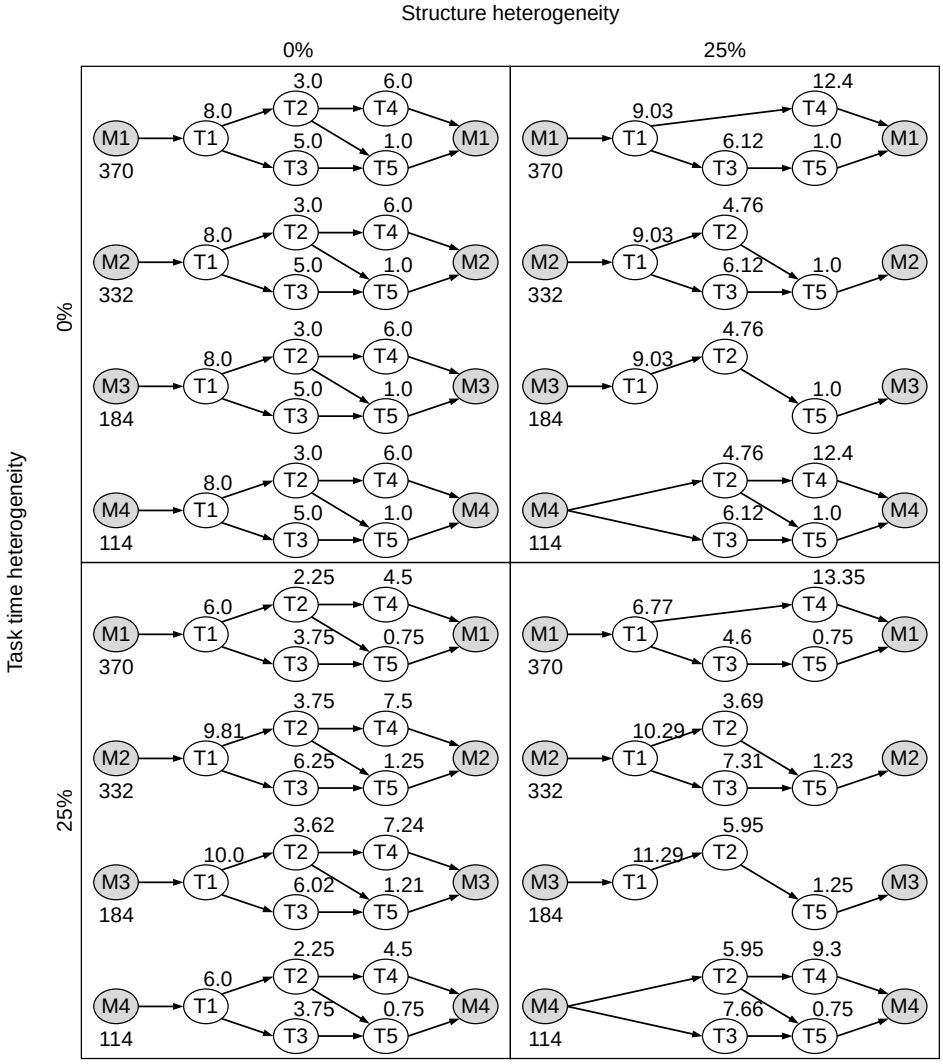


Figure 2.9: Instance generation scheme.

Twenty-five percent of the 20 task-model nodes in the base case have been removed randomly. The task times are identical for all models. For example, task $T3$ always requires 6.12 time units. The task times are derived such that the overall workload for each task, i.e., the sumproduct of demands and task times, is the same as in the base case. For example, the overall workload for task $T3$ is 5000 time units in both cases. Because task $T3$ is removed for model $M3$, the average task time of $T3$ increases compared to the base case. In the lower left corner, the instance with 0% structure and 25% task time heterogeneity is shown. As can be seen, the structures of the precedence graphs are identical to the base case. The task times, however, are not. For example, the task

times for task $T1$ are allowed to deviate from the demand-weighted mean $\bar{q_{T1}} = 8.0$ by 25%, i.e., within the range $[6.0, 10.0]$. The task times are chosen such that their heterogeneity is maximized and such that the overall workload for each task is identical to the base case. Finally, the combination of 25% structure and 25% task time heterogeneity is depicted in the lower right corner. First, we randomly remove twenty-five percent of the 20 task-model nodes in the base case. Second, we derive the demand-weighted mean task times $\bar{q_t}$ for constant overall workload. Third, we choose the task times from the interval $[0.75\bar{q_t}, 1.25\bar{q_t}]$ such that their heterogeneity is maximized and the overall workloads are the same as in the base case.

We repeat the instance generation scheme using three different random seeds. When combining all four structure and task time heterogeneity levels with the three random seeds, we obtain a total of 48 multi-model instances for each single-model instance in the data set of Scholl (1993). The exact data set used in our analysis can be obtained upon request from the corresponding author.

Concerning the underlying shop floor layout, we assume a checkerboard pattern. We employ the Manhattan metric to measure distances as this best reflects potential AGV pathways. The application of other metrics is straightforward. Without loss of generality, we set the distance between two neighboring locations to one distance unit.

In order to compare the efficiency of FALs to LALs, we determine the solutions to all our instances for both an FAL as well as an LAL. To find the best LAL, we adapt the MMAL balancing model by Y. Bukchin and Rabinowitch (2006) as shown in Appendix A.2. We use the model by Y. Bukchin and Rabinowitch (2006), because it allows for task duplication and explicitly considers capacity constraints for individual models. The MMAL balancing problems are characterized by much lower complexity than the FALDPs and can be solved efficiently using a generic MILP solver.

Our solution approaches and the random instance generation are implemented in a program application written in Python 3 and interfaced with Gurobi 7.5.1. All experiments are run on a computer using an Intel Xeon E5-4660 processor with 2.10 GHz and 8 GB RAM.

2.5.2 Computational performance

Figure 2.10 shows the distributions of the CPU times of the exact approach for different intervals of the number of tasks. As can be traced by the shape of the distributions, the CPU times of the exact approach grow rapidly with the number of tasks. While all instances with up to twelve tasks require CPU times of less than one second, the CPU times increase to approximately ten seconds for instances comprising 13 to 24

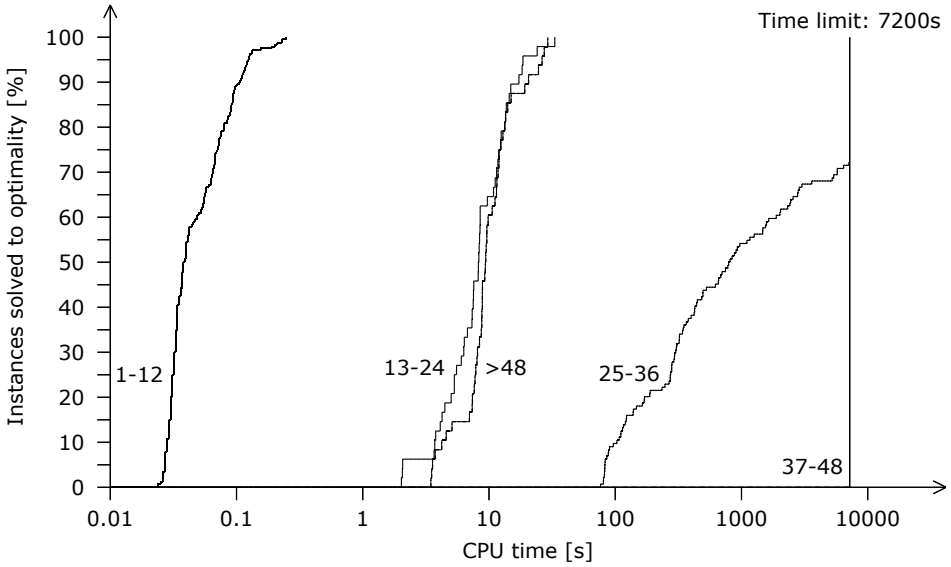


Figure 2.10: Solution performance of exact approach for different intervals of the number of tasks.

tasks. Prohibitive CPU times inhibit us from executing the exact approach on many instances comprising more than 24 tasks. For the group of instances comprising 25 to 36 tasks, we can solve only 72.2% of the instances to optimality within a time limit of 7200 seconds. For the group of instances comprising 37 to 48 tasks, the exact approach does not terminate for any instance within 7200 seconds. The instances comprising more than 48 tasks do not follow this trend. The exact approach can handle all of them in CPU times of around ten seconds. The fast CPU times are only due to the characteristics of the considered instances in our data set and not generalizable. We use standard instances from the literature. In the considered instances with more than 48 tasks, i.e., the instances based on Hahn (1972) in the data set of Scholl (1993), only a relatively small number of stations is needed even though the number of tasks is high. In total, the exact approach is capable of solving 440 out of the 528 instances to optimality within a time limit of 7200 seconds.

In order to justify our matheuristic solution approach, we investigate the effect of the two heuristic stages on the solution time and quality. For our evaluation, we use the subset of 440 instances that the exact approach is capable of solving to optimality within a time limit of 7200 seconds.

We first investigate the effect of the route reduction stage. We therefore compare the solutions of the exact approach when considering the full and the reduced set of routes. Our analysis shows that reducing the number of routes has no effect on the number of

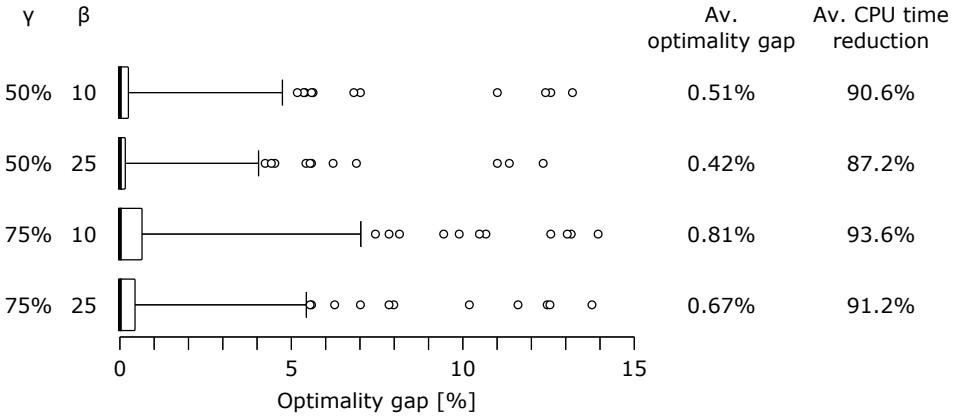


Figure 2.11: Comparing flow intensity optimality gaps and CPU time reductions for different parameter settings in the matheuristic.

stations in the obtained solutions for all 440 instances. Hence, the optimality gap on efficiency, our primary objective, is 0%. The average optimality gap on flow intensity, our subordinate objective, is, at 0.14%, extremely low. The average CPU time reduction is 88.3%. Using the route reduction heuristic alone, we can solve 465 out of the 528 instances within a time limit of 7200 seconds.

Next, we elucidate the effect of introducing the fix-optimize stage instead of solving the restricted FALDPs brute-force. We evaluate different parameter settings for the percentage of fixed task-location assignments γ and for the maximum number of consecutive non-improving iterations β. For γ, we tested the values 50% and 75%. For β, we tested the values 10 and 25. By design, the fix-optimize stage cannot deteriorate the solution quality of the primary objective. Therefore, the optimality gap on efficiency remains 0% for all 440 instances. In Figure 2.11, we summarize the results of the analysis concerning solution quality of the subordinate objective. The figure shows the box plots on the optimality gap on flow intensity as well as the reduction in CPU time compared to the exact approach for different combinations of the parameters γ and β. The box plots are highly skewed. For all parameter combinations, the median is 0%. This means that our matheuristic finds the optimal flow intensity for more than half of the 440 instances. As expected, the solution quality improves when fixing fewer task-location assignments ($\gamma = 50\%$) and allowing a larger number of consecutive non-improving iterations ($\beta = 25$). In contrast, the solution time improves when fixing more task-location assignments ($\gamma = 75\%$) and allowing a smaller number of consecutive non-improving iterations ($\beta = 10$). For all parameter combinations, the matheuristic solution approach terminates within 7200 seconds on all 528 instances in our data set.

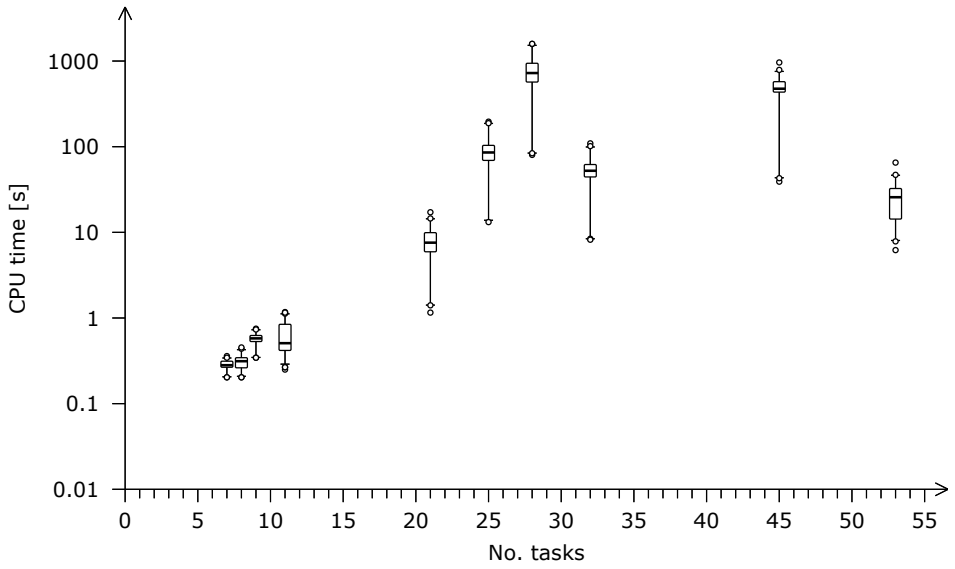


Figure 2.12: CPU time distribution of matheuristic subject to the number of tasks.

We use $\gamma = 50\%$ and $\beta = 25$ as parameter settings in our fix-optimize iterations, because they lead to the best solution quality while reducing the total solution time by almost 90%. Also, the reliability of these parameter settings is convincing. For 97.5% of the 440 instances, we observe optimality gaps below 4%. For only eight instances, we observe optimality gaps above 5%. The average optimality gap is 0.42%. We conclude that the reduction of routes deteriorates flow intensity by 0.14% and the fix-optimize iterations by 0.28%. Based on the good performance on small- and mid-sized instances, we expect that our matheuristic is capable of yielding high-quality solutions for large-sized instances as well.

Figure 2.12 shows the distribution of the CPU times of the matheuristic approach subject to the number of tasks. Similar to the exact approach, we observe an increase in CPU times with higher numbers of tasks. However, this increase is less severe. The matheuristic allows us to solve all 528 instances in acceptable time. The longest CPU time for an individual instance is 1588 seconds.

2.5.3 Characteristics of FALDP solutions

As an illustrative example, the solution to an FALDP instance comprising 32 tasks is given in Figure 2.13. The solution was generated using the exact approach. The solutions to most of the FALDP instances have similar characteristics to the one shown in the figure. First, the generated layouts are typically compact, meaning that they

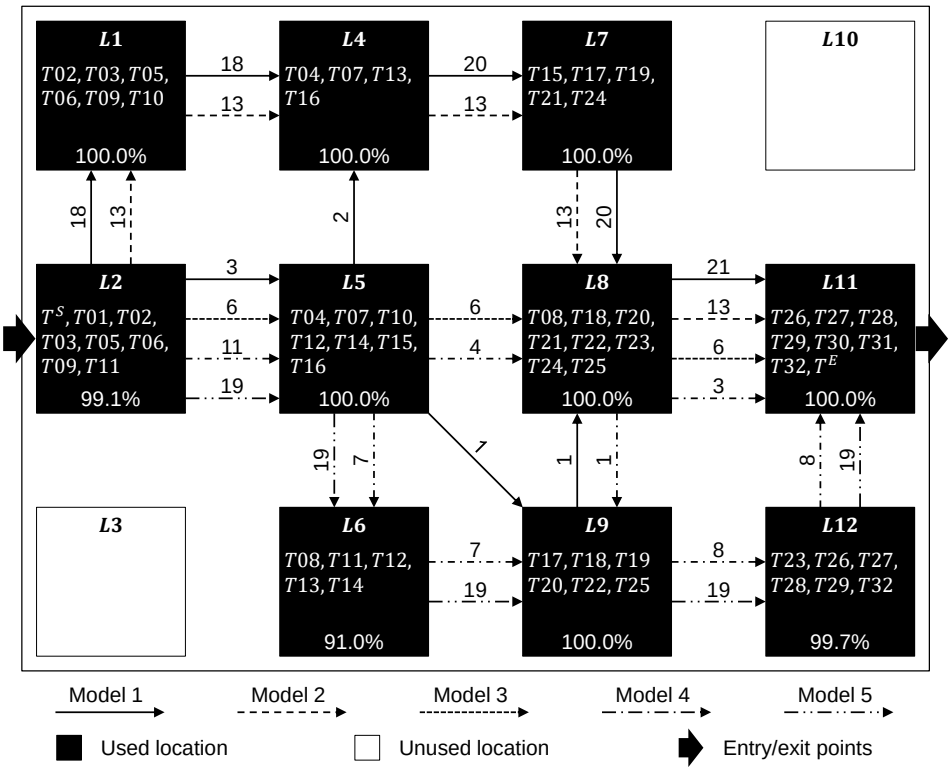


Figure 2.13: Illustrative example of an FAL.

are not much longer than they are wide. The layout in Figure 2.13 has a length of four stations and a width of three stations. Next, the entry and exit points are usually positioned in the center and both in the same row. The stations that are positioned on the main axis between the entry and the exit point have high utilization. In contrast, stations that are positioned on the outer parts of the layout have the lowest utilization. In Figure 2.13, we see that the entry and exit points are both in the center row and that station $L6$ on the border of the layout has a much lower utilization than the stations on the center axis. Finally, we observe that the majority of units of the same model are assembled along the same route. In the example, 18 out of the 21 units of model 1 are assembled along route $L2$-$L1$-$L4$-$L7$-$L8$-$L11$.

2.5.4 Efficiency comparison between flexible assembly layouts and line assembly layouts

Efficiency is a major performance indicator of a layout. It measures how well the resources in the layout are utilized. We use the classical efficiency measure for assembly lines as an efficiency indicator, that is, the efficiency equals the ratio of total workload

Table 2.5: Efficiency analysis for LALs in closed and open stations settings.

	Closed stations	Open stations		
	$\alpha = 0\%$	$\alpha = 5\%$	$\alpha = 15\%$	$\alpha = 33\%$
Av. efficiency of LALs	71.9%	74.2%	79.4%	88.7%
Av. gain in efficiency for FALs	24.5%	22.2%	17.0%	7.7%

to installed capacity and is equivalent to the average utilization of the stations. Let $l \in L^U$ be the set of locations used and u_l be the utilization of the station at location l, we calculate the efficiency of FALs and LALs using Equation (2.6).

$$\text{efficiency} = \frac{\sum\limits_{m \in M | t \in T_m} q_{m,t} \cdot d_m}{\tau \cdot |L^U|} = \frac{\sum\limits_{l \in L^U} u_l}{|L^U|} \tag{2.6}$$

Our analysis shows that FALs dominate LALs in terms of efficiency. Measured across all 528 instances, the average efficiency of FALs is 96.4%. In contrast, the average efficiency of LALs is only 71.9%. Consequently, we see an average gain in efficiency of 24.5% for FALs. The higher efficiency values of FALs are intuitive. Since the stations in FALs are neither paced nor coupled, stations are only occupied by a vehicle while tasks are being performed. Wasting station capacity due to a smaller workload than the cycle time is avoided. The vehicles only visit a station if the corresponding tasks are needed.

The analysis above applies to LALs with closed stations ($\alpha = 0\%$ in Constraints (A.2d) in Appendix A.2). When using closed stations, workers are not allowed to drift out of their stations, which means that the cycle time has to be respected for all models at each station. The efficiency disadvantage of LALs declines when using open stations ($\alpha > 0\%$), such that workers are allowed to drift out into downstream stations. As shown in Table 2.5, the average efficiency of LALs increases and the average gain in efficiency for FALs decreases with higher drift factors α on the line. When allowing a maximum excess of cycle time of $\alpha = 33\%$, we observe an average efficiency of LALs of 88.7% and an average gain in efficiency for FALs of only 7.7%. However, allowing workers to drift out of their stations implies several challenges. For example, workers might interfere with the operations of workers in downstream stations. Moreover, operational sequencing and part supply become more complex.

It should be noted that we are investigating efficiency on a strategic design level. Our efficiency indicator does not take into account blocking and starving of the stations that might occur in scheduling on the operational level. Also, it is not surprising that the

efficiency of FALs is never worse than for LALs, because the LAL solution is included in the solution space of the FALDP.

2.5.5 Effect of vehicle heterogeneity

Vehicle heterogeneity is the main driver that motivates OEMs to investigate FALs. As described earlier, OEMs experience inefficiencies when producing a heterogeneous set of vehicles simultaneously in the same LAL. The industry expects that FALs can alleviate these inefficiencies. We therefore elucidate the effect of vehicle heterogeneity on the efficiency of both FALs and LALs. For this purpose, we conduct a series of multiple linear regressions on our instance set using different dependent variables. The independent variables are always structure and task time heterogeneity. We employ the ordinary least squares (OLS) method to estimate regression parameters.

We first investigate the effect of structure and task time heterogeneity on the efficiency of FALs. Figure 2.14 and Table 2.6 show our regression results. In Figure 2.14, we depict projections of the regression plane from a structure and task time heterogeneity point of view respectively. The OLS regression results in Table 2.6 show that the efficiency of FALs is insensitive to vehicle heterogeneity. The regression coefficients of both structure and task time heterogeneity are insignificant. These results are supported by the negative adjusted coefficient of determination R^2_{adj}, which indicates that the independent variables structure and task time heterogeneity are not suitable for explaining the variation in the efficiency of FALs.

The efficiency of LALs, in contrast, is negatively affected by vehicle heterogeneity. Figure 2.15 and Table 2.7 show the regression results for LALs with closed stations ($\alpha = 0\%$). For low levels of vehicle heterogeneity, we observe higher values of efficiency than for high levels. For homogeneous vehicles, LALs with closed stations achieve efficiency values close to 80%. With increasing vehicle heterogeneity, the efficiency drops to values below 70%. The regression coefficients and corresponding p-values of both structure and task time heterogeneity indicate strong negative correlations that are significant on the 1% level. Based on R^2_{adj}, we conclude that structure and task time heterogeneity are two important determinants for the efficiency of LALs. More than 15% of the variation of the dependent variable is determined by these two independent variables.

The gain in efficiency for FALs is defined as the difference between the efficiencies of FALs and LALs. Figure 2.16 and Table 2.8 show our regression results when comparing FALs to LALs with closed stations ($\alpha = 0\%$). The results point out that the gain in efficiency for FALs depends on the level of vehicle heterogeneity. We observe efficiency

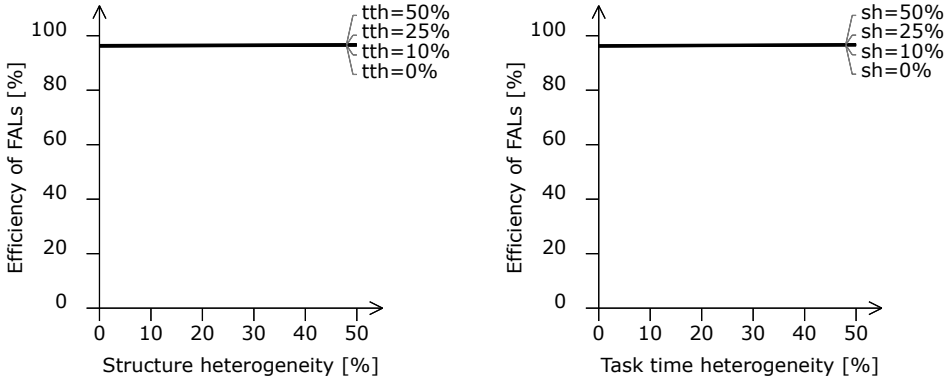


Figure 2.14: OLS regression results for efficiency of FALs.

Table 2.6: OLS regression results for efficiency of FALs.

	Coefficient	P-value	Significance
Constant	96.0891	0.0000	***
Structure heterogeneity	0.0067	0.4963	
Task time heterogeneity	0.0084	0.4462	
R^2	0.0017		
R^2_{adj}	−0.0021		

*** on 1% level, ** on 5% level, * on 10% level

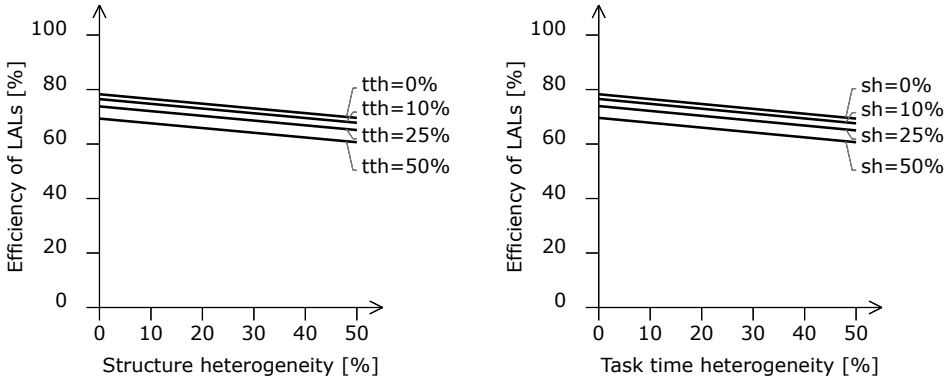


Figure 2.15: OLS regression results for efficiency of LALs with closed stations.

Table 2.7: OLS regression results for efficiency of LALs with closed stations.

	Coefficient	P-value	Significance
Constant	78.2790	0.0000	***
Structure heterogeneity	−0.1736	0.0000	***
Task time heterogeneity	−0.1788	0.0000	***
R^2	0.1623		
R^2_{adj}	0.1600		

*** on 1% level, ** on 5% level, * on 10% level

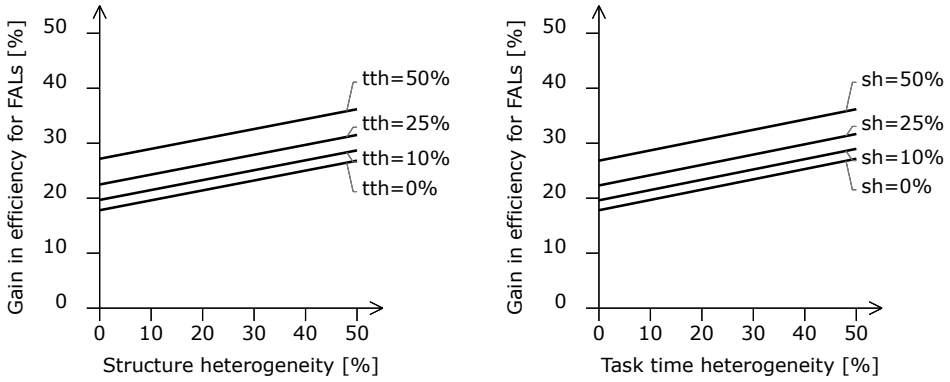


Figure 2.16: OLS regression results for gain in efficiency for FALs compared to LALs with closed stations.

Table 2.8: OLS regression results for gain in efficiency for FALs compared to LALs with closed stations.

	Coefficient	P-value	Significance
Constant	17.8100	0.0000	***
Structure heterogeneity	0.1803	0.0000	***
Task time heterogeneity	0.1872	0.0000	***
R^2	0.2023		
R^2_{adj}	0.1992		

*** on 1% level, ** on 5% level, * on 10% level

gains below 20% for homogeneous vehicles. With increasing structure and task time heterogeneity, the gain in efficiency for FALs increases substantially. For high structure and task time heterogeneity, we note efficiency gains of around 30%. The positive correlations are significant on the 1% level. Nearly 20% of the variation in efficiency gain is explained by structure and task time heterogeneity.

When comparing FALs to LALs with open stations ($\alpha > 0\%$), we find similar correlations just as for closed stations. While the efficiency of LALs is negatively affected by structure and task time heterogeneity, the gain in efficiency for FALs increases with higher levels of structure and task time heterogeneity. However, with larger α, we observe higher efficiency values for LALs and therefore lower gains in efficiency for FALs. The regression results shown in Table 2.9 point out that the gain in efficiency for FALs diminishes with higher drift factors α on the line, especially for homogeneous vehicles. For $\alpha = 33\%$, there are no significant gains in efficiency for FALs when assembling homogeneous vehicles.

Table 2.9: OLS regression results for gain in efficiency for FALs compared to LALs with closed and open stations.

	Closed stations	Open stations		
	$\alpha = 0\%$	$\alpha = 5\%$	$\alpha = 15\%$	$\alpha = 33\%$
Constant	17.8100 ***	14.4933 ***	7.9502 ***	1.1432
Structure heterogeneity	0.1803 ***	0.2144 ***	0.2631 ***	0.2011 ***
Task time heterogeneity	0.1872 ***	0.2076 ***	0.2276 ***	0.1542 ***

*** on 1% level, ** on 5% level, * on 10% level

2.6 Conclusion

In this chapter, we investigated the design and performance of FALs in the automotive assembly. We provided a formal representation of the FALDP and developed an exact as well as a matheuristic solution approach. Our computational analysis showed that our matheuristic is capable of finding high-quality solutions in acceptable time for instances of all sizes. The obtained FALs manifest two consistent design characteristics, i.e., compactness and centralization. Compactness means that FALs are typically not much longer than they are wide. Centralization means that the entry and exit points are usually positioned in the center and both are in the same row. The stations on the main axis between the entry and the exit point have the highest utilization, whereas stations on the outer parts of the layout have the lowest utilization. Also, the majority of units of the same model is typically assembled along the same route.

The comparison between FALs and LALs generated several valuable managerial insights. First, we showed that FALs have advantages in terms of efficiency. Compared to LALs with closed stations, the average gain in efficiency is 24.5%. This result is in line with estimations by Audi that predict efficiency gains of around 20%[12]. When LALs with open stations are used, such that workers are allowed to drift out of their stations, the gain in efficiency for FALs declines. Next, we showed that the efficiency of FALs is insensitive to vehicle heterogeneity. Conversely, the efficiency of LALs is negatively influenced by vehicle heterogeneity. In summary, FALs become more attractive with greater vehicle heterogeneity.

Our results are not only relevant for automotive OEMs. Many automotive components are characterized by high levels of heterogeneity. Therefore, first-tier suppliers might also benefit from a conversion to FALs. Moreover, other industries that are assembling heterogeneous products, e.g., helicopters, are starting to investigate FALs as well.

[12] https://www.handelsblatt.com/unternehmen/industrie/keine-fliessbaender-mehr-audi-plant-eine-revolution/14894190.html (published: 27/11/2016, retrieved: 09/12/2020)

The aim of this chapter is to demonstrate the capabilities of FALs. We therefore focused our performance analysis on efficiency. Additionally, other performance indicators deserve further attention. Due to the uniform workflow, LALs are expected to have benefits in terms of complexity. Planning and control are simple, because all vehicles pass through all stations in the same sequence and pace. In FALs, however, the exploitation of the routing and operation flexibility requires real-time scheduling. Due to the AGVs and buffers, FALs need more space and initial investment than LALs. A comprehensive analysis is therefore necessary to decide on the transition from LALs to FALs.

As shown in Figure 2.2, many decision problems need to be addressed when designing and operating FALs in the automotive assembly. At this point, we want to highlight the most important future research topics. In the FALDP, we excluded dynamics and uncertainty. In reality, automotive OEMs operate in a highly dynamic and stochastic market environment. The robust design of FALs is an important direction for future research. Especially demand is difficult to predict accurately. A two-stage stochastic FALDP formulation could be a way to address robustness. Stochastic task times could be incorporated by means of a sampling approach. Moreover, we focused on the initial design of FALs. The reconfiguration of FALs, however, is also interesting. LALs are known for their low reconfigurability. To introduce new models, react to demand shifts, or adjust the capacity, the entire line typically needs to be rebalanced while production is suspended. In contrast, FALs allow incremental adjustments without suspending production. Tasks can be reassigned or new stations can be added alongside the layout without affecting the operation of the existing stations. These reconfiguration flexibilities make FALs highly attractive for the automotive industry.

In our problem setting, the definition of the assembly segments was given. The problem of determining the best segmentation of the final assembly into LAL and FAL segments requires further research. Our results suggest that the segmentation should be based on the heterogeneity of the vehicles.

Another relevant research direction is to address lower-level planning problems. New approaches and algorithms are needed for master production scheduling, scheduling, and rescheduling in FALs.

3 Configuration of flexible assembly layouts for the automotive assembly

This chapter is based on an article submitted as:

Hottenrott, A., Schiffer, M., & Grunow, M. (2020). IoT-driven manufacturing in the automotive industry: An impact assessment of flexible assembly layouts. *Submitted for publication.*

Abstract

Automotive manufacturers take IoT-driven manufacturing to an unprecedented level, considering the deployment of FALs in which AGVs transport bodyworks on individual routes between assembly stations. To this end, a methodological framework that allows to assess the impact of technology choice decisions between traditional LALs and FALs as well as the impact of different flexibility levers and operational policies is necessary for optimal decision support. We provide such a framework based on an analytical analysis and a chance-constrained problem formulation. We further show how this problem formulation can be solved optimally using a tailored B&P algorithm.

Our results quantify the impact of different operational policies in FALs. We show that flexibility enables a simultaneous improvement in worker utilization and WIP, resolving a classical trade-off in manufacturing systems. Moreover, we find that worker utilization and output are up to 30% higher in FALs compared to LALs. Further, FALs prove to be especially beneficial during the ramp-up of vehicles with alternative drivetrain technologies, such as the current transition to electric vehicles.

3.1 Introduction

The IoT is seeping into various industries, enabled through major enhancements in robotics and communication technologies, big data processing, and artificial intelli-

gence (Olsen & Tomlin, 2020). In this context, the automotive industry introduces self-controlled, robotized facilities to improve flexibility and increase production efficiency. Major players, such as Audi, Volkswagen, and Tesla, even consider a precedent break with assembly line production[13], which has been the status quo in this industry for the past century. They design FALs in which AGVs transport bodyworks on individual routes between assembly stations[14]. These new layouts allow for improved handling of vehicle heterogeneity compared to conventional LALs, resulting in higher worker utilization at the stations. FALs can be designed with different degrees of flexibility, which bear different technological implementation challenges and levels of complexity in operational planning and control. Accordingly, technology selection and operational policy determination decisions reach a new level of complexity and are crucial for successful operations.

FALs invoke a paradigm shift in today's automotive manufacturing systems and reveal an additional perspective on technology-driven flexibility. It is well-known that manufacturing flexibility enables better performance, especially in dynamic environments (Anand & Ward, 2004). In operations management, however, flexibility has so far mostly been studied from a macroscopic supply chain perspective, e.g., by focusing on sourcing flexibility (see, e.g., Graves & Tomlin, 2003), or from mesoscopic perspectives, e.g., by analyzing the impact of flexibility on inventory levels (see, e.g., Simchi-Levi, Wang, & Wei, 2018). All of these research streams make the assumption that the production at operational level can be treated as a black box model, and do not capture recent flexibility improvements that result from IoT-driven manufacturing.

Automotive manufacturers optimized LALs for increasing vehicle heterogeneity over the past thirty years. With sequencing techniques, overcapacities, and utility workers, manufacturers managed to efficiently produce heterogeneous vehicles in an LAL. However, up to now, configuration options (e.g., sunroof or seat heating) remained the sole heterogeneity drivers that resulted in divergent tasks and workloads. While conventional measures were sufficient to counteract this heterogeneity, the current diffusion of alternative drivetrain technologies challenges this status quo, because the assembly of such vehicles involves significantly different tasks, tools, and worker qualifications, e.g., the battery assembly for electric vehicles differs completely from the assembly of internal combustion engines, and different safety standards apply[15].

[13] https://www.audi-mediacenter.com/en/audi-techday-smart-factory-7076/modular-assembly-7078 (published: 27/11/2016, retrieved: 09/12/2020)

[14] https://www.bcg.com/de-de/publications/2018/flexible-cell-manufacturing-revolutionize-carmaking.aspx (published: 08/10/2018, retrieved: 09/12/2020)

[15] https://www.strategyand.pwc.com/de/de/studien/2018/transforming-vehicle-production/transforming-vehicle-production.pdf (published: 09/10/2018, retrieved: 09/12/2020)

While dedicated LALs for different drivetrain technologies appear to be an obvious solution, automotive manufacturers aim for a joint assembly, as the heterogeneity involved affects only a few assembly segments. Moreover, alternative drivetrains constitute ramp-up technologies in an uncertain market. Investing into dedicated LALs bears a high financial risk as current production volumes for electric vehicles are too low for efficient operations, and a market shift towards yet another technology (e.g., from battery-powered to hydrogen-powered) may cause significant sunk cost. However, producing vehicles with different drivetrain technologies in an LAL challenges its design and operational performance, e.g., it may cause significantly reduced utilization.

To discuss these challenges, we presuppose some basic knowledge of the automotive assembly process (cf. Section 3.3) and introduce the following minimal case: we subdivide the assembly into multiple segments, between which the handover of the vehicles is paced by a *cycle time* c. We consider the production of two vehicles $V1$ and $V2$ that subsequently arrive in a segment with two stations $L1$ and $L2$. Processing each vehicle requires two tasks A and B, of which the former is performed on $L1$ and the latter on $L2$. Performing a task takes q_{1A}, q_{1B} time units for vehicle $V1$ and q_{2A}, q_{2B} time units for vehicle $V2$. $V1$ arrives in the segment at time 0 and $V2$ at time $0 + c$.

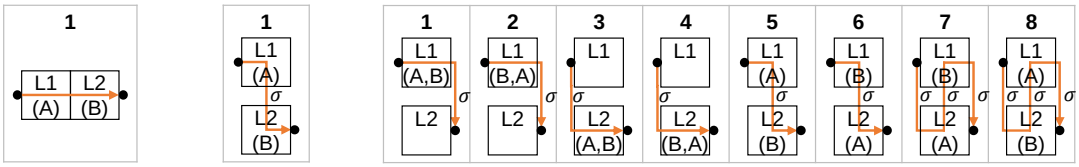
Figure 3.1a shows the only production strategy for the simplest case, a conventional LAL. In this paced, coupled setting, line stoppages occur whenever the workload at a station exceeds the cycle time, and idle times occur whenever the workload at a station is below the cycle time. Apparently, these imbalances may deteriorate the LAL's utilization and output level.

3.1.1 Benefits of flexible assembly layouts

In FALs, the abovementioned problems do not occur, because the workflow is not paced by a cycle time. Instead, AGVs transport the vehicles on individual routes between stations, which allows for variable production strategies. However, transportation between stations is time-consuming and vehicles may have to wait at occupied stations. Accordingly, we expect that FALs have advantages in utilization but disadvantages in WIP compared to LALs.

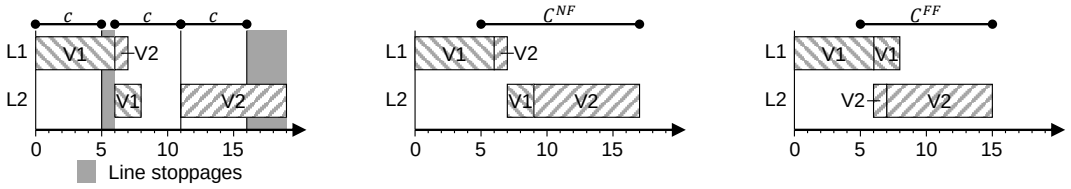
We now formally analyze our conjectures regarding the benefits and disadvantages of FALs and their operational policies. This analysis bases on the minimal case introduced above, which we extend as follows: we consider an FAL in which transportation between stations lasts σ time units.

Since the vehicle sequence remains unaltered, all vehicles have the same *segment cycle time* C, i.e., spend the same amount of time in the assembly segment. Figure 3.1b shows



(a) LAL. **(b)** FAL (NF). **(c)** FAL (FF).

Figure 3.1: Possible production strategies in the minimal case for a conventional LAL, an FAL without flexibility (NF), and an FAL with full flexibility (FF).



(a) LAL. **(b)** FAL (NF). **(c)** FAL (FF).

Figure 3.2: Layout-dependent schedules for the example instance.

the production strategy for an FAL without flexibility (NF), where both vehicles must pass through $L1$ and $L2$ subsequently. Figure 3.1c shows potential production strategies for an FAL with full flexibility (FF), i.e., a production system where vehicles $V1$, $V2$ can receive tasks A, B in arbitrary order (which we refer to as *operation flexibility*), and where both tasks A, B can be performed on each station $L1$, $L2$ (which we refer to as *routing flexibility*). As can be seen, there exists only a single production strategy for the inflexible configuration, while we can choose between eight production strategies for the fully flexible configuration.

We now use an example instance with $q_{1A} = 6$, $q_{1B} = 2$, $q_{2A} = 1$, $q_{2B} = 8$, $c = 5$, and $\sigma = 1$ to illustrate the differences between the three layouts. Figure 3.2a shows the schedule for an LAL with closed stations. Here, line stoppages occur in the first and third cycle as the workload at a station exceeds the cycle time. Idle times occur on both stations in the second cycle as the workloads are below the cycle time. In practice, one could add more stations to reduce line stoppages, but this inevitably increases the WIP. Also, allowing workers to drift into subsequent stations could reduce the utilization deterioration. However, the more drifting is allowed, the more the workers interfere.

Figures 3.2b and 3.2c show the optimal schedules for the FAL (NF) and the FAL (FF) respectively. Comparing the LAL to the FAL (NF), the LAL with two stations naturally requires a WIP of two, while the FAL (NF) requires a 20% higher WIP. While the LAL requires two workers (one for each station) for 19 time units, the FAL (NF) requires

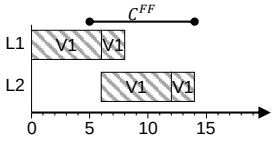


Figure 3.3: FAL (FF) schedule for an alternative vehicle sequence.

two workers for 17 time units, which equals a utilization improvement of 12% as task times remain equal. Accordingly, an FAL without flexibility can already yield utilization improvements compared to a paced LAL at the price of a higher WIP. Comparing the FAL (NF) to the FAL (FF), we observe that the minimum required segment cycle time is reduced by 17% from $C^{NF} = 12$ to $C^{FF} = 10$ and note that the WIP is proportional to C (cf. Section 3.3.1), which implies an equal reduction. Moreover, the FAL (FF) allows for a utilization improvement of 13%.

Theorem 1 provides generalized ranges for WIP decreases and utilization improvements that can result from flexibility in FALs in our minimal case, independent of the underlying instance.

Theorem 1 *Let NF, FF be two configurations of an instance of Problem 1 (cf. Section 3.3), each containing a single vehicle sequence with two vehicles $V1$, $V2$. Let $L1$, $L2$ be the available stations, with A, B being the tasks that must be processed for each vehicle. Further, available production strategies hold as depicted in Figure 3.1b for configuration NF and as depicted in Figure 3.1c for configuration FF. Then, for any instance it holds that*

$$\frac{WIP^{FF}}{WIP^{NF}} \in [0.5; 1.0] \quad \textit{and} \quad \frac{U^{FF}}{U^{NF}} \in [1.0; 2.0],$$

with WIP^X being the WIP of configuration X and U^X being the respective utilization.

We refer to Appendix B.1 for a proof of Theorem 1. Remarkably, Theorem 1 indicates that flexibility in FALs may improve WIP and utilization simultaneously, which generally constitutes a trade-off in conventional assembly systems.

Figure 3.3 shows the optimal schedule for the FAL (FF) when producing a sequence of two consecutive vehicles $V1$. Keeping the FAL feasible for this sequence requires a minimum segment cycle time of $C^{FF} = 9$, which is one time unit shorter compared to the sequence shown in Figure 3.2c. This highlights the importance of considering manifold vehicle sequences when deciding on the segment cycle time of an FAL. However, these sequences are unknown at the time of system configuration. Indeed, the segment cycle time constitutes an additional flexibility lever, as an increased segment cycle time enables feasibility for a larger variety of vehicle sequences at the expense of a larger WIP.

The minimal example and the analytical findings of Theorem 1 indicate that FALs may offer a significant improvement potential for automotive manufacturing, and recent developments in practice confirm the manufacturers' trust in this concept: Toyota spends major investments in increasing the flexibility of their assembly systems[16]. Tesla already uses AGVs in their production site in Fremont, California[17]. Also, Audi launched an AGV-based production in Győr, Hungary[18].

However, it remains an open question to which extend FALs can improve the efficiency of automotive manufacturing for complex settings that comprise manifold (uncertain) vehicle manufacturing sequences. As quantifying improvement potentials in closed analytical form remains intractable for such settings, the remainder of this chapter focuses on deriving an algorithmic framework that allows to solve such complex settings and provides an extensive numerical study to analyze the benefits of FALs in real-world environments.

3.1.2 Contributions

Specifically, our contributions are fivefold. First, we show analytically that even for a minimal case, the selection of operational policies in FALs can have a significant impact on operational performance. Second, we present a chance-constrained integer program that formalizes the flexibility configuration problem in FALs, which minimizes the segment cycle time for a multitude of potential sequences. This formulation covers all operational policies, i.e., all possible combinations of the operation and routing flexibility levers. Third, we show how this problem can be decomposed into deterministic subproblems, and we develop a B&P framework to solve these subproblems. Fourth, we apply this framework to an extensive computational study in order to evaluate the impact of FALs in automotive manufacturing. Here, we are the first to develop a set of realistic test instances for this new problem, based on expert knowledge from industry. Fifth, we provide managerial insights on configuration options for different flexibility levers in FALs by quantifying their effect on operational performance. We find that increasing flexibility allows to improve utilization at lower WIP levels. Moreover, we compare the performance of FALs to conventional LALs and show that FALs improve utilization and output levels by up to 30%. Finally, we find that FALs are especially

[16] https://www.bloomberg.com/news/articles/2019-10-03/toyota-revamps-its-biggest-car-plant-for-hybrid-suvs (published: 03/10/2019, retrieved: 09/12/2020)
[17] https://www.wired.co.uk/gallery/tesla-factory-fremont-tour-photos-pictures (published: 20/07/2017, retrieved: 09/12/2020)
[18] https://www.automotivelogistics.media/audi-starts-electric-motor-production-at-győr/21237.article (published: 25/07/2018, retrieved: 09/12/2020)

beneficial during the upcoming ramp-up of alternative drivetrain technologies, where they can be adjusted easily to different demand mixes.

3.1.3 Organization

The remainder of this chapter is organized as follows: We review related literature in Section 3.2. In Section 3.3, we introduce our problem setting, while Section 3.4 presents our methodology. We then detail our design of experiments in Section 3.5 and discuss results in Section 3.6. Section 3.7 concludes the chapter by summarizing its main insights.

3.2 Literature review

In the following, we concisely review related literature. We first focus on general flexibility studies, before we review related operational planning models.

Different levers of flexibility have been studied from a strategic perspective, e.g., analyzing the impact of sourcing flexibility in order to increase supply chain robustness towards demand uncertainty (see, e.g., Graves & Tomlin, 2003; Hopp, Iravani, & Xu, 2010; Jordan & Graves, 1995; Muriel, Somasundaram, & Zhang, 2006). Studying flexibility through the lens of newsvendor networks showed that inventory management and allocation strategies also allow to mitigate demand uncertainty (see, e.g., Bassamboo, Randhawa, & van Mieghem, 2010; Tomlin & Wang, 2005; van Mieghem, 2007; van Mieghem & Rudi, 2002). Further works focused particularly on the placement of safety stocks in supply chains and support these findings (see, e.g., Graves & Willems, 2000; Humair & Willems, 2006). Overall, these works provide profound insights at strategic level and one may identify parallels at operational level as demand uncertainty remains the influencing factor on both levels and sourcing flexibility constitutes the strategic counterpart of exploiting task duplicates at operational level. However, this research cannot be extended to analyze FALs, because it does not account for operational policy and system structure decisions, which are necessary to capture all operational interdependencies in order to allow for a thorough anaylsis.

From an operational perspective, specific flexibility levers have been included into standard scheduling problems. Sawik (2012) focused on simultaneous balancing and cyclic sequencing in LALs that comprise task duplicates, which have also been studied from a design perspective (Bard, 1989; Y. Bukchin & Rabinowitch, 2006). Some job shop scheduling variants consider flexible task sequences (see, e.g., Mohammadi, Karampourhaghghi, & Samaei, 2012), task duplicates (see, e.g., Mastrolilli & Gambardella, 2000), or both (see, e.g., Zhang & Wong, 2015). These works cannot be applied to study

the benefits of FALs, because they either focus on single flexibility levers or propose heuristic algorithms that cannot be used for comparative analyses between different configurations. Moreover, the investigated job shop variants lack elementary requirements to model FALs in the automotive assembly, e.g., transportation times between stations.

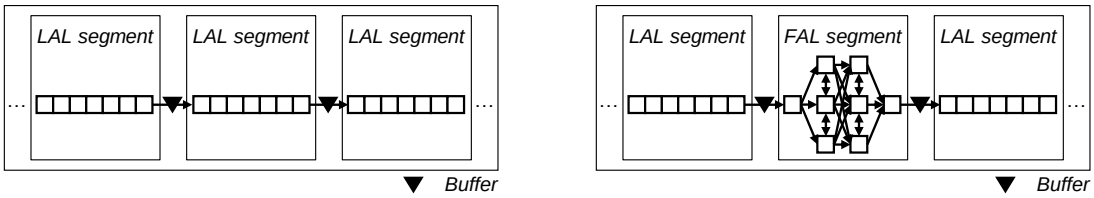
Concluding, the importance of studying flexibility in supply chains became evident through strategic analyses. However, studies on flexibility from an operational perspective are still scarce and focus on single flexibility levers from a technical viewpoint. While this might be plausible for conventional assembly systems, studying FALs requires a generic analysis of all core levers of flexibility at operational level, their interaction, and their implications for overall (comparative) system evaluation. However, no study on these effects exists so far as only a single publication studied FALs but focused exclusively on their strategic design (Hottenrott & Grunow, 2019).

3.3 Problem setting

Up to now, automotive manufacturers organized their final assembly in a pure assembly line layout (cf. Figure 3.4a), i.e., a serial arrangement of workstations, at which different tasks are performed in a predefined order. Hereby, all vehicles pass through all stations, and the workflow is paced by a cycle time. To limit adverse effects of disruptions, such an assembly line consists of multiple segments, decoupled by small buffers. In a mixed layout (cf. Figure 3.4b), an FAL replaces the LAL in one or multiple segments of the original assembly line. Here, stations are neither arranged serially nor is the workflow paced. Instead, AGVs transport vehicles between stations, each taking a unique route relating to its specifications. For example, electric vehicles visit the battery assembly station, whereas conventional vehicles bypass it. Both concepts offer different degrees of flexibility.

Line assembly layouts: LALs allow for limited flexibility, which mainly remains at the strategic design level. The manufacturer decides on the number of stations and the assignment of tasks to stations. Since the layout design predefines the vehicles' assembly schedules, hardly any flexibility exists at tactical and operational level.

Flexible assembly layouts: FALs possess similar degrees of flexibility as LALs during strategic design, as the manufacturer again decides on the number of stations and the assignment of tasks to stations. Moreover, the manufacturer sets the stations' locations



(a) Pure line layout. **(b)** Mixed layout with an FAL segment.

Figure 3.4: Schematic example of a pure line layout and a mixed layout with an FAL segment.

and the number of task duplicates. Two additional flexibility levers exist at lower planning levels: modifying a task sequence (operation flexibility) and performing a task at different stations (routing flexibility).

Apparently, both layout concepts show different structural properties, which lead to advantages and disadvantages. In Section 3.3.1, we discuss these structural properties and a manufacturer's tactical planning problem, before we introduce a formal problem definition in Section 3.3.2.

3.3.1 Structural properties and tactical decision making

A fundamental difference exists between LALs and FALs in the worker-to-vehicle relation: in LALs workers wait for vehicles to arrive at their station; in FALs vehicles wait at stations for workers to become available. Accordingly, FALs may better adapt to varying arrival sequences of vehicles, especially for heterogeneous vehicles from changing demand mixes. This improved adaptability allows for a higher utilization and output levels but also results in a higher WIP, incurred by waiting vehicles. While a higher WIP may not necessarily be seen as a disadvantage in such a setting but rather as an additional flexibility lever, incontestable disadvantages remain with respect to *i*) worker confusion, i.e., workers facing different assembly states when performing a task on different vehicles due to missing standard task sequences, and *ii*) complicated material supply due to real-time routing, i.e., JIS stocking is no longer possible such that part kits have to be prepared in advance to be transported together with a vehicle on its AGV.

Obviously, FALs increase the complexity of planning and controlling manufacturing operations. Hence, automotive manufacturers consider to combine FALs and LALs in their final assembly, hereby including FALs in-between LAL segments (cf. Figure 3.4b) and only for segments with a high vehicle-dependent task heterogeneity. Studying such a setting remains the focus of this chapter.

Production planning usually follows a hierarchical structure, i.e., one first takes high-level, strategic decisions, anticipating their impact on lower planning levels. In our studies, we use the results from Hottenrott and Grunow (2019) to account for strategic decisions and focus on tactical planning which is central to exploit the abovementioned flexibility levers. Here, an automotive manufacturer faces a flexibility configuration problem, i.e., the manufacturer decides on an appropriate WIP target for an FAL segment and on the exploitation of operation and routing flexibility. The WIP target itself is an additional flexibility lever in FAL segments, since, similar to operation and routing flexibility, a higher WIP facilitates scheduling on the operational level.

Deciding on a WIP target is equivalent to setting a vehicle makespan in an FAL segment. Every vehicle spends an equal amount of time in the FAL segment as it is surrounded by up- and downstream LAL segments and the vehicle sequence remains unaltered. We refer to this time span as the segment cycle time C. Due to the surrounding LAL segments, vehicles arrive and leave the FAL segment in constant increments of the cycle time c, such that the WIP in an FAL segment is proportional to the segment cycle time, formally, WIP$= C/c$.

In this context, an automotive manufacturer faces a trade-off between feasibility for many vehicle sequences and a small segment cycle time, which translates into a low WIP, i.e., among others lower investments into AGVs and simplified system control. Obviously, the manufacturer must take this decision before the vehicle sequences at operational level are known. Hence, a high-quality approximation of the unknown vehicle sequences and a mechanism to balance between feasibility and efficiency for (a subset of) these sequences is crucial for this planning task. Technically, the manufacturer faces a chance-constrained optimization problem which can be stated in general form as

$$\min\ f(x,\xi) \tag{3.1a}$$

s.t.

$$g(x,\xi) = 0 \tag{3.1b}$$

$$\mathcal{P}(h(x,\xi) \geq 0) \geq p \tag{3.1c}$$

where f is the objective function, $g(x,\xi)$ is a function of equality constraints, $h(x,\xi)$ is a function of inequality constraints, x is the decision vector, and ξ remains the uncertainty vector. We then seek to minimize the objective function such that the inequality constraints are valid with a probability $p \in [0,1]$ for a specified ξ, formally, $\mathcal{P}(h(x,\xi) \geq 0) \geq p$.

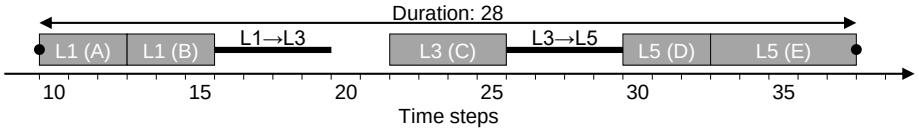


Figure 3.5: Example of a route with five tasks ($\mathcal{I}_{sv} = \{A, B, C, D, E\}$) on three stations ($\mathcal{L} = \{L1, L3, L5\}$).

3.3.2 Problem definition

We now define our specific planning problem as a chance-constrained integer program. This optimization problem aims to minimize the segment cycle time $C(\xi)$ for a set of unknown vehicle sequences $\mathcal{S}(\xi)$, which must be produced on a set of stations $l \in \mathcal{L}$ over a discretized time horizon with time steps $t \in \mathcal{T}$. Here, a sequence $s \in \mathcal{S}(\xi)$ is a list of vehicles that arrive in an assembly segment in a predefined order. Accordingly, the solution to our problem consists of schedules Π_s, each encoding the assemblies for all vehicles $v \in \mathcal{V}_s$ of a sequence $s \in \mathcal{S}(\xi)$. Then, a schedule $\Pi_s = (r_1, ..., r_{|\mathcal{V}_s|})$ is a $|\mathcal{V}_s|$-tuple, which contains one route $r_v \in \mathcal{R}_{sv}$ for every vehicle, with $\mathcal{R}_{sv}$ being the set of feasible routes for vehicle v in sequence s. Each route encodes a sequence of the required tasks $i \in \mathcal{I}_{sv}$ and allocates each task to a station. Moreover, a route includes the start and end times of tasks at stations. Thus, each route has a specific duration w_{svr}, defined as the sum of all processing, transportation, and waiting times. The binary parameter b_{svrlt} indicates if route r for vehicle v in sequence s occupies station l at time t. Figure 3.5 illustrates such a route.

With the notion of a feasible route, we imply that *i*) the route covers all mandatory tasks to process a vehicle, *ii*) tasks are allocated to stations where they can be conducted, *iii*) precedence relations between tasks are respected, *iv*) the time between subsequent tasks is sufficient to move a vehicle from one station to another if required, *v*) stations are visited at most once to avoid worker confusion, and *vi*) the route follows a directed flow, i.e., it does not include backward transfers to upstream stations in order to reduce AGV traffic and attenuate the risk of collisions.

We state our chance-constrained integer program using binaries X_{svr} to denote whether route r is used to process vehicle v in sequence s ($X_{svr} = 1$) or not ($X_{svr} = 0$), and continuous variables C_s to indicate the minimum required segment cycle time for sequence s to be feasible.

Problem 1

$$\min\ C(\xi) \tag{3.2a}$$

s.t.

$$\sum_{r\in\mathcal{R}_{sv}} X_{svr} = 1 \qquad \forall s\in\mathcal{S}(\xi), v\in\mathcal{V}_s \tag{3.2b}$$

$$\sum_{v\in\mathcal{V}_s}\sum_{r\in\mathcal{R}_{sv}} b_{svrlt}X_{svr} \leq 1 \qquad \forall s\in\mathcal{S}(\xi), l\in\mathcal{L}, t\in\mathcal{T} \tag{3.2c}$$

$$C_s \geq \sum_{r\in\mathcal{R}_{sv}} w_{svr}X_{svr} \qquad \forall s\in\mathcal{S}(\xi), v\in\mathcal{V}_s \tag{3.2d}$$

$$\mathcal{P}\Big(C(\xi) \geq C_s\Big) \geq 1-\rho \qquad \forall s\in\mathcal{S}(\xi) \tag{3.2e}$$

$$X_{svr} \in \{0,1\} \qquad \forall s\in\mathcal{S}(\xi), v\in\mathcal{V}_s, r\in\mathcal{R}_{sv} \tag{3.2f}$$

Objective (3.2a) minimizes the segment cycle time. Constraints (3.2b) select exactly one route for each vehicle in all sequences. Constraints (3.2c) forbid that two vehicles in a sequence occupy the same station at the same time. In Constraints (3.2d), we derive the minimum required segment cycle times for all sequences. Constraints (3.2e) represent our chance constraints, ensuring that the overall segment cycle time is feasible for a predefined share $p = 1-\rho$ of all sequences $s\in\mathcal{S}(\xi)$, from here on referred to as the *feasibility target*. Finally, Constraints (3.2f) state the binary variable domain.

Two comments on this modeling approach are in order. First, we consider deterministic processing and transportation times, and incorporate neither breaks, nor maintenance, nor breakdowns at stations. This deterministic setting is status quo for tactical configuration problems of assembly lines. It can be readily applied to a tactical FAL configuration problem, because a robust AGV routing can exploit the system's flexibility to resolve any disorder at a lower level. Second, we forbid task preemption, which reflects current practice in the automotive assembly.

3.4 Methodology

In this section, we first show how Problem 1 can be decomposed into deterministic suproblems (Section 3.4.1), before we develop a B&P framework to solve each subproblem (Section 3.4.2).

3.4.1 Problem decomposition

The computational tractability of Problem 1 depends on the characteristics of the considered uncertainty, i.e., whether it is possible to decouple decisions from random variables

such that one can transform probabilistic to deterministic constraints, e.g., via probability density functions.

Our problem resembles a chance-constrained optimization problem as an automotive manufacturer aims for a segment cycle time that is feasible for a large share but not for the complete distribution of vehicle sequences. However, it diverges from standard chance-constrained optimization problems as the uncertainty affects the set of vehicle sequences $\mathcal{S}(\xi)$, and thus only indirectly affects w_{svr}, which implies decoupled variables. Accordingly, one may study the problem's sample counterpart, which becomes computationally tractable as it bears only a finite number of constraints.

We now present a sampling-based decomposition, which ensures that the segment cycle time is above a certain threshold to fulfill a predefined feasibility target of $1-\rho$. We introduce binary variables Y_s which indicate whether the minimum required segment cycle time C_s for sequence s is in the lower $1-\rho$ percentile of all sample sequences ($Y_s = 1$) or not ($Y_s = 0$). Further, we refer to $\mathcal{S}$ as the sample set of vehicle sequences and reformulate Problem 1 as follows:

Problem 2

$$\min\ C \tag{3.3a}$$

s.t.

$$\sum_{r\in\mathcal{R}_{sv}} X_{svr} = 1 \qquad \forall s\in\mathcal{S}, v\in\mathcal{V}_s \tag{3.3b}$$

$$\sum_{v\in\mathcal{V}_s}\sum_{r\in\mathcal{R}_{sv}} b_{svrlt}X_{svr} \leq 1 \qquad \forall s\in\mathcal{S}, l\in\mathcal{L}, t\in\mathcal{T} \tag{3.3c}$$

$$C_s \geq \sum_{r\in\mathcal{R}_{sv}} w_{svr}X_{svr} \qquad \forall s\in\mathcal{S}, v\in\mathcal{V}_s \tag{3.3d}$$

$$\sum_{s\in\mathcal{S}} Y_s \geq (1-\rho)|\mathcal{S}| \tag{3.3e}$$

$$C \geq C_s - (1-Y_s)|\mathcal{T}| \qquad \forall s\in\mathcal{S} \tag{3.3f}$$

$$X_{svr} \in \{0,1\} \qquad \forall s\in\mathcal{S}, v\in\mathcal{V}_s, r\in\mathcal{R}_{sv} \tag{3.3g}$$

$$Y_s \in \{0,1\} \qquad \forall s\in\mathcal{S} \tag{3.3h}$$

The functionalities of Objective (3.3a) and Constraints (3.3b) - (3.3d) match their counterparts in Problem 1. We transform the chance constraints (Constraints (3.2e)) to deterministic constraints, in which we select the feasible sequences in Constraint (3.3e) and obtain the segment cycle time C in Constraints (3.3f). Finally, Constraints (3.3g) - (3.3h) define the binary variable domains.

Problem 2 has a typical min-max objective and shows a distinct block structure in which every sample sequence forms a specific block. Only Constraint (3.3e) links these different blocks. This allows us to decompose Problem 2 into $|\mathcal{S}|$ smaller subproblems, one for each sample sequence s.

Problem 3

$$\min\ C_s \tag{3.4a}$$

s.t.

$$\sum_{r\in\mathcal{R}_{sv}} X_{svr} = 1 \qquad \forall v\in\mathcal{V}_s \tag{3.4b}$$

$$\sum_{v\in\mathcal{V}_s}\sum_{r\in\mathcal{R}_{sv}} b_{svrlt}X_{svr} \leq 1 \qquad \forall l\in\mathcal{L}, t\in\mathcal{T} \tag{3.4c}$$

$$C_s \geq \sum_{r\in\mathcal{R}_{sv}} w_{svr}X_{svr} \qquad \forall v\in\mathcal{V}_s \tag{3.4d}$$

$$X_{svr} \in \{0,1\} \qquad \forall v\in\mathcal{V}_s, r\in\mathcal{R}_{sv} \tag{3.4e}$$

Objective (3.4a) minimizes the required segment cycle time C_s for the considered sequence. We select one route for each vehicle (Constraints (3.4b)) and ensure that no station is occupied by two vehicles at the same time (Constraints (3.4c)). In Constraints (3.4d), we derive the objective value, and Constraints (3.4e) state the binary variable domain.

We can solve Problem 2 by solving Problem 3 for all sample sequences and sorting all C_s in increasing order. Then, the overall segment cycle time C corresponds to the C_s at position $\lceil(1-\rho)|\mathcal{S}|\rceil$ of the sorted values. This reduces the problem's complexity significantly, but even Problem 3 remains NP-hard (cf. Appendix B.2) and computationally intractable for instances of practical interest. In the following, we develop a B&P framework that resolves this intractability.

3.4.2 Branch-and-price framework

For our discussion, we assume that the interested reader is familiar with the general concept of B&P, where we integrate column generation into a B&B algorithm.

In a nutshell, we start solving a restricted master problem (RMP) (Section 3.4.2.1), i.e., the linear programming (LP) relaxation of our original problem with a limited set of columns. We then iterate between this RMP and pricing problems (Section 3.4.2.2) to improve the solution by adding additional columns with negative reduced cost, or to proof optimality in case no more such columns exist. Then, we apply branching

(Section 3.4.2.3) to obtain an integer solution and terminate when we found the optimal integer solution. In the following, we explain the mentioned algorithmic components and embed the resulting B&P algorithm into a framework that exploits tight upper bounds in order to speed up computational times (Section 3.4.2.4).

3.4.2.1 Restricted master problem

The RMP results from the LP relaxation of Problem 3.

Problem 4

$$\min \; C_s \tag{3.5a}$$

s.t.

$$\sum_{r \in \bar{\mathcal{R}}_{sv}} X_{svr} + Z_{sv} = 1 \qquad \forall v \in \mathcal{V}_s \tag{3.5b}$$

$$\sum_{v \in \mathcal{V}_s} \sum_{r \in \bar{\mathcal{R}}_{sv}} b_{svrlt} X_{svr} \leq 1 \qquad \forall l \in \mathcal{L}, t \in \mathcal{T} \tag{3.5c}$$

$$C_s \geq \sum_{r \in \bar{\mathcal{R}}_{sv}} w_{svr} X_{svr} + \phi \sum_{v_2 \in \mathcal{V}_s} Z_{sv_2} \qquad \forall v \in \mathcal{V}_s \tag{3.5d}$$

$$X_{svr} \geq 0 \qquad \forall v \in \mathcal{V}_s, r \in \bar{\mathcal{R}}_{sv} \tag{3.5e}$$

$$Z_{sv} \geq 0 \qquad \forall v \in \mathcal{V}_s \tag{3.5f}$$

Here, we consider only subsets of routes $\bar{\mathcal{R}}_{sv} \subseteq \mathcal{R}_{sv}$, i.e., a reduced set of columns. We extend the convexity constraints (Constraints (3.5b)) by dummy variables Z_{sv} to ensure feasibility in case $\bar{\mathcal{R}}_{sv}$ precludes a feasible solution, e.g., if the available routes of all vehicles occupy the same station at the same time. In this case, the dummy variables allow to find a feasible solution, although Constraints (3.5b) and (3.5c) would be conflicting. We penalize the usage of dummy variables in Constraints (3.5d) with a sufficiently high cost term ϕ to ensure convergence towards a feasible solution. Constraints (3.5e) - (3.5f) state the variable domains. Here, Constraints (3.5e) differ from Constraints (3.4e) by relaxing the integrality of the X_{svr} variables.

Solving this RMP, we obtain the dual multipliers of its constraints, which we use in the pricing problems to generate routes with negative reduced cost that iteratively enlarge $\bar{\mathcal{R}}_{sv}$. Specifically, we obtain a cost parameter β_{lt} for occupying station l at time t from (3.5c); a cost per time step γ_v for each vehicle from (3.5d); and a maximum cost threshold for promising routes κ_v for each vehicle from (3.5b), such that a route has negative reduced cost if its cost are below κ_v.

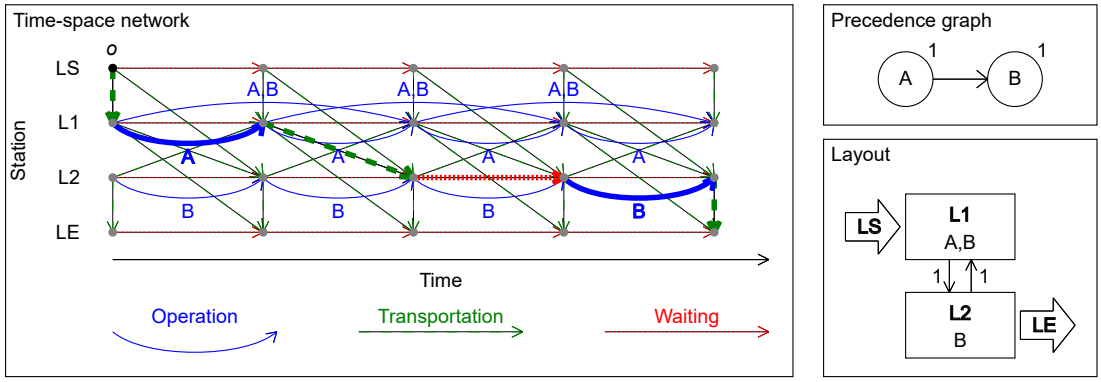


Figure 3.6: Example of a time-space network for a simplified example instance.

3.4.2.2 Pricing problems

We solve the pricing problem for each vehicle as an elementary shortest path problem with resource constraints (ESPPRC), modeled on a time-space network, formalized as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ in which nodes $n \in \mathcal{N}$ represent time-station combinations (t, l). We add an initial (LS) and a final (LE) dummy station that constitute the entry and exit points of the FAL, and we define an origin node $o = (t^0, LS)$ as a tuple of the vehicle's arrival time t^0 and station LS. To model the network, we use three types of arcs $a \in \mathcal{A}$: *i) operation arcs* expand in the time dimension and their length depends on the processing times of included tasks, *ii) waiting arcs* also expand in the time dimension and have a length of one time step, and *iii) transportation arcs* can expand in both the time and the station dimension. Then, a route is a sequence of consecutive arcs, and we solve a pricing problem for each vehicle to find a route with negative reduced cost that covers all tasks $i \in \mathcal{I}_{sv}$, respects task precedences, and fulfills the AGV's flow restrictions.

Figure 3.6 shows an example of such a network, its underlying precedence graph, and its layout for a simplified setting with two tasks (A, B) that must be processed in sequence, and two stations ($L1$, $L2$) that can perform either both tasks ($L1$) or solely task B ($L2$). We highlight a potential route that consists of a transportation arc from LS to $L1$, followed by an operation arc at $L1$ to perform task A, a transportation arc to $L2$, a waiting arc at $L2$, an operation arc at $L2$ to perform task B, and a transportation arc to LE. The duration of this route equals the number of time steps between starting at LS and ending at LE and is four time units.

We develop a mono-directional forward labeling algorithm to solve the ESPPRC on such a time-space network. This algorithm propagates partial routes through $\mathcal{G}$ in order to find the route with maximum negative reduced cost and holds as follows:

Route representation: We represent a partial route r from origin node o to a node $n \in \mathcal{N}$ by a label $\Gamma_r = (T_r^{pos}, T_r^{dur}, T_r^{cost}, (T_r^{task_i})_{i \in \mathcal{I}_{sv}}, (T_r^{station_l})_{l \in \mathcal{L}})$, consisting of the following resources:

T_r^{pos}	denotes the station where route r ends;
T_r^{dur}	denotes the duration of route r;
T_r^{cost}	denotes the cost of route r;
$T_r^{task_i}$	indicates whether task $i \in \mathcal{I}_{sv}$ is performed along route r ($T_r^{task_i} = 1$) or not ($T_r^{task_i} = 0$);
$T_r^{station_l}$	indicates whether station $l \in \mathcal{L}$ is visited or unreachable on route r ($T_r^{station_l}$=1), or reachable and not yet visited ($T_r^{station_l}$=0).

Resource extension functions: We introduce the following notation to define resource extension functions (REFs), which propagate the label of a partial route to its next node. We represent an arc a as the combination of its start node $(t_a^{start}, l_a^{start})$ and end node (t_a^{end}, l_a^{end}). As the AGVs are not allowed to move upstream, i.e., backwards in the station hierarchy (cf. Section 3.3.2), we define the sets $\mathcal{U}_l \subseteq \mathcal{L}$ that include all stations that are unreachable from station l. We use η_{alt} to indicate whether arc a occupies station l at time t ($\eta_{alt} = 1$) or not ($\eta_{alt} = 0$), and note that only operation arcs occupy stations. Let $\mathcal{I}_a^{cover}$ be the set of tasks which are performed along arc a. We then initialize all resources at the origin node o to zero such that

$$T_r^{pos} = T_r^{dur} = T_r^{cost} = T_r^{task_i} = T_r^{station_l} = 0,$$

and use the following REFs $F_a(\Gamma_r)$ to extend a route r by arc a to r' such that $\Gamma_{r'} \leftarrow F_a(\Gamma_r)$ with

$$F_a^{pos}(\Gamma_r) = l_a^{end}; \tag{3.6a}$$

$$F_a^{dur}(\Gamma_r) = T_r^{dur} + t_a^{end} - t_a^{start}; \tag{3.6b}$$

$$F_a^{cost}(\Gamma_r) = T_r^{cost} + \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T} | \eta_{alt}=1} \beta_{lt} + \gamma_v (t_a^{end} - t_a^{start}); \tag{3.6c}$$

$$F_a^{task_i}(\Gamma_r) = T_r^{task_i} + \begin{cases} 1 & \text{if } i \in \mathcal{I}_a^{cover} \\ 0 & \text{else} \end{cases} \qquad \forall i \in \mathcal{I}_{sv}; \tag{3.6d}$$

$$F_a^{station_l}(\Gamma_r) = T_r^{station_l} + \begin{cases} 1 & \text{if } (l = l_a^{start} \vee l \in \mathcal{U}_{l_a^{end}}) \wedge T_r^{station_l} = 0 \\ 0 & \text{else} \end{cases} \qquad \forall l \in \mathcal{L}. \tag{3.6e}$$

The end station of the extended route corresponds to the end station of arc a, i.e., l_a^{end} (3.6a). Straightforwardly, we propagate the route duration by adding the duration of arc a (3.6b) and the cost resource by summing up occupation cost and duration cost along arc a (3.6c). For all tasks $i \in \mathcal{I}_a^{cover}$ that are performed along arc a, we set the corresponding resource $T_r^{task_i}$ to one (3.6d). Finally, we set $T_r^{station_l}$ to one for all stations that have been visited along route r or that are unreachable from the end station l_a^{end} (3.6e). Note that we can only extend a partial route by an arc a if none of the tasks in $\mathcal{I}_a^{cover}$ have been performed yet, and if all predecessors of the tasks in $\mathcal{I}_a^{cover}$ have already been performed.

Dominance rules: We eliminate partial routes as soon as they are dominated by another partial route to keep the number of explored states as small as possible. Our REFs are monotonously increasing and allow for the following dominance check. Let $\Gamma_k = [T_k^{pos}, T_k^{dur}, T_k^{cost}, (T_k^{task_i})_{i \in \mathcal{I}_{sv}}, (T_k^{station_l})_{l \in \mathcal{L}}], k \in 1, 2$, be two labels associated with two different partial routes. Then, Γ_1 dominates Γ_2, i.e., Γ_2 and its corresponding route can be withdrawn from our search if all of the following conditions are fulfilled.

$$T_1^{pos} = T_2^{pos} \tag{3.7a}$$

$$T_1^{dur} \leq T_2^{dur} \tag{3.7b}$$

$$T_1^{cost} + \gamma_v (T_2^{dur} - T_1^{dur}) \leq T_2^{cost} \tag{3.7c}$$

$$T_1^{task_i} \geq T_2^{task_i} \qquad \forall i \in \mathcal{I}_{sv} \tag{3.7d}$$

$$T_1^{station_l} \leq T_2^{station_l} \qquad \forall l \in \mathcal{L} \tag{3.7e}$$

We ensure that we only compare two labels that end at the same station (3.7a). Γ_1 dominates Γ_2 if its route has a shorter duration (3.7b), lower cost (3.7c), covers a superset of tasks (3.7d), and visits a subset of stations (3.7e) compared to the route of Γ_2. For a correct cost comparison, we have to artificially align the durations of both routes by excluding the cost that arise due to duration differences between both routes, i.e., $\gamma_v(T_2^{dur} - T_1^{dur})$. If Γ_1 dominates Γ_2 and $\Gamma_1 \neq \Gamma_2$, we discard Γ_2. When two labels Γ_1 and Γ_2 are equal, we keep the label that was created first.

Feasibility check: To discard labels that cannot be extended to a feasible route with negative reduced cost anymore, we use the following additional notation: let σ_l be the minimum transportation time from station l to the dummy end station LE; q_{vi} be the

processing time of task i for vehicle v; λ_{li} be a binary parameter that states whether task i can be performed at station l; and C_s^{UB} is the current upper bound, i.e., the best integer-feasible solution found so far. Then, we discard a label if at least one of the following conditions is met:

$$T_r^{cost} + \gamma_v(\sigma_{T_r^{pos}} + \sum_{i \in \mathcal{I}_{sv} | T_r^{task_i} = 0} q_{vi}) \geq \kappa_v; \tag{3.8a}$$

$$\exists i \in \mathcal{I}_{sv} : T_r^{task_i} = 0 \wedge \sum_{l \in \mathcal{L} | T_r^{station_l} = 0} \lambda_{li} = 0; \tag{3.8b}$$

$$T_r^{dur} + \sigma_{T_r^{pos}} + \sum_{i \in \mathcal{I}_{sv} | T_r^{task_i} = 0} q_{vi} \geq C_s^{UB}. \tag{3.8c}$$

Condition (3.8a) discards r as soon as it no longer yields negative reduced cost, anticipating the remaining duration cost. Condition (3.8b) discards r if a missing task cannot be completed without violating the flow restrictions of the AGVs. Condition (3.8c) discards r as soon as it cannot be completed to a feasible route with a duration below C_s^{UB}, because a longer route cannot be part of an improving integer-feasible solution.

3.4.2.3 Branching strategies

If the optimal solution to the RMP is fractional, we use a B&B algorithm to obtain integer-feasible solutions. We use a depth-first strategy and branch on the node that has the highest depth in the B&B tree. In case of a tie, we prioritize the node with the smaller lower bound. Whenever we find an integer LP solution, we update the upper bound C_s^{UB}. We prune nodes with integer-feasible solutions as well as nodes with a lower bound that exceeds $C_s^{UB} - 1$, because these cannot yield an improving integer-feasible solution. Our search terminates when all nodes are pruned, i.e., when the global lower bound matches the global upper bound.

We apply two branching techniques: *i*) on the assignment of a task to a station for a vehicle (assignment-based branching), and *ii*) on the start time of a vehicle's task (temporal branching). If both strategies can be applied to a fractional solution, we prioritize assignment-based branching.

Assignment-based branching: We iterate over all vehicles and check the integrality of task-to-station assignments in the LP solution. If multiple task-to-station assignments are fractional, we branch on the assignment of the vehicle with the higher makespan in the LP solution. If multiple fractional task-to-station assignments exist for the same vehicle,

we choose the one that is closest to integrality, i.e., whose absolute distance to zero or one is minimal. We create two branches. In the left/right branch, we forbid/enforce that the vehicle receives the task at the respective station.

Temporal branching: We iterate over all vehicles and their tasks and monitor the start times of the operation arcs selected in the LP solution. If a task is split onto multiple arcs with different start times, we denote the earliest/latest start time of the arcs involved by τ_1/τ_2. We then branch as follows: in the left/right branch, we forbid/enforce that the operation arc which includes the task starts at time $\left\lceil \frac{\tau_1+\tau_2}{2} \right\rceil$ or afterwards. In case of multiple split tasks, we prioritize the vehicle with the higher makespan in the LP solution.

We note that both branching techniques only affect the route generation in the pricing problems and allow to exclude arcs in the time-space network for the respective vehicle. Accordingly, the size of the pricing problems reduces the deeper we develop the B&B tree.

3.4.2.4 Algorithmic framework

Solving Problem 3, we face a large amount of symmetry that results from the problem's min-max objective in which only a single column determines the objective value. Usually, we exploit the problems' structure to derive tight bounds that resolve such a symmetry problem. In our specific case, finding such bounds remains a tedious task, because the problem by nature shows no promising characteristics that allow to do so. Accordingly, we embed our B&P algorithm into a framework that iteratively exploits artificial upper bounds.

Figure 3.7 shows the pseudocode of this framework. We first solve the root node LP relaxation of the RMP to obtain a global lower bound C_s^{LB} on the minimum required segment cycle time C_s. Then, we set an artificial upper bound $\hat{C}_s^{UB} = \lceil C_s^{LB} \rceil + 1$ and solve the problem using the B&P algorithm. By doing so, $\hat{C}_s^{UB}$ strengthens Condition (3.8c) of the feasibility check in the pricing problems, where we generate much fewer columns. Thereby, we prevent evaluating a large number of valueless linear combinations of long and short routes in the RMP, and the RMP resembles to a large extend to a feasibility problem. Our search terminates when we find an integer-feasible solution C_s^{UB} below $\hat{C}_s^{UB}$ as this solution is always optimal. Otherwise, we increase our artificial upper bound $\hat{C}_s^{UB}$ by one time unit and reiterate.

```
1  Solve LP relaxation of RMP (C_s^LB);
2  Ĉ_s^UB ← ⌈C_s^LB⌉ + 1;
3  loop
4  |   Solve B&P with artificial upper bound Ĉ_s^UB;
5  |   if C_s^UB < Ĉ_s^UB exists then
6  |   |   break;
7  |   else
8  |   |   Ĉ_s^UB ← Ĉ_s^UB + 1;
9  |   end
10 end
11 C_s ← C_s^UB;
```

Figure 3.7: Pseudocode of the algorithmic framework.

3.5 Design of experiments

This section details our design of experiments. We first outline the scope of our studies in Section 3.5.1, before we detail the corresponding computational design in Section 3.5.2.

3.5.1 Scope

The scope of our studies is twofold. First, we analyze the benefits of different flexibility levers in FALs and their impact on operational performance. Second, we compare FALs to LALs.

3.5.1.1 Flexibility analyses

To analyze different flexibility levers within an FAL segment, we study the following configurations, which allow us to quantify the benefit of each flexibility lever and to identify potential positive reinforcements between operation and routing flexibility (cf. Figure 3.8).

NF: The no flexibility (NF) configuration bases on predefined task sequences and task-to-station assignments. This setting constitutes a baseline for our analyses.

OF: The operation flexibility (OF) configuration considers predefined task-to-station assignments but allows for optimized task sequences to exploit operation flexibility.

RF: The routing flexibility (RF) configuration considers predefined task sequences but allows for optimized task-to-station assignments to exploit routing flexibility.

FF: The FF configuration combines the OF and RF configurations such that both task sequences and task-to-station assignments can be optimized.

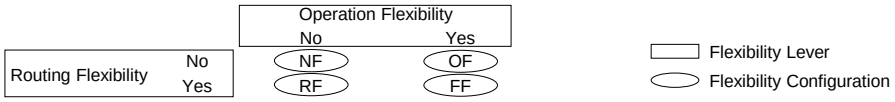


Figure 3.8: Relation between flexibility levers and flexibility configurations.

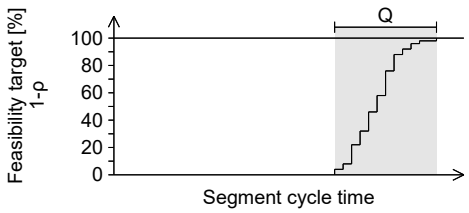


Figure 3.9: Example of the variation width.

We solve all configurations by running our algorithmic framework from Section 3.4 but fix certain decisions a priori depending on the configuration. Appendix B.3 details how we fix task sequences and task-to-station assignments a priori for each configuration.

To evaluate the results, we use the following four performance indicators.

WIP: The WIP, which is proportional to the segment cycle time of an FAL segment, denotes the number of vehicles simultaneously processed in a segment. In our results, we state the average WIP per station, i.e., the total WIP divided by the number of stations L.

Variation width (Q): Given a set of sample sequences, the variation width denotes the spread of the minimum required segment cycle times for each sequence. Figure 3.9 shows an example of the distribution of minimum required segment cycle times for a set of sequences and shows the resulting Q. Accordingly, a low Q implies that the segment cycle time is more robust towards various vehicle sequences. We report the variation width divided by the cycle time c.

Utilization (U): The utilization denotes the ratio of the workers' realized workload compared to the total work time. A high utilization indicates well-utilized workers, whereas a low utilization indicates that workers are often idle.

Output level (O): A feasibility target below 100% causes delays in some sequences. These delays may lead to line stoppages in succeeding segments. We use the output level, which denotes the ratio of the actual output rate compared to the target output rate $1/c$, to measure these delays.

3.5.1.2 Flexible assembly to line assembly comparison

We compare FALs to LALs for both a stationary demand mix and the ramp-up of alternative drivetrain technologies, specifically electric vehicles. Here, we study the respective utilization, output level, and WIP. While the WIP in an LAL equals its number of stations L^{line}, determining the utilization and output level for a fair comparison remains non-trivial. We determine the utilization in an LAL segment based on the optimal vehicle sequence that results from a status-quo mixed-model sequencing problem (cf. Appendix B.4), neglecting sequencing constraints of other segments. We use this optimal sequence to simulate its assembly and denote the duration of line stoppages, which occur whenever a worker is not able to complete tasks within the limits of her station. We use the cumulative duration of these line stoppages to calculate the average cycle time including line stoppages $\bar{c}$. We then adapt the standard utilization formula for assembly lines (3.9) using the average workload per vehicle $\bar{u}$. For the LAL output level, we compare $1/\bar{c}$ to $1/c$ (3.10). By so doing, we obtain a worst-case estimate on the FAL benefits by accounting for an utopian best case for the LAL assessment.

$$U^{line} = \frac{\bar{u}}{L^{line} \cdot \bar{c}} \tag{3.9}$$

$$O^{line} = \frac{1/\bar{c}}{1/c} = \frac{c}{\bar{c}} \tag{3.10}$$

3.5.2 Computational design

We implemented all algorithms in C++, using Gurobi 8.1 to solve linear programs and ran all experiments on a standard computer with an i7-4810 CPU at 2.80 GHz and 16 GB of RAM.

To avoid non-disclosure conflicts, we develop a realistic instance set for our studies by adapting a popular standard data set from literature[19]. We verified its plausibility with the help of our industry partner. Our instances comprise eleven tasks and are representative for a segment within the automotive assembly that may potentially be replaced with an FAL. We account for significantly different vehicle types, e.g., conventional and electric vehicles, as follows:

[19] https://assembly-line-balancing.de/

1. We randomly split the set of tasks $\mathcal{I}$ into three disjoint sets $\mathcal{I} = \mathcal{I}^C \cup \mathcal{I}^E \cup \mathcal{I}^A$. While $\mathcal{I}^C$ and $\mathcal{I}^E$ include three tasks that are exclusive for conventional and electric models, $\mathcal{I}^A$ contains the remaining five tasks that apply to all types of models.

2. We consider two conventional and two electric models by randomly assigning tasks from $\mathcal{I}^C, \mathcal{I}^A$ to conventional models and from $\mathcal{I}^E, \mathcal{I}^A$ to electric models, and draw tasks with a probability of 75%. We choose each task's processing time from a uniform distribution that can deviate up to 50% from the original instance's value (see Hottenrott & Grunow, 2019).

3. To generate vehicle sequences, we account for a certain demand share $(\pi^C : \pi^E)$ between conventional and electric vehicles, and consider random permutations of these vehicles. With these random permutations, we account for the fact that an FAL segment should be able to process any permutation that is favorable for the up- and downstream LAL segments. Accordingly, our procedure ensures that one may derive an operational sequencing that focuses solely on the requirements of the more restricted LAL segments.

4. We repeat steps 1-3 with different random seeds to create multiple instances.

We generate instances based on this scheme to study the following setups:

3.5.2.1 Flexibility analyses

For our flexibility analyses, we consider twelve instances. For each instance we solve 50 sequences with 20 vehicles, which we identified as a sufficient size for unbiased results during preliminary analyses (cf. Appendix B.5). We account for a balanced demand mix between conventional and electric vehicles with $(\pi^C : \pi^E) = (50 : 50)$, and we generate the strategic FAL design for each instance as described in Hottenrott and Grunow (2019). Here, we assume Manhattan metric and set the transportation time between two neighboring stations to $\omega = 1.0c$. We note that the AGV speed remains an additional field of study during subsequent analyses.

3.5.2.2 Flexible assembly to line assembly comparison

To compare the performance of FALs to LALs, we use an instance setup similar to the flexibility analyses but account for additional demand mix scenarios. Besides the balanced mix with a (50 : 50) vehicle split, we consider two additional demand mixes with vehicle splits of (70 : 30) and (90 : 10) to model potential ramp-up stages for new

vehicle technologies. Here, we generate LALs for comparison by using the standard mixed-model assembly line balancing problem, minimizing the number of stations (cf. Appendix B.6), and compute the cycle time based on Hoffmann (1992).

We account for task duplicates as follows: in an FAL, every task can be assigned to two stations, such that at most one duplicate of each task exists. Allowing for additional task duplicates reveals only diminishing effects (cf. Appendix B.7). We compare such an FAL against an LAL without task duplicates. While this comparison may initially appear biased, we chose this design, because the benefits for which one would prefer an LAL over an FAL (e.g., no condition-based, real-time decisions; JIS stocking at stations) could not be ensured if the LAL has task duplicates.

3.6 Results

This section details the results of our computational studies. For the sake of conciseness, we report aggregated values throughout this section and refer to the paper's electronic companion for detailed results. To ensure the validity of the reported average values, we performed two-sided Wilcoxon rank tests to ensure statistical significance. We highlight 10%, 5%, and 1% significance levels with single, double, and triple asterisks respectively.

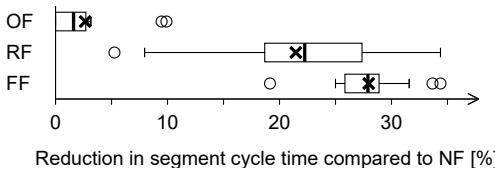
3.6.1 Flexibility analyses

We first study the impact of the flexibility levers in an FAL segment. In the following, we report results for a 90% feasibility target without further notice as this constitutes a common threshold in practice. We refer to Appendix B for extended results on different feasibility targets.

Figure 3.10 shows the impact of all three flexibility levers for a representative instance by denoting the required segment cycle time for a certain feasibility target $1 - \rho$ and each flexibility configuration. A high segment cycle time renders all vehicle sequences for all flexibility configurations feasible at the price of a high WIP. To achieve a certain feasibility target at a lower segment cycle time, we notice a clear dominance between the four configurations as the NF configuration requires the highest segment cycle time for all feasibility targets. While the OF configuration slightly outperforms the NF configuration, the RF and FF configurations show significant improvements, with the FF configuration dominating the RF configuration.

Figure 3.11 summarizes the segment cycle time reductions for the OF, RF, and FF configurations compared to the NF configuration across all instances. The relative reductions of the segment cycle time equal the WIP reductions as they share the proportional

Figure 3.10: Impact of the flexibility levers on the feasibility target for a representative instance.



	Min	Max	Median	Mean
OF	0.0	9.9	1.6	2.7
RF	5.3	34.4	22.3	21.5
FF	19.1	34.4	27.9	28.0

Figure 3.11: Reduction in the segment cycle time (WIP) due to flexibility for a feasibility target of 90%.

relation described in Section 3.3.1. We find that all flexibility configurations achieve a reduction compared to the NF configuration, because operation and routing flexibility allow for additional production strategies which reduce the waiting times of AGVs at stations. The OF configuration, however, falls short compared to the RF and FF configurations for two reasons. First, options to interchange a task sequence are limited as the vehicles' precedence graphs are dense. Second, the flow restrictions of the AGVs limit the exploitation of operation flexibility. Interestingly, the reduction potential of the FF configuration exceeds the sum of the RF and OF reductions. This shows a positive reinforcement between both flexibility levers, i.e., operation flexibility allows to better exploit routing flexibility and vice versa. These effects remain similar for varying feasibility targets (cf. Appendix B.8).

Table 3.1 reports average values for all four performance indicators across all instances. Additionally, Δ states their relative deviations from the NF configuration. We note significant WIP reductions of up to 28%, which show that operation and routing flexibility can mitigate the main disadvantage of an FAL. Moreover, the variation width can be reduced by up to 54%, which shows that flexibility allows to reduce an FAL's sensitivity towards unfavorable vehicle sequences. Further, increasing flexibility allows to significantly improve the utilization by up to 6.9%. The output level shows no significant changes across all configurations as it mainly depends on the feasibility target.

At a first glimpse, a simultaneous improvement of utilization and WIP seems to contradict classical production theory. This is possible, because the flexibility in an FAL reduces both waiting times of vehicles and idle times of workers, which highlights the

Table 3.1: Average results for a feasibility target of 90%.

Configuration	WIP [1/L]	Δ^{WIP}	Q [1/c]	Δ^{Q}	U [%]	Δ^{U}	O [%]	Δ^{O}
NF	2.4		1.6		76.1		99.92	
OF	2.3	-2.7% ***	1.4	-8.8% *	76.6	+0.7% ***	99.92	+0.0%
RF	1.9	-21.5% ***	1.0	-31.2% **	79.8	+4.9% ***	99.94	+0.0%
FF	1.7	-28.0% ***	0.7	-54.0% ***	81.3	+6.9% ***	99.95	+0.0%

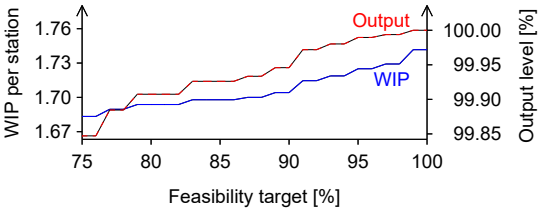


Figure 3.12: Impact of the feasibility target on the average WIP and output level for the FF configuration.

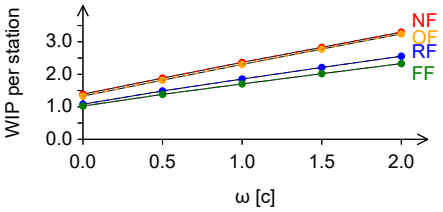


Figure 3.13: Impact of the AGV transportation time ω on the average WIP for the FF configuration.

conceptual novelty of FALs. For all quantities but the output level, we note an improvement hierarchy that is similar to our initial analyses. While the FF configuration yields the highest improvements, revealing a positive reinforcement between operation and routing flexibility, the OF configuration yields the lowest improvements.

Figure 3.12 shows the impact of the feasibility target on the WIP and the output level for the FF configuration. As can be seen, an increased feasibility target results in a higher WIP and a higher output level. By definition, a feasibility target of 100% achieves an output level of 100%, because no deteriorating delays occur. However, preventing delays requires a higher segment cycle time which entails a higher WIP. We notice that these trends remain consistent for all other flexibility configurations (cf. Appendix B.9).

Finally, we analyze how the efficiency of the AGV system, i.e., the AGV transportation speed, affects the performance of an FAL. Figure 3.13 shows how the average WIP per station changes for different transportation times between neighboring stations ω. For the artificial case of $\omega = 0.0c$, we observe a WIP per station close to 1.0. The WIP increases almost linearly with increasing ω for all flexibility configurations. We conclude that an efficient AGV system is a key success factor when operating an FAL, since high WIP levels require more space and complicate AGV routing.

Concluding, our studies show significant benefits of routing flexibility and combined routing and operation flexibility in an FAL, while the benefits of sole operation flexibility remain limited. While these results indicate quantifiable benefits, one may want to consider additional factors when deciding on the right flexibility configuration of an FAL segment in practice. Avoiding operation and routing flexibility allows for standardized

Table 3.2: Increase in the utilization and output level for FALs compared to LALs with closed stations.

	NF	OF	RF	FF
U	+21.3% ***	+22.1% ***	+27.4% ***	+29.9% ***
O	+31.5% ***	+31.5% ***	+31.5% ***	+31.6% ***

Table 3.3: Increase in the WIP for FALs compared to LALs depending on the AGV transportation time ω.

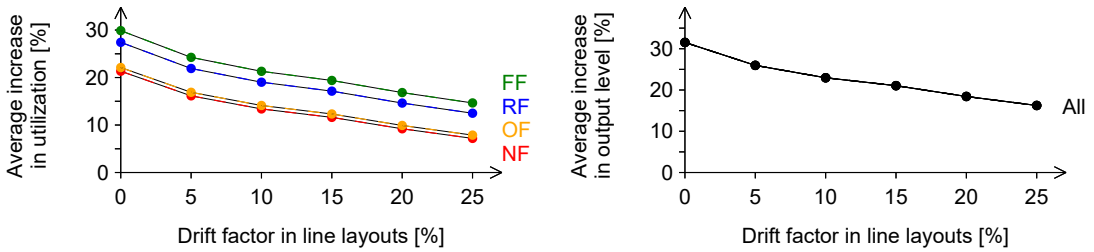
$\omega[c]$	NF	OF	RF	FF
0.0	+36.9% ***	+31.1% ***	+6.5% *	+0.7%
0.5	+85.8% ***	+79.3% ***	+46.8% ***	+36.5% ***
1.0	+133.8% ***	+127.3% ***	+82.8% ***	+68.2% ***
1.5	+179.3% ***	+173.9% ***	+118.1% ***	+99.2% ***
2.0	+225.6% ***	+220.2% ***	+152.2% ***	+129.8% ***

task sequences that prevent worker confusion. Further, it allows for predefined task locations that enable station stocking.

3.6.2 Comparison of flexible assembly layouts and line assembly layouts

In the following, we compare FALs to LALs. Herein, we first consider a stationary demand mix, before we analyze the performance of both layouts during a ramp-up scenario.

Stationary scenario: Table 3.2 analyzes the average increase in utilization and output level between an FAL and an LAL with closed stations. We see that an FAL achieves higher utilization at higher output levels. While the utilization improvements vary between 21.3% and 29.9% depending on the flexibility configuration of the FAL, the output level improvements remain constant at approximately 31.5%. In practice, one may improve the utilization and the output level of an LAL by allowing workers to drift into subsequent stations when facing varying workloads. Figure 3.14 shows the average increase in utilization and output level for an FAL compared to an LAL with different drift factors. As can be seen, the FAL preserves a minimum average utilization improvement of more than 7% and an average output level improvement of 16% for any drift factor up to 25%. We note that drifting remains challenging in practice, because it requires tasks at neighboring stations to be independent of each other and to be performed at different positions of the vehicle. Accordingly, we consider an LAL drift factor of 25% as a worst-case evaluation of FAL benefits.



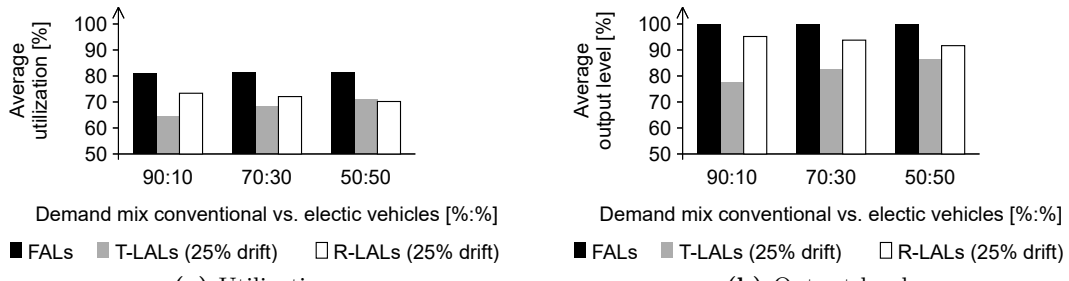
(a) Utilization. **(b)** Output level.

Figure 3.14: Increase in the utilization and the output level for FALs compared to LALs with opened stations.

Clearly, an FAL allows for these improvements in exchange for an increased WIP, which depends significantly on the efficiency of the used AGV system. Table 3.3 reports the average total WIP increase depending on the transportation time between neighboring stations ω in an FAL. The results confirm our findings from Section 3.6.1 and reveal an average WIP increase of up to 225.6% depending on the AGV speed and the flexibility configuration. Only for the artificial case of $\omega = 0.0c$, the FF configuration of an FAL shows a negligible WIP difference compared to an LAL. Further, we observe a significant deterioration for configurations without routing flexibility.

Ramp-up scenario: So far, our analyses focused on a stationary demand mix with an equal share of conventional and electric vehicles for which both the FAL and the LAL segments have been designed. However, the diffusion of electric vehicles is currently a major ramp-up process for many automotive manufacturers, during which FALs can be of particular advantage compared to LALs. Once designed, LALs are known to be inflexible regarding shifting demand mixes. Contrary, FALs are expected to be capable of mitigating shifting demand mixes without a need for overcapacities. Against this background, we now study the performance of FAL and LAL segments during a ramp-up scenario where the demand mix between conventional and electric vehicles shifts from $(90:10)$ to $(70:30)$ to the target mix of $(50:50)$. For this analysis, we detail the results for the FF configuration and refer to Appendix B.10 for results of other flexibility configurations.

Typically, automotive manufacturers account for ramp-up scenarios by including overcapacities into the design of an LAL. Accordingly, we study the performance of three different segment designs: *i*) an LAL exclusively designed for the target demand mix (T-LAL), *ii*) an LAL that accounts for ramp-up overcapacities (R-LAL), i.e., an LAL that has been designed according to Appendix B.6 but with Constraints (B.9d) being



(a) Utilization. **(b)** Output level.

Figure 3.15: Performance of FALs (FF configuration), T-LALs, and R-LALs during ramp-up.

duplicated for all relevant demand mixes, and *iii*) an FAL that remains as in earlier studies designed for the (50 : 50) target demand mix. For both LAL designs, we consider opened stations, where workers are allowed to drift 25% into the subsequent station.

Figure 3.15 shows the average utilization (3.15a) and output level (3.15b) for each segment design. We see that the FAL segment is capable of processing all demand mixes at an output level close to 100% with a constant utilization of around 81%. This shows that the FAL can balance even large changes in the demand mix without any overcapacities by solely adjusting its segment cycle time. The T-LAL falls behind the performance of the FAL, revealing an additional decrease of up to 7% and 9% in utilization and output level, depending on the ramp-up stage. The R-LAL reveals a better performance than the T-LAL, especially in the early stages of the ramp-up, but still falls short compared to the FAL. We note that the R-LAL requires on average 8.2% more stations than the T-LAL, and hence a corresponding WIP increase, to realize these improvements.

Figure 3.16 shows the required adjustments of the FAL's segment cycle time for both ramp-up stages. As can be seen, these adjustments remain on average 3.0% for the (90 : 10) mix and 1.0% for the (70 : 30) mix. Even for the most extreme demand mix deviation, the worst increase in segment cycle time remains below 12.5%. Adjusting the segment cycle time does not cause major disturbances of the production in an FAL, because it does not require any changes at the stations. This shows that an FAL can accommodate different demand mixes during ramp-up scenarios without overcapacities and with only minor adjustments of the segment cycle time, while an LAL reveals significant disadvantages, even if overcapacities are considered during its design process.

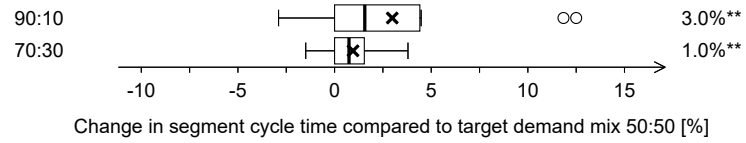


Figure 3.16: Adjustments of segment cycle time in FALs (FF configuration) during ramp-up.

3.7 Conclusion

In this chapter, we studied FALs with a particular focus on their deployment in IoT-driven automotive manufacturing. We derived analytical insights on the benefits of FALs for a minimal example. To confirm these insights for realistic instances, we proposed a chance-constrained problem formulation, presented a problem-specific decomposition, and developed a B&P algorithm to solve the resulting subproblems. We applied this methodological framework to an extensive numerical study in order to analyze the impact of different operational policies resulting from the combination of the different flexibility levers within FALs. To support technology selection, we compared the performance of FALs to LALs. Our results allow to conclude this paper with the following managerial insights:

FALs show a clear impact hierarchy for different flexibility levers. At the price of a high WIP, a high segment cycle time renders all vehicle sequences feasible for all flexibility configurations. To realize feasibility at a low WIP, routing flexibility remains the main improvement lever, allowing for average WIP reductions of 21.5%. Operation flexibility reveals significantly lower improvement potentials and reduces the WIP on average by 2.7%. However, both flexibility levers reinforce each other such that fully exploiting both levers allows to reduce the WIP on average by 28.0%.

Flexibility in FALs resolves the well-known trade-off between WIP and utilization improvements. In classical production theory, there exists a well-known trade-off between improving a layout's utilization or its WIP. Our results show that operation and routing flexibility can resolve this trade-off in an FAL.

FALs outperform LALs in terms of utilization and output level. Our results show that FALs allow for up to 30% higher utilization and output levels compared to LALs. We obtain these best-case benefits when comparing fully flexible FALs to LALs with closed stations. However, even in the worst case, i.e., when comparing FALs without operation and routing flexibility to LALs with opened stations where workers are allowed to drift 25% into the subsequent station, a significant improvement of a 7% higher utilization and a 16% higher output level remains.

The operational performance of FALs depends on the efficiency of the AGV

system. The FAL improvements come at the price of a higher WIP. Our results show that this WIP disadvantage is sensitive to the transportation times between stations, i.e., the speed and operational efficiency of the AGV system employed. While the WIP remains moderate in fully flexible FALs for transportation times that do not exceed the cycle time, we observe significant deteriorations for higher transportation times and configurations without routing flexibility.

FALs are particularly beneficial during ramp-up stages for new technologies. Our results show that FALs are highly flexible during ramp-up stages with shifting demand mixes and preserve stable utilization and output levels. This can be achieved through minor adaptations of the segment cycle time. The adjustments of this flexibility lever do not require physical system reconfigurations. LALs, in contrast, show a significant performance deterioration.

4 Robust car sequencing for conventional line assembly layouts

This chapter is based on an article published as:

Hottenrott, A., Waidner, L., & Grunow, M. (2020). Robust car sequencing for automotive assembly. *European Journal of Operational Research*.
https://doi.org/10.1016/j.ejor.2020.10.004

Abstract

JIS material supply is the status quo in the automotive industry. In this process, the assembly sequence of vehicles is set several days prior to production and communicated to the suppliers. The committed sequence is essential for efficient operations both at the OEM and its suppliers. In practice, however, sequence stability is insufficient. Short-term disruptions, such as quality problems and missing parts, put the sequence at risk. If a disruption occurs, the affected vehicle is removed from the sequence. The resulting gap is closed by bringing the succeeding vehicles forward. Such sequence alterations, however, cause workload changes and potentially work overloads at the assembly stations. As a remedial measure, additional sequence alterations are necessary, which further disturb material supply. Robustness against short-term sequence alterations is currently a key objective of automotive manufacturers.

In this chapter, we propose a sequencing approach that includes the vehicles' failure probabilities in order to generate robust sequences. Robust sequences are sequences that can be operated without modifications, even when vehicles fail. We develop a B&B algorithm that optimally solves small-sized instances. For large-sized instances, we design a sampling-based ALNS metaheuristic. The superiority of our approach is validated in a simulation study using real-world data from a major European manufacturer. We find reductions in the expected work overloads of 72% and 80%, compared to the industry solution and compared to an approach taken from literature which does not take failures into account.

4.1 Introduction

JIS material supply is the status quo in the automotive industry. In this process, the assembly sequence of vehicles is set several days prior to production and communicated to the suppliers. The committed sequence is essential for efficient operations both at the OEM and its suppliers. In practice, however, sequence stability is insufficient (Inman, 2003; Lehmann & Kuhn, 2020; Meissner, 2010). Short-term disruptions, such as quality problems and missing parts, put the sequence at risk. If a disruption occurs, the affected vehicle is removed from the sequence. In order to maintain the efficiency of the MMAL, the resulting gap is rarely left idle. Instead, the succeeding vehicles are brought forward. Such sequence alterations, however, cause workload changes and potentially work overloads at the assembly stations. As a remedial measure, additional sequence alterations are necessary, which further disturb material supply. Robustness against short-term sequence alterations is currently a key objective of automotive manufacturers.

We define robust sequences to be those that can be operated without modifications, even when vehicles fail. With regard to vehicle failures, a robust sequence achieves high efficiency, i.e., no empty hangers, and does not cause work overloads at the stations. Robust sequences are beneficial in terms of material supply. Only the parts for the failed vehicles have to be sorted out, whereas the JIS supply for the other vehicles remains unaffected. In order to plan robust sequences, the vehicles' failure probabilities need to be taken into account. Although most OEMs possess sufficient data to determine the vehicles' failure probabilities, the analysis of such data is currently not undertaken. We contribute to the industry's desire for data-driven planning by developing a robust car-sequencing approach that exploits this data.

The job of the sequence planner is to create the assembly sequence of the vehicles produced in a shift. Since multiple variants with a variety of options are assembled on the same MMAL, the workloads at the stations differ between the vehicles. The workloads of some vehicles are higher than the cycle time, whereas the workloads of others are lower. If several consecutive vehicles require high workloads at the same station, work overloads occur. All of the numerous OEMs we have recently collaborated with use car-sequencing approaches to minimize work overloads. Herein, so-called H_o/N_o sequencing rules are used: out of any subsequence of N_o vehicles, only H_o vehicles are allowed to require option o. These sequencing rules are usually experience-based. For example, empirical evidence has shown that work overloads do not occur when only one out of two consecutive vehicles is a long version. The corresponding sequencing rule is $H_o/N_o = 1/2$. Sequence A in Figure 4.1 complies with this sequencing rule. The vehicles

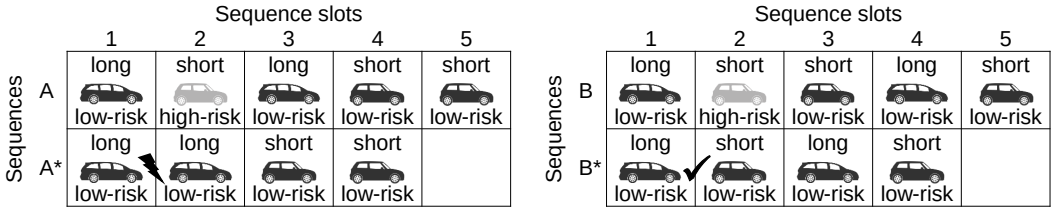


Figure 4.1: Examples for a non-robust sequence (left) and a robust sequence (right).

in sequence slots 1 and 3 are long versions, whereas the vehicles in sequence slots 2, 4, and 5 are short versions.

Whenever a failure on a vehicle occurs, the initially planned sequence is altered. Let us assume that the vehicle in sequence slot 2 of sequence A has a high risk of failure. If this vehicle is not available on time, the resulting gap in the sequence would be closed by bringing the succeeding vehicles forward as illustrated in sequence A*. However, this would cause a violation of the sequencing rule between sequence slots 1 and 2, and therefore potentially a work overload. Sequence B is a robust sequence. For this sequence, a failure on the high-risk vehicle would not cause a work overload (sequence B*).

The failure probabilities depend on the vehicles' specifications. Typical failure drivers are:

- Body color, e.g., paint quality defects occur more often for certain colors;
- Body variant, e.g., the sunroof cutting process frequently causes problems;
- Selected options, e.g., part suppliers vary in delivery date adherence.

A key failure driver is the body color. While the overall paint shop reliability affects the failure probabilities of all vehicles, we also observe significant differences between colors. These differences occur due to setups, paint age, and paint type. Moreover, experience plays an important role. The more often a color is used, the less likely failures are. Figure 4.2 shows the demand shares and failure probabilities of ten body colors at a major European OEM. The failure probabilities represent the shares of vehicles that were not available on time for assembly. We see that demand shares and failure probabilities are negatively correlated. Popular colors, e.g., $C6$, have low failure probabilities. For color $C4$, on the other hand, the demand share is only 1.5% while the failure probability is almost 20%. One particular challenge for an OEM is the introduction of a new color. On the one hand, demand for new colors is usually high, because dealers like to display them in their showrooms. On the other hand, new processes are less stable and entail higher failure probabilities.

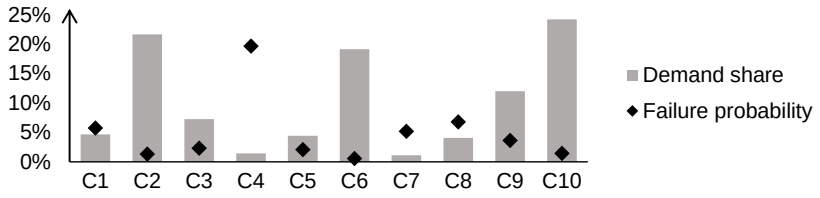


Figure 4.2: Demand share and failure probability by color at a European manufacturer.

Current car-sequencing approaches are insufficient, because they neglect the vehicles' failure probabilities. This is surprising considering their huge impact and the fact that this data is available to most OEMs. In this chapter, we study the research question of how this data can be used in order to create robust sequences that can be forwarded to the suppliers to enable a JIS supply of the required parts. Accordingly, we assume that failed vehicles are removed from the sequence. We aim to close the resulting gaps by bringing succeeding vehicles forward without causing work overloads.

The reinsertion of failed vehicles is planned in real time by a resequencing controller. Vehicles may be reinserted in the same shift or in later shifts. The decision on when to reinsert a failed vehicle depends on the required time to resolve the failure, the required time to properly supply all stations with the vehicle's parts (including JIS parts), the urgency of the vehicle and the availability of a sequence slot in which the reinsertion does not cause work overload at any of the stations. Because it is only possible to deal with these influencing factors in real time, we do not consider the reinsertion of failed vehicles in our planning approach for a robust JIS supply.

In this chapter, we focus on sequence planning for the final assembly. For our approach, it is irrelevant if the same sequence is used throughout all production stages (body shop, paint shop, and final assembly) or if different sequences are used and the target sequence is obtained in resequencing buffers between the production stages. This is possible as long as the sequences do not differ too much and the used buffers have sufficient size and allow for random access.

We contribute to research into car sequencing in multiple ways:

- We formulate the robust car-sequencing problem as a mixed-integer non-linear program.
- We develop a B&B algorithm that solves small-sized instances optimally. We derive tailored lower bounds based on individual options that significantly improve the algorithmic performance.

- We propose a sampling-based ALNS heuristic, which builds on observations we extract from optimal B&B solutions. For adapted, small-sized standard benchmark instances, we compare our heuristic against the exact algorithm. Our heuristic generates solutions with an average optimality gap of 0.49%.
- We solve the robust car-sequencing problem for a major European OEM, using extensive real-world data from 51 shifts. The average run time of our heuristic on these industry instances is below ten minutes. In a comprehensive simulation study, we quantify the benefits of including vehicles' failure probabilities in sequence planning. We find reductions in the expected work overloads of 72% and 80%, compared to the industry solution and compared to a literature approach which does not take failures into account. When a new color is launched, the outcome of our approach is only marginally affected, whereas the outcomes of the two other approaches deteriorate significantly. Moreover, we show that the relative benefits of our approach are consistent for different paint shop reliabilities. The largest absolute benefits are found when paint shop reliability is low.

This chapter is structured as follows: In Section 4.2, we review the related literature. In Section 4.3, we formally define the robust car-sequencing problem. Our exact B&B algorithm is described in Section 4.4. From the optimal solutions to illustrative instances, we derive insights for the design of our sampling-based robust car-sequencing heuristic (RCSH), which is introduced in Section 4.5. We assess the computational performance of our algorithms and present the results of our simulation analysis in Section 4.6. In Section 4.7, we summarize our findings and discuss future research directions.

4.2 Literature review

Sequencing vehicles on MMALs has received considerable attention in the scientific literature. For a comprehensive review, we refer to Boysen et al. (2009). The ultimate goal in sequence planning is to minimize work overloads at the stations. Three approaches exist, i.e., mixed-model sequencing, car sequencing, and level scheduling. While work overloads are addressed explicitly in mixed-model sequencing, surrogate objectives are used in car sequencing and level scheduling. Level scheduling seeks to balance part consumption over time. In car sequencing, so-called H_o/N_o sequencing rules are used, which limit the number of work intensive options in a subsequence of vehicles.

The car-sequencing approach is most common in industry (Lehmann & Kuhn, 2020). An overview of the literature is given by Solnon, van Cung, Nguyen, and Artigues

(2008). The car-sequencing problem is proven to be NP-hard in the strong sense (Kis, 2004). Many solution approaches have been proposed. These range from exact approaches, like integer programming (Drexl & Kimms, 2001; Gravel, Gagné, & Price, 2005) and constraint programming (Brailsford, Potts, & Smith, 1999), to heuristics, such as greedy search (Hindi & Ploszajski, 1994), local search (Benoist, 2008; Estellon, Gardi, & Nouioua, 2008), genetic algorithms (Warwick & Tsang, 1995), ant colony optimization (Solnon, 2008), and combinations of them (C. C. Ribeiro, Aloise, Noronha, Rocha, & Urrutia, 2008).

We identify three particularly relevant research streams on car sequencing. One research stream extends the scope of car sequencing beyond the final assembly (e.g., Briant, Naddef, & Mounié, 2008; Cordeau, Laporte, & Pasin, 2008; Gagné, Gravel, & Price, 2006). The goal is to determine sequences which remain unchanged throughout the paint shop and the final assembly. While the sequencing rule violations are minimized for the final assembly, the number of color changes is minimized for the paint shop. This streams aims at unchanged sequences between the paint shop and the final assembly. However, it does not address uncertain vehicle failures and sequence stability.

The drawback of identical sequences is that they are a compromise between the requirements of the paint shop and the final assembly. When the sequences are allowed to differ, work overloads can be reduced, but buffers become inevitable. Therefore, another research stream addresses resequencing in buffers between the paint shop and the final assembly. Boysen et al. (2012) review the literature in this research stream. Three buffer types exist, i.e., mix banks, pull-off tables, and random access buffers. Heuristics for resequencing in mix banks are proposed by Choi and Shin (1997); Ding and Sun (2004); Taube and Minner (2018). Boysen, Golle, and Rothlauf (2011) study resequencing using pull-off tables. The goal is to reshuffle the outgoing sequence from the paint shop in order to minimize the violations of the sequencing rules in the final assembly. In practice, many OEMs employ random access buffers. Inman (2003) determines the required sizes of such buffers. Gusikhin, Caprihan, and Stecke (2008) investigate resequencing in random access buffers. Given stochastic processing times in the paint shop, they optimize the paint shop sequence such that there is a high probability that the planned assembly sequence can be restored. Their heuristic essentially postpones frequent body-color combinations, while infrequent ones are painted earlier compared to their position in the assembly sequence. Thereby, they increase the probability that an adequately painted body is available on time for assembly. We follow a different approach. We want to be able to close gaps occurring due to failures by bringing succeeding vehicles forward without causing work overloads.

A third research stream targets the definition of the H_o/N_o sequencing rules. Even though the definition of appropriate sequencing rules is crucial, this research stream has received little attention. Bolat and Yano (1992a) derive sequencing rules based on operational characteristics of the assembly line. Lesert, Alpan, Frein, and Noiré (2011) define sequencing rules using processing times. Golle, Boysen, and Rothlauf (2010) develop a multiple-sequencing-rules approach to further improve the accuracy. All approaches in the literature assume deterministic processing times and neglect the risk of sequence alterations due to failures. To increase robustness, the sequencing rules could be defined more strictly. However, this would neglect the valuable information about the vehicles' failure probabilities and thus unnecessarily restrict sequence planning.

Despite the large amount of research on the car-sequencing problem, only the paper of Gusikhin et al. (2008) addresses uncertainties. Specifically, no paper exists that determines sequences for the final assembly which are robust against work overloads in the event of vehicle failures.

4.3 Robust car-sequencing problem

In this section, we formally define the robust car-sequencing problem as a mixed-integer non-linear program. Given a set of vehicles $v \in V$ with a set of options $o \in O$, the goal is to assign every vehicle to a sequence slot $t = 1, \ldots, T$ such that the expected number of sequencing rule violations is minimized. In contrast to traditional car-sequencing literature, we sequence vehicles instead of models. Nowadays, the vehicles are very heterogeneous and their failure probabilities depend on a multitude of specifications. Thus, a problem formulation based on vehicles is more appropriate in our opinion.

X_{vt} is a binary decision variable that states whether vehicle v is assembled in slot t. The binary parameter a_{vo} shows if vehicle v requires option o. The sequencing rule for option o is encoded in the parameters H_o and N_o. Violations occur whenever more than H_o out of N_o successive vehicles require option o.

Every vehicle is associated with a failure probability f_v. The failure probability is based on the vehicle's specifications, e.g., its color, body variant, or the contained options, and can be derived from historical data. We assume that the vehicles' failure probabilities are independent. This assumption might be critical, because batching in preceding production stages, e.g., the press shop, can cause correlated failures on multiple vehicles. The delay of a truck can affect the JIS part supply of many vehicles. Also regarding paint quality defects, the main source of failures, correlated failures may occur, because many OEMs still batch vehicles of the same color. However, modern paint

Table 4.1: Problem notation.

Index sets	
$v \in V$	Vehicles
$o \in O$	Options
$c \in C$	All $2^{\lvert V \rvert}$ possible failure scenarios
$t = 1, \ldots, T$	Sequence slots
Parameters	
a_{vo}	1 if vehicle v requires option o, otherwise 0
b_{cv}	1 if vehicle v exists in failure scenario c, otherwise 0
H_o/N_o	Sequencing rule for option o: at most H_o out of N_o successively sequenced vehicles require option o
f_v	Failure probability of vehicle v
p_c	Probability of failure scenario c
Decision variables	
X_{vt}	1 if vehicle v is assembled in slot t, otherwise 0
Y_{co}	Number of sequencing rule violations for option o in failure scenario c
Z_{cot}	Length of evaluation window for option o starting at slot t in failure scenario c

shops can produce nearly any sequence and do not require extensive setups anymore, such that batching is no longer necessary reducing the respective correlations.

We determine all $2^{|V|}$ possible failure scenarios $c \in C$. In every scenario, a vehicle either exists or fails. The binary parameter b_{cv} is 1 if vehicle v exists in scenario c and 0 if it fails. We derive the probability p_c for failure scenario c using Equation (4.1).

$$p_c = \prod_{v \in V} \Big(b_{cv}(1 - f_v) + (1 - b_{cv}) f_v \Big) \qquad \forall c \in C \tag{4.1}$$

Let Y_{co} be a decision variable that tracks the number of violations of the sequencing rule for option o in failure scenario c. We define this variable to be continuous, however, Constraints (4.2d) ensure that its value is integer. Z_{cot} is an auxiliary integer decision variable that indicates the length of the evaluation window for option o starting at slot t in failure scenario c. We count violations using the logic of Bolat and Yano (1992b), because Golle, Rothlauf, and Boysen (2014) showed that this logic gives the best outcomes in terms of anticipating work overloads. We extend the sequence by sufficiently many dummy vehicles prior to the first slot $t = 1$ and after the last slot $t = T$. These dummy vehicles do not require any option and never fail. With the notation defined in Table 4.1, we formalize the robust car-sequencing problem as shown in Equations (4.2a) - (4.2g).

$$\min \quad \sum_{c\in C} p_c \sum_{o\in O} Y_{co} \tag{4.2a}$$

s.t.

$$\sum_{t=1}^{T} X_{vt} = 1 \qquad \forall v\in V \tag{4.2b}$$

$$\sum_{v\in V} X_{vt} = 1 \qquad \forall t=1,\ldots,T \tag{4.2c}$$

$$Y_{co} = \sum_{t=H_o-N_o+2}^{T-H_o} \max\left\{0;\ \sum_{t'=t}^{t+Z_{cot}} \sum_{v\in V} b_{cv}a_{vo}X_{vt'} - H_o - N_o\left(1-\sum_{v\in V} b_{cv}X_{vt}\right)\right\} \qquad \forall c\in C; o\in O \tag{4.2d}$$

$$N_o = \sum_{t'=t}^{t+Z_{cot}} \sum_{v\in V} b_{cv}X_{vt'} \qquad \forall c\in C; o\in O; t=H_o-N_o+2,\ldots,T-H_o \tag{4.2e}$$

$$X_{vt} \in \{0,1\} \qquad \forall v\in V; t=1,\ldots,T \tag{4.2f}$$

$$Z_{cot} \in \mathbb{Z}^+ \qquad \forall c\in C; o\in O; t=H_o-N_o+2,\ldots,T-H_o \tag{4.2g}$$

In the Objective (4.2a), we minimize the expected number of violations across all options. Constraints (4.2b) and (4.2c) ensure that every vehicle is assigned to one slot and that every slot is filled by one vehicle. In Constraints (4.2d), we derive the number of violations of the sequencing rule for option o in failure scenario c. Violations occur whenever more than H_o vehicles requiring option o are assigned to a subsequence of size N_o. In order to track the violations, we consider the final sequences in the failure scenarios, i.e., the sequences that remain after failed vehicles are removed and succeeding vehicles are brought forward. The second line of Constraints (4.2d) ensures that we only count the violations in an evaluation window starting at slot t if the vehicle assigned to slot t does not fail in the respective failure scenario. The problem is non-linear, because the length of the evaluation window depends on the failed vehicles in the failure scenario. We define the variable Z_{cot} in order to track the required length of the evaluation window. The variable denotes how many sequence slots we have to look ahead such that we obtain a subsequence of N_o unfailed vehicles (Constraints (4.2e)). Finally, in Constraints (4.2f) and (4.2g), we define the variable domains.

The robust car-sequencing problem is a combinatorial optimization problem. In general, all $|V|!$ vehicle permutations are feasible. To calculate the expected number of

violations in a permutation, we must evaluate all $2^{|V|}$ failure scenarios. The problem size inhibits us from using standard solving software. We can linearize the formulation shown. However, the key purpose of the mathematical model is to provide a formal definition of our problem and not an input for off-the-shelf solvers. As the linear equivalent is still intractable for off-the-shelf solvers, we continue to develop a B&B algorithm instead.

4.4 Exact branch-and-bound algorithm with tailored lower bounds

We develop a B&B algorithm to solve the robust car-sequencing problem optimally for small numbers of vehicles. We first introduce the base algorithm (Section 4.4.1) and then propose two algorithmic improvements (Sections 4.4.2 and 4.4.3). From the optimal solutions to small-sized instances, we derive insights (Section 4.4.4) for the design of our ALNS heuristic presented in Section 4.5.

4.4.1 Base algorithm

In our B&B algorithm, we perform a tree search. The tree levels represent the sequence slots. The root level represents the first slot $t = 1$ and the lowest leaf level the last slot $t = T$. Hence, our trees have $|V|$ levels. We create $|V|$ trees in parallel. The root node of the v^{th} tree represents vehicle v.

We extend a node by considering all remaining, yet unplanned vehicles. A path from a root node to another node represents a partial sequence. We determine the expected number of violations in all partial sequences. Let $V'_q \subseteq V$ be the vehicles in partial sequence q and $c \in C'_q$ be the respective failure scenarios with probabilities p'_{qc}. We derive the final partial sequences for all failure scenarios, i.e., the partial sequences that remain after failed vehicles are removed and succeeding vehicles are brought forward. Knowing the final partial sequences resolves the non-linearity of Constraints (4.2d) and (4.2e), because the length of the evaluation windows in the final partial sequences is N_o. Let T'_{qc} denote the number of vehicles in the final partial sequence of failure scenario c. We obtain T'_{qc} using Equations (4.3), where b'_{qcv} denotes whether vehicle v exists in failure scenario c. The parameter a'_{qcot} indicates whether the vehicle at position t in the final partial sequence of failure scenario c requires option o. We can then adapt the determination of the expected number of violations in Constraints (4.2d) and the Objective (4.2a), and calculate the expected number of violations Y'_q in partial sequence q using Equation (4.4).

Table 4.2: Example data.

v	$a_{v,O1}$	f_v
$V1$	1	0.1
$V2$	1	0.2
$V3$	0	0.3

Table 4.3: Failure scenarios for node 4 in Figure 4.3.

c	$b'_{4,c,V1}$	$b'_{4,c,V2}$	$p'_{4,c}$	$T'_{4,c}$
1	0	0	0.02	0
2	0	1	0.08	1
3	1	0	0.18	1
4	1	1	0.72	2

$$T'_{qc} = \sum_{v \in V'_q} b'_{qcv} \qquad \forall c \in C'_q \tag{4.3}$$

$$Y'_q = \sum_{c \in C'_q} p'_{qc} \sum_{o \in O} \sum_{t=H_o-N_o+2}^{T'_{qc}-H_o} \max\left\{ \sum_{t'=t}^{t+N_o-1} a'_{qcot'} - H_o; 0 \right\} \tag{4.4}$$

$$LB_q = Y'_q \tag{4.5}$$

Example Figure 4.3: An example with three vehicles is shown at the top of Figure 4.3. We create three trees with three levels each. The root nodes of the first, second, and third tree correspond to the vehicles $V1$, $V2$, and $V3$ respectively. Let us consider a single option $O1$ with sequencing rule $H_{O1}/N_{O1} = 1/2$. Additionally, we assume the vehicle characteristics shown in Table 4.2. For illustrative purposes, let us investigate the partial sequence of node 4. The set of vehicles is $V'_4 = \{V1, V2\}$ and there are four relevant failure scenarios as summarized in Table 4.3. While no violations occur in the first three failure scenarios, one violation occurs if both vehicles exist. Because the probability of this failure scenario is 0.72, the expected number of violations Y'_4 is 0.72.

The expected number of violations in the partial sequence Y'_q is a lower bound LB_q for the expected number of violations in the complete sequence (cf. Equation (4.5)). (*Example Figure 4.3:* Any complete sequence that includes the partial sequence of node 4 will have at least 0.72 expected violations.)

We employ a best-first search strategy. We choose the node with the lowest lower bound for further extension. That means node q_1 is chosen over node q_2 if $LB_{q_1} < LB_{q_2}$. If there is a tie, we choose the node with the greater depth in the tree. This is when

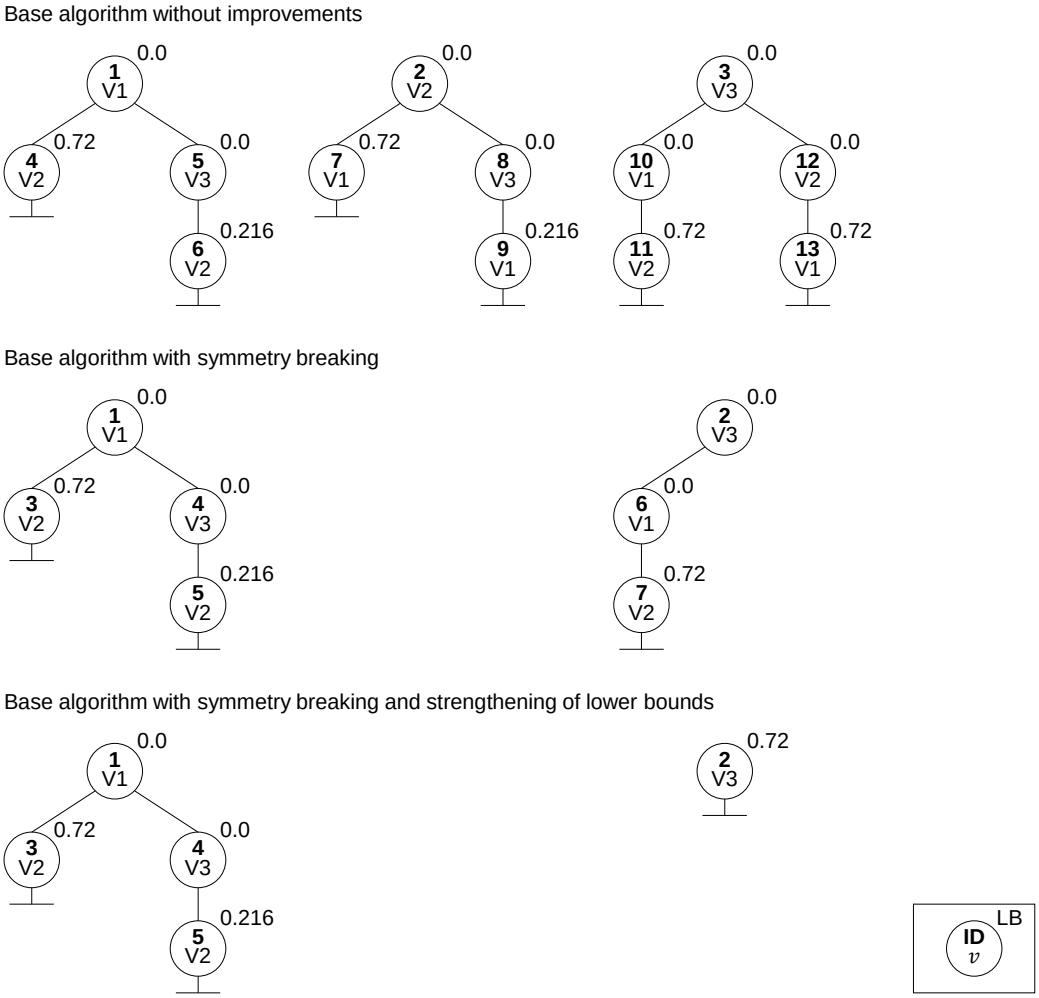


Figure 4.3: Example for B&B tree search without/with algorithmic improvements.

$LB_{q_1} = LB_{q_2}$, q_1 is extended before q_2 if $|V'_{q_1}| > |V'_{q_2}|$. (*Example Figure 4.3:* Node 5 is chosen before node 4 and before node 2.)

Whenever a node q^* represents a complete sequence ($|V'_{q^*}| = |V|$), it is pruned and we update the upper bound UB, formally $UB \leftarrow min(UB, Y'_{q^*})$. Initially, UB is set to ∞. We prune a node q^- whenever its lower bound is greater or equal to the upper bound ($LB_{q^-} \geq UB$). (*Example Figure 4.3:* The upper bound is updated to 0.216 in node 6 and node 4 is pruned afterward.)

The B&B algorithm terminates as soon as all nodes in all trees are either extended or pruned. Then, the upper bound UB shows the optimal solution. (*Example Figure 4.3:* The optimal solution is 0.216 and it is found in node 6. The optimal sequence is $V1{-}V3{-}V2$. The B&B algorithm evaluates a total of 13 nodes to prove optimality.)

4.4.2 Symmetry breaking

The trees exhibit significant symmetries, since every sequence can be reversed to form another sequence with same objective value. Without loss of generality, we break symmetry by enforcing that vehicle $V2$ is sequenced later than vehicle $V1$. The corresponding constraint is denoted in Equation (4.6).

$$\sum_{t=1}^{T} t\, X_{V1\,t} \leq \sum_{t=1}^{T} t\, X_{V2\,t} \tag{4.6}$$

(*Example Figure 4.3:* The optimal sequence $V1{-}V3{-}V2$ in the example shown in Figure 4.3 can be reversed to form the optimal sequence $V2{-}V3{-}V1$ found in node 9. Constraint (4.6) excludes symmetric sequences from the solution space. In the center of Figure 4.3, we show the trees with symmetry breaking. The B&B algorithm terminates after seven nodes.)

4.4.3 Strengthening of lower bounds based on individual options of unplanned vehicles

To reduce the tree sizes further, we strengthen the lower bounds for the partial sequences. The lower bounds described above are not tight, because they only consider the violations that occur in the sequence that has been planned already, but ignore the violations that occur in the remaining, yet unplanned sequence. By anticipating future violations, we obtain more effective lower bounds. We conservatively approximate the expected future violations by analyzing every option individually and neglecting interactions between the options. Also, we neglect violations that occur at the crossover between the planned sequence and the unplanned sequence. Instead of considering all failure scenarios, we only consider the one in which all remaining vehicles exist. This scenario has by far the highest probability, because the vehicles' failure probabilities are generally low. Thereby, we reduce the problem of estimating the expected number of future violations to the well-known deterministic car-sequencing problem.

Let $v \in V_q''$ be the remaining, yet unplanned vehicles for partial sequence q. Since we evaluate every option individually, we define the binary decision variable X''_{qvto}, which shows whether vehicle v is placed in slot t when investigating option o. The continuous variable $\bar{Y}_{qo}$ denotes the minimum number of future violations for option o when none of the remaining vehicles fail. For every option o, we solve the optimization problem

shown in Equations (4.7a) - (4.7e). It corresponds to the deterministic car-sequencing problem with a single option.

$$\min \quad \bar{Y}_{qo} \tag{4.7a}$$

s.t.

$$\sum_{t=1}^{|V_q''|} X''_{qvto} = 1 \qquad \forall v \in V_q'' \tag{4.7b}$$

$$\sum_{v \in V_q''} X''_{qvto} = 1 \qquad \forall t = 1,\ldots,|V_q''| \tag{4.7c}$$

$$\bar{Y}_{qo} = \sum_{t=H_o-N_o+2}^{|V_q''|-H_o} \max\left\{0; \sum_{t'=t}^{t+N_o-1} \sum_{v \in V_q''} a_{vo} X''_{qvt'o} - H_o\right\} \tag{4.7d}$$

$$X''_{qvto} \in \{0,1\} \qquad \forall v \in V_q''; t = 1,\ldots,|V_q''| \tag{4.7e}$$

To solve this problem, we simply have to evaluate all unique permutations of the binary a_{vo} values of the remaining, yet unplanned vehicles V_q''. We ignore dominated permutations. If existing, it is always optimal to place a 1 in the last H_o positions of the permutation. Also, it is always optimal to place a 1 in the first H_o positions of the permutation, since we neglect the violations that occur at the crossover between the planned sequence and the unplanned sequence.

Let p_q'' represent the probability of the failure scenario in which all remaining vehicles exist. The minimum expected number of future violations Y_q'' can then be estimated conservatively using Equation (4.8). We can strengthen the lower bound of node q as the sum of Y_q' and Y_q'' (cf. Equation (4.9)).

$$Y_q'' = p_q'' \sum_{o \in O} \bar{Y}_{qo} \tag{4.8}$$

$$LB_q = Y_q' + Y_q'' \tag{4.9}$$

(*Example Figure 4.3:* At the bottom of Figure 4.3, we illustrate the tree search with symmetry breaking and strengthened lower bounds. We see that the lower bound of node 2 has increased from 0.0 to 0.72. The unplanned vehicles for this node are $V1$ and $V2$. Both require option $O1$. Assuming both vehicles exist, the best permutation of the a_{vO1} values is $[1,1]$. This permutation causes one violation of the $H_{O1}/N_{O1} = 1/2$ sequencing rule. The probability that both vehicles exist is $0.9 \cdot 0.8 = 0.72$. Therefore,

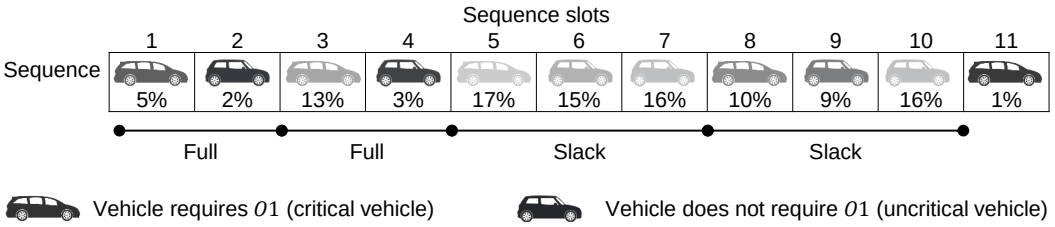


Figure 4.4: Optimal sequence for an instance with eleven vehicles and one option $O1$.

the expected number of future violations can be conservatively approximated at 0.72. Due to the strengthened lower bound, we prune node 2 after evaluating node 5 and do not have to extend it further. Our B&B algorithm evaluates only five nodes.)

4.4.4 Observations from optimal sequences

We study the optimal sequences for randomly generated, small-sized instances with a single option $O1$ to derive insights for the design of our heuristic. As an example, Figure 4.4 shows the optimal sequence for an instance with eleven vehicles. In this instance, the sequencing rule for option $O1$ is $H_{O1}/N_{O1} = 1/2$. The option assignments are uniformly distributed with a 50% probability that a vehicle requires $O1$. The vehicles' failure probabilities are drawn from a uniform distribution on the interval (0.0, 0.2). In the figure, the failure probabilities are indicated below the vehicle icons.

Let us use the term "critical vehicle" for vehicles that require $O1$ and "uncritical vehicle" for vehicles that do not require $O1$. A "subsequence" starts with a critical vehicle and ends before the ${H_{O1}}^{\text{th}}$ critical vehicle following this critical vehicle. We refer to a "full subsequence" if there are exactly $N_{O1} - H_{O1}$ uncritical vehicles in the subsequence. In a "slack subsequence", in contrast, we find more than $N_{O1} - H_{O1}$ uncritical vehicles. The following four observations are persistent in the optimal sequences of different instances:

Observation 1 - Composition of subsequences: The optimal sequences consist of the minimal number of full subsequences and the maximal number of slack subsequences. Full subsequences entail a high risk of violation, while slack subsequences hedge against violations. Any surplus of uncritical vehicles is equally distributed among the slack subsequences, because the benefits of additional uncritical vehicles are degressive. In the example in Figure 4.4, two full subsequences and two slack subsequences are scheduled. One additional uncritical vehicle is placed in every slack subsequence.

Observation 2 - Distribution of uncritical vehicles: Uncritical vehicles with low failure probabilities are placed in full subsequences, and uncritical vehicles with high failure probabilities in slack subsequences. This is intuitive since a failure of an uncritical vehicle in a full subsequence immediately causes a violation, whereas in a slack subsequence at least two uncritical vehicles would need to fail. In the example, the uncritical vehicles with low failure probabilities, i.e., 2% and 3%, are found in full subsequences, whereas the uncritical vehicles with high failure probabilities, i.e., 9%, 15%, 16%, are found in slack subsequences.

Observation 3 - Begin/end of sequence: We find H_{O1} critical vehicles with low failure probabilities at the begin and end of optimal sequences. Critical vehicles that are unlikely to fail are most likely to cause violations. It is intuitive to place them at the begin and end of the sequence. In the example, the critical vehicle with a failure probability of 5% is placed at the begin and the critical vehicle with a failure probability of 1% is placed at the end of the sequence.

Observation 4 - Surplus of critical vehicles: If there are more critical vehicles than capacity in full subsequences, we find additional critical vehicles with high failure probabilities at the begin or end of the sequences (not shown in example).

4.5 Sampling-based robust car-sequencing heuristic (RCSH)

We use the observations from studying optimal sequences to design a robust car-sequencing heuristic. The exact B&B algorithm introduced in Section 4.4 is not scalable to large-sized instances. We are facing two challenges when the number of vehicles increases. First, it is computationally intractable to evaluate all $2^{|V|}$ failure scenarios. Second, it is too time-consuming to find the optimal sequence across all $|V|!$ vehicle permutations.

4.5.1 Sampling approach

We propose a sampling approach to address the first challenge. We observe that most failure scenarios have marginal probabilities. Instead of considering all $2^{|V|}$ possible failure scenarios explicitly, we create a sample of failure scenarios $c \in C^S$. The sample needs to be sufficiently large so that it is representative. We do not consider scenario probabilities anymore. However, more likely scenarios might appear multiple times in our sample. As objective value, we consider the average violations in the final sequences of all failure scenarios in the sample.

We employ the descriptive sampling approach introduced by Saliby (1990) in order to reduce the variability of the results. In descriptive sampling, a deterministic set of purposive selected values is used. For each sample, the sequence of these values is permuted. For our application, we use a set of $|V|$ values, defined as shown in Equation (4.10), where U^{-1} is the inverse cumulative distribution function of a uniform distribution on the interval (0.0, 1.0).

$$xd_i = U^{-1}[(i - 0.5)/|V|] \qquad \forall i = 1, \ldots, |V| \tag{4.10}$$

We randomly shuffle the list of values for every failure scenario. If the value at the v^{th} position is less than the failure probability of vehicle v, then vehicle v is considered to fail. Otherwise, it is considered to be produced in its scheduled sequence position.

Example: For ten vehicles ($|V| = 10$), the list of values is [0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95]. In each failure scenario, we randomly shuffle this list of values, e.g., [0.85, 0.65, 0.15, 0.35, 0.75, 0.05, 0.45, 0.95, 0.25, 0.55]. Let us consider vehicles' failure probabilities of [0.01, 0.05, 0.16, 0.03, 0.19, 0.12, 0.01, 0.04, 0.20, 0.06], then the vector of failed (0) and unfailed (1) vehicles in this failure scenario would be [1, 1, 0, 1, 1, 0, 1, 1, 1, 1].

4.5.2 Adaptive large neighborhood search

We address the second challenge by introducing an ALNS heuristic. We chose ALNS, because it provides a framework to employ multiple modification operators. In Section 4.4.4, we identified several persistent characteristics of optimal sequences. ALNS allows us to integrate operators that are inspired by different observations in optimal sequences. To explore a large part of the solution space, we additionally employ a random operator and simulated annealing.

ALNS was first introduced by Ropke and Pisinger (2006). The idea is to improve a feasible initial solution (Section 4.5.2.1) by iteratively applying modification operators that alter the incumbent solution. The difference to other neighborhood search strategies is that multiple operators exist (Section 4.5.2.2). The operators compete to be used. The probability that an operator is selected depends on its performance in previous iterations. The newly obtained solution is either accepted and becomes the incumbent solution, or it is rejected and the incumbent solution remains unchanged (Section 4.5.2.3). Depending on the outcome, the applied operator is rewarded (Section 4.5.2.4). Thereby, the search adapts to the instance at hand and to the state of the search. The iterations continue

until a stopping criterion is met (Section 4.5.2.5). The best solution found by the end of the search is taken as the result of the heuristic.

4.5.2.1 Initial solution

The starting point of our RCSH is a feasible initial solution. Because all $|V|!$ vehicle permutations are feasible, we could select any of them. However, we aim for a good initial solution to speed up our algorithm. We therefore solve the deterministic counterpart, i.e., the car-sequencing problem without vehicle failures as proposed by Bolat and Yano (1992b), using Gurobi 8.1.0.

4.5.2.2 Modification operators

We design five modification operators. The operators select one vehicle and move it to a new position or swap the positions of two vehicles. Such a modification alters the sequence. However, many parts of the sequence remain unaffected. We therefore do not reevaluate the entire sequence. It is computationally faster to only assess the changes in objective value that occur in the neighborhoods around the modifications. For illustrative purposes, assume a sequence of 300 vehicles in which the last vehicle is moved forward by two positions. Obviously, the violations in the beginning of the sequence are not affected and do not need to be reevaluated. The sizes of the affected neighborhoods depend on the sequencing rule of the option to be evaluated and the failure scenario.

Critical vehicle operator: Our first operator is inspired by the optimal composition of subsequences (Observation 1). It seeks to move a critical vehicle that is likely to cause violations in its current position to a position where it is less likely to cause violations. We investigate every option individually and consider the sequence without failures. We identify vehicles that require the option and are positioned in a full or overfull subsequence, i.e., there are at least H_o vehicles that require the option in a subsequence of size N_o. Next, we identify candidate destination positions. These are positions in which only slack subsequences occur, even when the critical vehicle is moved there. For every option, we obtain a list of critical vehicles and a list of candidate destination positions. By pairing the two lists for all options, we create a list of move possibilities. Pairing two lists means to take the Cartesian product of the two lists, i.e., to get the list of all ordered pairs (a, b) where a belongs to list A and b belongs to list B. We then randomly pick one element from this paired list.

Uncritical vehicle operator: The idea of this operator is also inspired by Observation 1. It reverses the logic of the first operator. We aim to relieve the risk of violations in a full or overfull subsequence by adding an uncritical vehicle. Again, we look into a single option and consider the sequence without failures. We identify vehicles which do not require the option and are positioned in slack subsequences. Then, we identify candidate destination positions in full or overfull subsequences. We obtain two lists for every option. The first list contains the uncritical vehicles that can be moved, the second list the candidate destination positions. We pair both lists for all options and obtain a list of move possibilities from which we randomly draw one element.

Swap operator: Contrary to the operators discussed so far, the swap operator does not move a single vehicle but swaps the positions of two vehicles. It is motivated by the observation that we find uncritical vehicles with a low failure probability in full subsequences and the ones with a high failure probability in slack subsequences (Observation 2). We again assess every option individually and consider the sequence without failures. We identify pairs of uncritical vehicles, i.e., vehicles that do not require the option. If the vehicle with the higher failure probability is placed in the subsequence with less slack, the vehicle pair is added to the list of swap possibilities from which we randomly select one pair.

Begin/end operator: This operator makes use of the observation that H_o critical vehicles are placed at the begin and end of optimal sequences (Observation 3). We check whether this characteristic is fulfilled for all options in the incumbent sequence without failures. If not, we identify all vehicles that require the respective option and randomly choose one of them to be moved either to the begin or to the end of the sequence respectively.

Random operator: Ropke and Pisinger (2006) state that it is beneficial to add a random operator, because purely myopic operators are prone to get stuck in local optima. We account for this with an operator that performs a random move. We randomly pick a vehicle and randomly choose a new position for it.

4.5.2.3 Acceptance criterion

We add a simulated annealing acceptance criterion to our RCSH. In general, we accept every new solution s' over the incumbent solution s if s' is better than s. Furthermore, we also accept new solutions s' with a worse objective value with probability

$p = e^{-(F(s')-F(s))/\tau}$ where F is the objective value and $\tau > 0$ is the current temperature. We employ an instance-specific cooling schedule. Starting from an initial temperature τ_0, the temperature is multiplied by a factor γ at the end of each iteration, where $0<\gamma<1$. Ropke and Pisinger (2006) suggest that τ_0 should be chosen based on the initial solution of the instance at hand. We set τ_0 such that a solution which is 5% worse than the initial solution is accepted with 50% probability. For the cooling factor γ, we choose the conservative value $\gamma = 0.99975$ as proposed in the literature (G. M. Ribeiro & Laporte, 2012; Ropke & Pisinger, 2006). Slow cooling reduces the risk of getting stuck in local optima.

4.5.2.4 Adaptive mechanism

We select a modification operator in each iteration using a roulette wheel mechanism. The selection probability is based on the operator's performance in previous iterations. Therefore, each operator is associated with a weight ρ_j, which indicates the operator's success in previous iterations. Additionally, we introduce a minimum selection probability of $\phi^{min} = 5\%$ for all operators. Especially the random operator is mainly used to diversify the search, however, it is less likely to find improved solutions. The minimum selection probability ensures that its application is not excluded in later search phases. The probability ϕ_j that operator j is chosen then follows Equation (4.11).

$$\phi_j = \phi^{min} + \frac{\rho_j}{\sum_{j' \in J} \rho_{j'}}(1 - |J|\phi^{min}) \qquad \forall j \in J \tag{4.11}$$

We adjust the weights dynamically. Initially, all operators have the same weights $\rho_j = 1, \forall j \in J$. We divide our search into segments (each segment has 100 iterations in our implementation). At the end of a segment, the weights are updated based on the operators' performances during the last segment, as shown in Equation (4.12). The performance of operator j in the last segment is encoded in the score π_j. The scores of all operators are set to zero in the beginning of a new segment. After each iteration, the score of the chosen operator j^* is increased by the reward parameters σ_1, σ_2, or σ_3, as shown in Equation (4.13).

$$\rho_j \leftarrow (1-\eta)\rho_j + \eta\pi_j \qquad \forall j \in J \tag{4.12}$$

$$\pi_{j^*} \leftarrow \pi_{j^*} + \begin{cases} \sigma_1 & \text{if } s' \text{ is new global best} \\ \sigma_2 & \text{if } s' \text{ is accepted and } s' \text{ is better than } s \\ \sigma_3 & \text{if } s' \text{ is accepted and } s' \text{ is worse than } s \\ 0 & \text{if } s' \text{ is rejected} \end{cases} \tag{4.13}$$

As recommended by G. M. Ribeiro and Laporte (2012), we use the values 50, 20, and 5 for σ_1, σ_2, and σ_3 respectively. The decay parameter $\eta \in [0, 1]$ controls the sensitivity of the weight adjustment to the performance in the last segment. We set $\eta = 0.1$ as proposed by Ropke and Pisinger (2006).

4.5.2.5 Stopping criterion

The improvement iterations are continued until a stopping criterion is met. We stop as soon as the best solution found has improved by less than 0.1% over the past 5000 iterations. We choose these conservative values to ensure convergence also on large-sized instances.

4.6 Analysis

Our analysis is structured into two parts. In Section 4.6.1, we assess the computational performance of our exact B&B algorithm and our RCSH. We use a set of small-sized benchmark instances that we adapt from literature. In Section 4.6.2, we show the results of real-world robust car-sequencing instances from our partner OEM. We conduct a simulation study to evaluate the benefits of including the vehicles' failure probabilities in sequence planning. We therefore compare the robustness of the sequences planned by our RCSH with the industry solution and with an approach from literature that does not account for failures. All code is written in C++. We run our experiments on a standard computer equipped with an Intel(R) Core(TM) i7-4810 CPU at 2.80 GHz and 16 GB of RAM.

4.6.1 Computational performance

We use a set of randomly adapted benchmark instances to evaluate the computational performance of our solution algorithms. The instances are based on the new, difficult

Table 4.4: Average number of evaluated nodes in exact B&B algorithm.

	$\|V\| = 4$	$\|V\| = 7$	$\|V\| = 10$
Base algorithm without improvements	34	5285	1 889 823
+ Symmetry breaking	19	2374	786 501
+ Strengthening of lower bounds	19	2201	447 516

data set of Gravel et al. (2005) which can be obtained at `http://csplib.org/Problems/prob001/data/`. We vary the number of vehicles between $|V| = 4$ and $|V| = 10$ and create 50 instances each. All instances feature five options. The vehicles are randomly chosen from the models in the data sets according to the demand shares. The vehicles' failure probabilities are drawn from a uniform distribution on the interval (0.0, 0.2). We generate 1000 failure scenarios using descriptive sampling, as introduced in Section 4.5.1.

4.6.1.1 Computational performance of exact branch-and-bound algorithm

Table 4.4 reports the average number of evaluated nodes in the B&B algorithm as a function of the number of vehicles. One entry represents the average across 50 instances. We present three result sets. The first row shows the performance of the base algorithm without improvements. The second row depicts the performance with symmetry breaking, and the last row the performance with symmetry breaking and strengthening of lower bounds.

We see an exponential increase in the average number of evaluated nodes with a higher number of vehicles. We find that symmetry breaking is very powerful in reducing the number of evaluated nodes. The reduction increases slightly in the number of vehicles and ranges between 44% and 58%. When we additionally strengthen the lower bounds, we can reduce the number of evaluated nodes even further. Hereby, we observe an increasing benefit for larger instances. While the reduction compared to symmetry breaking alone is 0% for instances with four vehicles, it increases to 43% for instances with ten vehicles.

We find similar results for the average run time of the exact B&B algorithm. As shown in Table 4.5, the average run time increases rapidly in the number of vehicles. Adding symmetry breaking and strengthening of lower bounds speeds up the termination, especially on large-sized instances. When comparing the performance with all improvements to the performance without improvements, we compute average run time reductions between 0% on instances with four vehicles and 90% on instances with ten vehicles. Using the algorithm with all improvements, the average run time on instances with ten vehicles is 854 seconds. For larger instances, the run times are prohibitively high.

Table 4.5: Average run time of exact B&B algorithm (in seconds).

	$\|V\| = 4$	$\|V\| = 7$	$\|V\| = 10$
Base algorithm without improvements	0.0	1.0	8652.9
+ Symmetry breaking	0.0	0.4	1984.5
+ Strengthening of lower bounds	0.0	0.4	854.2

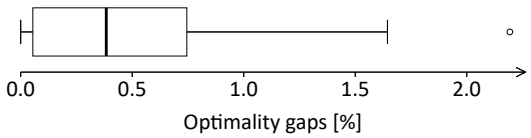


Figure 4.5: Box plot of optimality gaps for RCSH on instances with ten vehicles.

4.6.1.2 Computational performance of RCSH

We assess the performance of our heuristic on the 50 instances with ten vehicles. After running RCSH, we compute the expected number of violations in the obtained sequences and compare it to the optimal value provided by the B&B algorithm.

Figure 4.5 shows the optimality gaps across the 50 instances with ten vehicles. We note that RCSH performs very well. The average optimality gap is 0.49%. The box plot shows that 75% of the instances have gaps below 0.74%. The largest gap is 2.19%. The run times of RCSH are also satisfactory. We observe an average run time of 24 seconds. The exact B&B algorithm, in contrast, requires an average run time of 854 seconds for the same instances.

4.6.2 Simulation study

4.6.2.1 Design of experiments

We use real-world data of our partner OEM to assess the benefits of including the vehicles' failure probabilities in sequence planning. The OEM provided us with the production volumes and the planned sequences for 51 shifts (17 days). In every shift, 300 vehicles are produced. The vehicles differ in the contained options and color. Four options and 16 colors are considered in sequence planning. We estimate the failure probabilities based on failure rates over past weeks. Since the color is the dominant determinant, we assume that the failure probability only depends on the color.

Scenario 1 is our base case and resembles the current situation at the OEM. For this scenario, we use the 51 instances as introduced above. In Scenario 2, we investigate the launch of a new color. On the one hand, new colors are often in high demand. On the other hand, they entail a high probability of failure. According to our partner OEM, a

demand share of ten percent and a failure probability of 25% are typical for a new color. To map these characteristics, we artificially increase the failure probability of a color that has a demand share of approximately ten percent to 25%. In Scenario 3, we assess the impact of paint shop reliability. Paint shop reliability affects all vehicles similarly. We therefore perform a sensitivity analysis in which we systematically alter the failure probabilities of all vehicles by ±50% compared to the base case.

We compare three different approaches to generate sequences. First, we consider our **RCSH** as described in Section 4.5. Second, we consider the sequences that were planned by our partner OEM. We refer to this approach as **OEM**. The OEM is currently using a third-party software tool for sequence planning, which also employs car-sequencing rules. Unfortunately, we do not have detailed knowledge about the algorithms used in this software. Third, we consider an approach from literature that does not account for failures. That is to say, we use a generic MILP solver, in our case Gurobi 8.1.0, to solve the car-sequencing problem as proposed by Bolat and Yano (1992b). We refer to this approach as **LIT**. LIT corresponds to the deterministic counterpart of the robust car-sequencing problem and, hence, to the initial solution of our RCSH.

We optimize the sequences for all 51 instances using all three approaches. We then perform 10 000 simulation runs for each. In the simulation runs, vehicles are removed from the planned sequences based on their failure probabilities. We again apply descriptive sampling as introduced by Saliby (1990). We use the term "planned sequence" to refer to the optimized sequence before vehicles are removed. All planned sequences have a length of 300. The term "final sequence" refers to the sequence that remains in each simulation run after failed vehicles have been removed and succeeding vehicles have been brought forward. The final sequences are shorter than the planned sequences. Having generated the final sequences, we use the logic of Bolat and Yano (1992b) to count violations. We use this logic, because Golle et al. (2014) have shown that it performs best in terms of anticipating work overload. Finally, we determine the expected number of violations in the final sequences by taking the average across all 10 000 simulation runs for every instance. We thus obtain samples with 51 sample points for all three approaches. These samples are the basis for our evaluation. Note that we use common random numbers in the simulation runs. This means that the same vehicles fail for all three approaches.

4.6.2.2 Simulation-based determination of number of failure scenarios

We perform an analysis to determine the appropriate number of failure scenarios in the sampling for our RCSH. We use the data from Scenario 1 and generate instances with

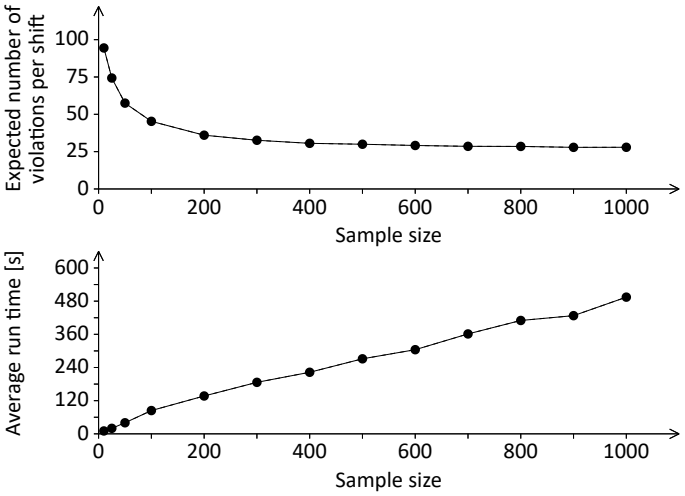


Figure 4.6: Analysis of appropriate sample size of failure scenarios.

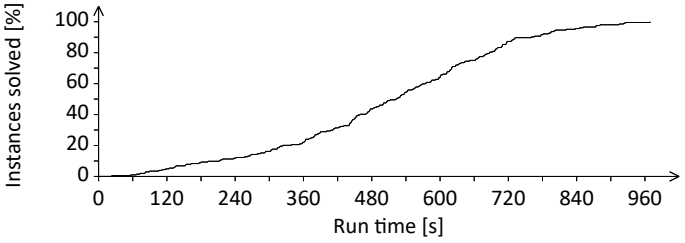


Figure 4.7: Run times of RCSH.

different sample sizes $|C^S|$. We then derive the expected number of violations for all instances using RCSH as described in Section 4.6.2.1.

In Figure 4.6, we compare the average number of violations across all instances for different sample sizes $|C^S|$. Also, we visualize the average run time of RCSH. We see that the average number of violations in the final sequences converges for sample sizes above 500. The average run time, on the other hand, increases nearly linearly in the sample size. We conclude that a sample size of 1000 is sufficiently large. Increasing the sample size beyond 1000 only increases run time without providing significant gains in solution quality.

Figure 4.7 shows the distribution of the run times of RCSH for the chosen sample size of 1000. We note an average run time of 505 seconds. For 90% of the instances, the run time is below 760 seconds. The longest run time is 969 seconds. These run times are acceptable, because the sequence is planned a few days ahead of production. We conclude that RCSH is capable of solving all real-world instances efficiently.

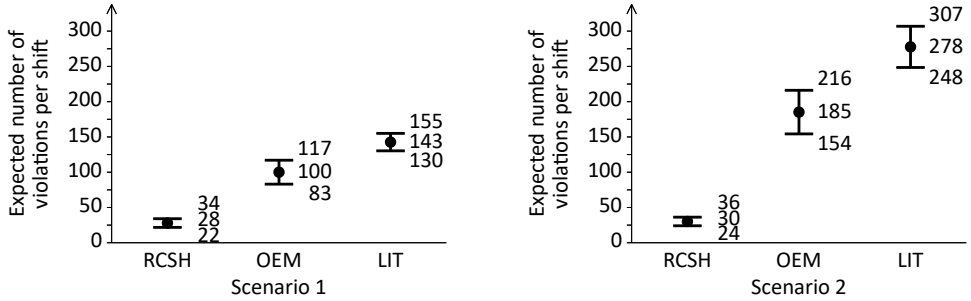


Figure 4.8: Means and 95% confidence intervals for expected number of violations in Scenario 1 (left) and Scenario 2 (right). Mean performance of OEM in Scenario 1 is normalized to 100.

4.6.2.3 Scenario 1: Current situation at partner OEM

Scenario 1 is the base case and represents the current situation at the OEM. On the left-hand side of Figure 4.8, we show the results of our simulation analysis. We plot the average number of violations in the final sequences across all instances as well as the 95% confidence intervals for the three approaches. For confidentiality reasons, we show normalized results. The mean performance of OEM in Scenario 1 is set to 100. All other values are scaled accordingly.

We see that RCSH outperforms the other approaches. OEM comes in second. As expected, LIT performs worst, because it does not anticipate vehicle failures. The final sequences of RCSH have on average 72% fewer violations than the final sequences of OEM. Compared to LIT, we see an average improvement of 80%. OEM, in turn, performs 30% better than LIT. Since the confidence intervals are not overlapping, all performance differences are significant on the 5% level.

We assume that the reason for OEM outperforming LIT is an approach which spaces the options across the sequence. Nevertheless, the comparison with RCSH proves that the full potential for robustness has not yet been tapped in industry. Also, OEM exhibits the highest fluctuation in performance across the instances, as demonstrated by the widest confidence interval. For RCSH, in contrast, we find the narrowest confidence interval and thus the most stable results.

4.6.2.4 Scenario 2: Introduction of a new color

In Scenario 2, we study the introduction of a new color. Both the demand for the new color and its failure probability are high. The right-hand side of Figure 4.8 shows the results of our simulation analysis. We find similar relations as in Scenario 1. RCSH generates by far the best results, followed by OEM. LIT comes in last.

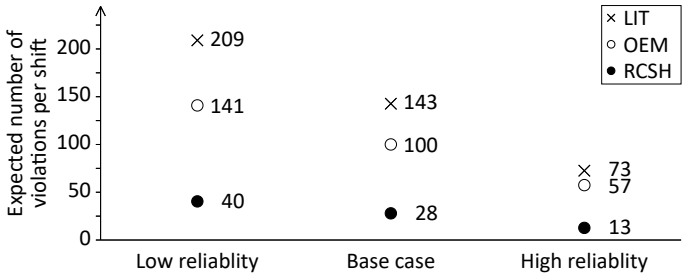


Figure 4.9: Effect of paint shop reliability on mean expected number of violations. Mean performance of OEM in base case is normalized to 100.

Compared to Scenario 1, we observe that the average number of violations increases for all three approaches. Also, the fluctuations in performance across the instances, as represented by the widths of the confidence intervals, increase. However, the performance deterioration is marginal for RCSH compared to the other approaches. Consequently, the benefits of using RCSH are higher in Scenario 2 than in Scenario 1. On average, RCSH generates 84% and 89% fewer violations than OEM and LIT respectively.

We conclude that OEM and LIT are unable to generate robust sequences for the challenging situation of launching a new color. Our RCSH, on the other hand, is adequate for addressing this challenge.

4.6.2.5 Scenario 3: Sensitivity toward paint shop reliability

In Scenario 3, we investigate the sensitivity of our results with regard to paint shop reliability. Paint shop reliability affects the failure probabilities of all vehicles. Figure 4.9 summarizes our findings. We plot the expected number of violations for different levels of paint shop reliability. We note a negative correlation between the reliability of the paint shop and the expected number of violations for all three approaches. The lower the paint shop reliability, the higher the expected number of violations. We observe the same ranking of approaches as in the previous scenarios. RCSH outperforms OEM and LIT for all levels of reliability. We note consistent improvements of above 70% compared to OEM. We conclude that our approach is valuable for all levels of paint shop reliability, even if the absolute number of violations that can be avoided by our approach is of course highest when paint shop reliability is low.

4.7 Conclusion

In this chapter, we studied a real-world car-sequencing problem. The goal was to generate assembly sequences that are robust in the event of vehicle failures. We formulated the problem as a mixed-integer non-linear program and proposed an exact B&B algorithm. In order to solve large-sized instances, we developed a sampling-based adaptive large neighborhood search heuristic. We showed that our heuristic generates high-quality solutions for real-world instances in acceptable run times. In an extensive simulation study, we validated the applicability of our approach. We studied three scenarios. Scenario 1 is the base case that represents the current situation at our partner OEM. In Scenario 2, we studied the performance of our approach when a new color is launched. In Scenario 3, we assessed the sensitivity of our results with regard to the reliability of the paint shop. In our study, we considered paint defects as the key failure driver. However, the algorithms proposed are independent of the failures considered and the application to other failure types is straightforward.

We derive valuable managerial insights from the results of our simulation study. The most important insight is that it is beneficial to consider the vehicles' failure probabilities in sequence planning in order to reduce expected work overloads. Compared to the industry solution and a literature approach that does not account for failures, we noticed improvements of 72% and 80% respectively. The improvements were significant on the 5% level. The results of our approach were not only better than the other approaches, they were also more stable. We noted fewer fluctuations in the anticipated work overloads.

The results for Scenario 2 showed that our approach can be of particular advantage when new colors are introduced. While the results of our approach were only marginally affected by the changed input data, the other approaches performed substantially worse. Thus, the benefits of our approach were higher in Scenario 2 compared to Scenario 1. We expect to see similar advantages for the launches of new options or even new models. Our approach can countervail the reduced stability that comes along with new processes, and the planned sequences are more robust.

The sensitivity analysis in Scenario 3 revealed that the relative benefits of our approach are consistent for different paint shop reliabilities, with improvements of above 70% compared to the industry approach. In absolute terms, though, OEMs would benefit most from our approach when paint shop reliability is low.

Since this is a first attempt to include the vehicles' failure probabilities in sequence planning, further research is required. Our experiments are based on the data provided

by our partner OEM, and we exclusively considered paint failures. To verify the applicability for other OEMs and for other failures, our approach should be tested on more and differently structured instances. Moreover, we considered sequence planning for a single shift. In reality, OEMs operate the final assembly in different shift schemes. To apply our algorithm for cases with continuous operations across shifts, our RCSH can be embedded in a rolling-horizon planning framework.

5 Conclusion

This chapter provides a summary of the research presented in the previous chapters and discusses the findings with regard to the research questions outlined in Section 1.3. Furthermore, we discuss directions for future research.

5.1 Summary

The Industry 4.0 revolution entails big challenges but also great opportunities for automotive OEMs. Transforming their production facilities into smart factories allows OEMs to cope with increasing vehicle heterogeneity that arises from introducing alternative drivetrain technologies to the product mix. Innovative FALs and data-driven planning algorithms are key enablers in such a smart factory. Both support OEMs in simultaneously improving efficiency, flexibility, and robustness, and thereby staying competitive in an increasingly dynamic and uncertain market environment.

This thesis aims at supporting OEMs in the transformation to a smart factory by answering the research questions outlined in Section 1.3. We contribute to both the research on FAL design and on data-driven sequencing algorithms for LALs. With our studies, we aim to foster the liaison between academic research and industrial practice. Our goal is to guide industrial practice by providing quantifiable scientific evidence.

In the following, we first summarize the findings for each of the research questions and provide a comprehensive conclusion at the end of the section. We answer research question 1 by providing detailed results on its deduced subquestions 1.A and 1.B.

RQ 1.A: How to strategically design FALs for the automotive assembly?
Chapter 2 investigated the strategic design of FALs. We formally defined the FAL design problem as a lexicographic MILP that comprises a station formation, station location, and flow allocation problem. The primary objective is to minimize the number of stations, which is equivalent to maximizing efficiency. The secondary objective is to minimize flow intensity. We developed an exact decomposition-based solution algorithm as well as an iterative fix-optimize matheuristic. In our computational study, we showed

that our matheuristic is capable of finding very good solutions in acceptable time for instances of industrial size.

We discovered two consistent design characteristics in the generated FALs. First, the obtained FALs are compact, i.e., they are typically not much longer than they are wide. Second, they are characterized by centralization, which means that the entry and exit points are usually positioned on the central axis. We noticed high utilization for the stations that are on this main axis between entry and exit points, whereas the stations in the outer parts of the layout are less utilized. Also, we found that most units of the same model follow the same route through the layout.

RQ 1.B: How to tactically configure FALs for the automotive assembly?

Chapter 3 focused on the tactical configuration of FALs. On this level, the OEM is confronted with a flexibility configuration problem. That is, the OEM has to decide on the exploitation of operation and routing flexibility as well as on an appropriate WIP target for the FAL segment. In general, OEMs strive for a low WIP to simplify AGV routing and reduce the space requirements of the FAL. However, a high WIP constitutes as an additional flexibility lever that facilitates scheduling on the operational level. Exploiting operation and routing flexibility allows to reduce the WIP without compromising operational performance. Nevertheless, operation and routing flexibility also have disadvantages. Operation flexibility comes along with variable task sequences that may confuse workers, and routing flexibility implies alternative task locations that preclude JIS stocking at stations. OEMs must consider these disadvantages when deciding on the right flexibility configuration of an FAL in practice.

We formulated the flexibility configuration problem in an FAL as a chance-constrained optimization problem and proposed a problem-specific decomposition. We then developed an exact B&P algorithm to solve the decomposed subproblems. In an extensive computational analysis, we quantified the effect of an FAL's flexibility levers on operational performance, and we derived valuable managerial insights.

We observed a clear impact hierarchy for the flexibility levers in an FAL. While routing flexibility is the main improvement lever, operation flexibility reveals significantly lower improvement potentials. Interestingly, we found that both flexibility levers reinforce each other such that exploiting them together leads to higher benefits than the sum of the individual benefits. Additionally, we showed that these flexibility levers can resolve the well-known trade-off between increasing a layout's utilization and reducing its WIP, as exploiting operation and routing flexibility improves both objectives simultaneously.

RQ 2: What are the advantages and disadvantages of FALs compared to LALs? For which application scenarios are FALs superior to LALs?

In Chapters 2 and 3, we compared FALs to LALs. Chapter 2 studied the effect of vehicle heterogeneity on the efficiency of both FALs and LALs, whereas Chapter 3 compared worker utilization, output levels, and WIP. From our computational analyses, we extracted several managerial insights.

In Chapter 2, we confirmed that FALs have an efficiency advantage compared to LALs. This efficiency advantage depends on the extend of drifting in LALs. Compared to LALs with closed stations, we computed efficiency gains of nearly 25%, which match industry predictions. For LALs with open stations, the efficiency gain of FALs diminishes. In addition, we found that the efficiency of FALs is insensitive to vehicle heterogeneity. The efficiency of LALs, in contrast, declines with higher vehicle heterogeneity. Consequently, we concluded that the attractiveness of FALs increases with higher vehicle heterogeneity.

In Chapter 3, we proved that FALs outperform LALs in terms of utilization and output level. Our results showed that FALs achieve up to 30% higher utilization and output levels compared to LALs with closed stations. When comparing to LALs with open stations, these improvements decrease. However, they still remain significant, even when operation and routing flexibility are not exploited in FALs. These results are in line with our findings from Chapter 2. The FAL improvements come at the price of a higher WIP. We saw that this WIP increase depends on the flexibility configuration of the FAL and the efficiency of the AGV system employed. When operation and routing flexibility are exploited and the AGV transports are fast, the WIP disadvantage of FALs is low. However, it increases considerably for configurations without routing flexibility and with slower AGV transports. Finally, we confirmed that FALs can be of particular advantage during ramp-up stages for new technologies. In our analyses, we found that FALs can accommodate changing demand mixes and achieve stable utilization and output levels by slightly adapting the segment cycle time. For LALs, in contrast, both performance measures deteriorate, even when overcapacities are considered in their design.

RQ 3: How to increase the robustness of sequence planning for conventional LALs?

In Chapter 4, we studied a real-world car-sequencing problem focusing on sequence stability in conventional LALs. We proposed a mixed-integer non-linear problem formulation and developed an exact B&B algorithm. Since the problem is characterized by a high degree of symmetry, we suggested problem-specific symmetry breaking constraints. Moreover, we derived tight problem-specific lower bounds to accelerate the termination of our B&B algorithm. These bounds are based on individual options and the prob-

lem's deterministic counterpart. Additionally, we developed a sampling-based ALNS heuristic to solve problem instances of real-world size. Our ALNS heuristic is inspired by insights we obtained from studying optimal solutions to small-sized instances. We employed descriptive sampling as introduced by Saliby (1990) to reduce the variability of the results. Our computational assessment showed that our heuristic is capable of solving real-world instances in acceptable run times. In an extensive simulation study, we validated the applicability of our approach by comparing our results to the industry solution and to a literature approach that does not account for vehicle failures. In our study, we considered paint defects as the key failure driver. However, the algorithms proposed are independent of the failures considered, and the application to other failure types is straightforward.

We proved that considering the vehicles' failure probabilities in sequence planning is beneficial in order to reduce expected work overloads, and hence improve sequence stability. We computed significant improvements of 72% and 80% compared to the industry solution and to the literature approach respectively. Moreover, we found that our heuristic generates more stable results than the other two approaches. We showed that our approach is particularly beneficial during the introduction of new colors. We expect similar benefits for the ramp-up of new options or even new models. Our approach can countervail the reduced stability that comes along with new processes, and the planned sequences are more robust. Finally, we found that the relative benefits of our approach are independent of the paint shop reliability. In absolute terms, however, OEMs benefit most from our approach when paint shop reliability is low.

Overall conclusion

In summary, this thesis provides important insights into the smart factory transformation in the automotive industry. We proposed quantitative planning approaches for designing new, innovative FALs and enhancing sequence planning for conventional LALs. We developed three data-driven planning approaches at different levels: *i)* a layout design approach at strategic level, *ii)* a layout configuration approach at tactical level, and *iii)* a sequencing approach at operational level.

We employed mixed-integer (non-)linear programming as methodology. With the approaches that we developed, we contribute to a multitude of algorithmic concepts, both exact and heuristic. As exact algorithms, we developed a decomposition-based algorithm, as well as tailored B&B and B&P algorithms. As heuristics, we created an iterative fix-optimize matheuristic and an ALNS metaheuristic. We applied our approaches to either real-world industry cases or we adapted popular standard data sets

from literature. Our computational analyses confirmed that the developed approaches are well-suited to solve large-sized, real-world problem instances. From our results, we derived valuable managerial insights related to FALs and LALs in the automotive assembly.

We provided industry guidance into the optimal design and configuration of FALs, and we generated quantifiable scientific evidence on the advantages and disadvantages of FALs compared to LALs. We showed that FALs have advantages in efficiency, worker utilization, and output level, but require a higher WIP and are more complex to plan and control. Moreover, we investigated the appropriate application scenarios of both layout types. We found that FALs become more beneficial with higher vehicle heterogeneity and frequent changes in the demand mix, e.g., during ramp-ups. LALs, in contrast, are attractive when producing stable, homogeneous product mixes, and when workers are allowed to drift into subsequent stations. These insights are highly useful for OEMs that consider to replace parts of their assembly line by an FAL.

Furthermore, we contributed to a robust sequence planning for conventional LALs. LALs continue to be of high relevance in the automotive industry, either in pure line layouts or as part of mixed layouts with FALs. Our robust car-sequencing approach enables a reliable supplier signal and thereby facilitates efficient JIS material supply. We tested our data-driven optimization approach on real-world data from a major European OEM and showed significant improvement potentials. Thereby, the OEM can increase efficiency, lower operational cost, and improve product quality, which in turn are pivotal success factors in today's competitive automotive market environment.

5.2 Future research directions

Given the novelty of the presented problems and the lack of previous academic research, this thesis provides a common ground for future research on the smart factory transformation in the automotive assembly. While we have answered many fundamental questions in this thesis, several promising opportunities for future research arise. We already suggested specific research topics in the previous chapters, and therefore outline more general research directions in this section. We sequence them by hierarchical level, starting at the strategic level and ending at the operational level.

We investigated new layouts for the automotive assembly, in which the vehicles can move flexibly between assembly stations while the workers are confined to these stations. An alternative idea is to maintain the serial, paced workflow of the vehicles as in an LAL, but lift the restriction of fixed assembly stations with confined workers. Instead,

tasks can be performed continuously along a U-shaped line and workers can move freely between different vehicles. While such layouts reduce the complexity in the vehicle flow, it becomes more challenging to manage the operations of the workers. This alternative concept to increase flexibility is currently also discussed among automotive practitioners, however, a strategic proof of concept and a benchmarking against FALs are still missing.

Regarding FALs, there are several opportunities for research on the strategic level. The question on how to segment the final assembly into FALs and LALs is unsolved. Our results from Chapter 2 give indication that vehicle heterogeneity is a key determinant for deciding between an FAL and LAL for an assembly segment. However, multiple other influencing factors play a role, e.g., assembly technology (manual vs. automated), required tools, and material supply. For OEMs, it is crucial to have quantitative insights on the effects of these influencing factors. Ideally, an algorithm could suggest the appropriate segmentation of the final assembly into FALs and LALs based on the production scenario. In case enough real-world industry data is available for training, this resembles a machine learning problem, where the production scenario is represented by a set of features, and the decision on the layout concept represents the algorithm's output.

Furthermore, we focused on the static, deterministic FAL design in this thesis. In reality, OEMs operate in a dynamic, stochastic market environment. We already mentioned that ramp-ups are frequent. Moreover, the demand mix bears uncertainties. Consequently, it is appropriate to consider robustness as an additional objective during FAL design. Two-stage stochastic programming and robust optimization may be useful methodologies in this context. Experts expect that FALs have even greater advantages compared to LALs when confronted with a dynamic, uncertain market environment.

A quantitative analysis on the benefits of adaptability in FALs is another open research topic. We considered green-field planning in this thesis, because FALs are a novel layout concept. Nevertheless, brown-field planning, i.e., FAL reconfiguration, is worth to be investigated. Hereby, OEMs trade-off increased efficiency against reconfiguration effort. FALs are easier to be reconfigured than LALs, because tasks can be reassigned and new stations can be added alongside the layout without interrupting production at the other stations. This adaptability is beneficial when new products are being introduced or when significant demand changes occur.

The operational level offers interesting research possibilities as well. While we investigated the strategic design and tactical configuration of FALs, the operational scheduling is only anticipated in our tactical planning. OEMs require intelligent, data-driven algorithms to optimize the real-time scheduling and routing of the AGVs. The operational planning is challenging, because it is highly dynamic and uncertain. Task execution

times are stochastic by nature and disruptions may occur. Therefore, the scheduling and routing need to be re-optimized continuously based on the system state. Since decisions have to be taken within a few seconds, iteratively applied look-ahead heuristics appear to be promising. The performance of these heuristics could be validated in simulation studies for a wide range of parameter settings.

The operational sequence planning for mixed layouts consisting of FAL and LAL segments is another unsolved problem. However, it should be similar to sequence planning in pure LALs, because we design FALs such that they can cope with any sequence permutation. In general, sequence planning needs to be integrated in a rolling-horizon framework to avoid deteriorative effects at the beginning and end of the planning horizon.

All our results are based on either data of a single OEM or on adapted instances from literature. To ensure the general applicability, our approaches should be verified using real-world data of other OEMs.

References

Anand, G., & Ward, P. T. (2004). Fit, flexibility and performance in manufacturing: Coping with dynamic environments. *Production and Operations Management*, *13*(4), 369–385.

Aneke, N. A. G., & Carrie, A. S. (1986). A design technique for the layout of multi-product flowlines. *International Journal of Production Research*, *24*(3), 471–481.

Askin, R. G., & Mitwasi, M. (1992). Integrating facility layout with process selection and capacity planning. *European Journal of Operational Research*, *57*(2), 162–173.

Bard, J. F. (1989). Assembly line balancing with parallel workstations and dead time. *International Journal of Production Research*, *27*(6), 1005–1018.

Bassamboo, A., Randhawa, R. S., & van Mieghem, J. A. (2010). Optimal flexibility configurations in newsvendor networks: Going beyond chaining and pairing. *Management Science*, *56*(8), 1285–1303.

Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, *168*(3), 694–715.

Benjaafar, S., & Sheikhzadeh, M. (2000). Design of flexible plant layouts. *IIE Transactions*, *32*(4), 309–322.

Benoist, T. (2008). Soft car sequencing with colors: Lower bounds and optimality proofs. *European Journal of Operational Research*, *191*(3), 957–971.

Bolat, A., & Yano, C. A. (1992a). Scheduling algorithms to minimize utility work at a single station on a paced assembly line. *Production Planning & Control*, *3*(4), 393–405.

Bolat, A., & Yano, C. A. (1992b). A surrogate objective for utility work in paced assembly lines. *Production Planning & Control*, *3*(4), 406–412.

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

Boysen, N., Fliedner, M., & Scholl, A. (2009). Production planning of mixed-model assembly lines: Overview and extensions. *Production Planning & Control*, *20*(5), 455–471.

Boysen, N., Golle, U., & Rothlauf, F. (2011). The car resequencing problem with pull-off tables. *Business Research*, *4*(2), 276–292.

Boysen, N., Scholl, A., & Wopperer, N. (2012). Resequencing of mixed-model assembly lines: Survey and research agenda. *European Journal of Operational Research*, *216*(3), 594–604.

Brailsford, S. C., Potts, C. N., & Smith, B. M. (1999). Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, *119*(3), 557–581.

Briant, O., Naddef, D., & Mounié, G. (2008). Greedy approach and multi-criteria simulated annealing for the car sequencing problem. *European Journal of Operational Research*, *191*(3), 993–1003.

Browne, J., Dubois, D., Rathmill, K., Sethi, S. P., & Stecke, K. E. (1984). Classification of flexible manufacturing systems. *The FMS magazine*, *2*(2), 114–117.

Bukchin, J., Dar-El, E. M., & Rubinovitz, J. (2002). Mixed model assembly line design in a make-to-order environment. *Computers & Industrial Engineering*, *41*(4), 405–421.

Bukchin, Y., & Rabinowitch, I. (2006). A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs. *European Journal of Operational Research*, *174*(1), 492–508.

Caux, C., Bruniaux, R., & Pierreval, H. (2000). Cell formation with alternative process plans and machine capacity constraints: A new combined approach. *International Journal of Production Economics*, *64*(1-3), 279–284.

Chen, D.-S., Wang, Q., & Chen, H.-C. (2001). Linear sequencing for machine layouts by a modified simulated annealing. *International Journal of Production Research*, *39*(8), 1721–1732.

Choi, W., & Shin, H. (1997). A real-time sequence control system for the level production of the automobile assembly line. *Computers & Industrial Engineering*, *33*(3-4), 769–772.

Cordeau, J.-F., Laporte, G., & Pasin, F. (2008). Iterated tabu search for the car sequencing problem. *European Journal of Operational Research*, *191*(3), 945–956.

Defersha, F. M., & Hodiya, A. (2017). A mathematical model and a parallel multiple search path simulated annealing for an integrated distributed layout design and machine cell formation. *Journal of Manufacturing Systems*, *43*, 195–212.

Ding, F.-Y., & Sun, H. (2004). Sequence alteration and restoration related to sequenced parts delivery on an automobile mixed-model assembly line with multiple departments. *International Journal of Production Research*, *42*(8), 1525–1543.

Drexl, A., & Kimms, A. (2001). Sequencing JIT mixed-model assembly lines under station-load and part-usage constraints. *Management Science*, *47*(3), 480–491.

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, *31*(2), 255–267.

Estellon, B., Gardi, F., & Nouioua, K. (2008). Two local search approaches for solving real-life car sequencing problems. *European Journal of Operational Research*, *191*(3), 928–944.

Gagné, C., Gravel, M., & Price, W. L. (2006). Solving real car sequencing problems with ant colony optimization. *European Journal of Operational Research*, *174*(3), 1427–1448.

Goldengorin, B., Krushinsky, D., & Pardalos, P. M. (2013). *Cell formation in industrial engineering* (Vol. 79). New York, NY: Springer.

Golle, U., Boysen, N., & Rothlauf, F. (2010). Analysis and design of sequencing rules for car sequencing. *European Journal of Operational Research*, *206*(3), 579–585.

Golle, U., Rothlauf, F., & Boysen, N. (2014). Car sequencing versus mixed-model sequencing: A computational study. *European Journal of Operational Research*, *237*(1), 50–61.

Gravel, M., Gagné, C., & Price, W. L. (2005). Review and comparison of three methods for the solution of the car sequencing problem. *Journal of the Operational Research Society*, *56*(11), 1287–1295.

Graves, S. C., & Tomlin, B. T. (2003). Process flexibility in supply chains. *Management Science*, *49*(7), 907–919.

Graves, S. C., & Willems, S. P. (2000). Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management*, *2*(1), 68–83.

Gusikhin, O., Caprihan, R., & Stecke, K. E. (2008). Least in-sequence probability heuristic for mixed-volume production lines. *International Journal of Production Research*, *46*(3), 647–673.

Hahn, R. (1972). *Produktionsplanung bei Linienfertigung.* Berlin: De Gruyter.

Heragu, S. S., & Chen, J.-S. (1998). Optimal solution of cellular manufacturing system design: Benders' decomposition approach. *European Journal of Operational Research*, *107*(1), 175–192.

Heragu, S. S., & Kusiak, A. (1988). Machine layout problem in flexible manufacturing systems. *Operations Research*, *36*(2), 258–268.

Hindi, K. S., & Ploszajski, G. (1994). Formulation and solution of a selection and sequencing problem in car manufacture. *Computers & Industrial Engineering*, *26*(1), 203–211.

Ho, Y.-C., & Moodie, C. L. (1998). Machine layout with a linear single-row flow path in an automated manufacturing system. *Journal of Manufacturing Systems*, *17*(1), 1–22.

Hoffmann, T. R. (1992). EUREKA: A hybrid system for assembly line balancing. *Management Science*, *38*(1), 39–47.

Hopp, W. J., Iravani, S. M. R., & Xu, W. L. (2010). Vertical flexibility in supply chains. *Management Science*, *56*(3), 495–502.

Hottenrott, A., & Grunow, M. (2019). Flexible layouts for the mixed-model assembly of heterogeneous vehicles. *OR Spectrum*, *41*(4), 943–979.

Humair, S., & Willems, S. P. (2006). Optimizing strategic safety stock placement in supply chains with clusters of commonality. *Operations Research*, *54*(4), 725–742.

Inman, R. R. (2003). ASRS sizing for recreating automotive assembly sequences. *International Journal of Production Research*, *41*(5), 847–863.

Jaramillo, J. R., & McKendall, A. R. (2010). The generalised machine layout problem. *International Journal of Production Research*, *48*(16), 4845–4859.

Jordan, W. C., & Graves, S. C. (1995). Principles on the benefits of manufacturing process flexibility. *Management Science*, *41*(4), 577–594.

Keller, B., & Buscher, U. (2015). Single row layout models. *European Journal of Operational Research*, *245*(3), 629–644.

Kis, T. (2004). On the complexity of the car sequencing problem. *Operations Research Letters*, *32*(4), 331–335.

Lehmann, M., & Kuhn, H. (2020). Modeling and analyzing sequence stability in flexible automotive production systems. *Flexible Services and Manufacturing Journal*, *32*(2), 366–394.

Lenstra, J. K., Rinnooy Kan, A., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, *1*, 343–362.

Lesert, A., Alpan, G., Frein, Y., & Noiré, S. (2011). Definition of spacing constraints for the car sequencing problem. *International Journal of Production Research*, *49*(4), 963–994.

Li, J., & Gao, J. (2014). Balancing manual mixed-model assembly lines using overtime work in a demand variation environment. *International Journal of Production Research*, *52*(12), 3552–3567.

Mastrolilli, M., & Gambardella, L. M. (2000). Effective neighbourhood functions for the flexible job shop problem. *Journal of Scheduling*, *3*(1), 3–20.

Meissner, S. (2010). Controlling just-in-sequence flow-production. *Logistics Research*, *2*(1), 45–53.

Meyr, H. (2004). Supply chain planning in the German automotive industry. *OR Spectrum*, *26*(4), 447–470.

Michalos, G., Makris, S., Papakostas, N., Mourtzis, D., & 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.

Mohammadi, G., Karampourhaghghi, A., & Samaei, F. (2012). A multi-objective optimisation model to integrating flexible process planning and scheduling based on hybrid multi-objective simulated annealing. *International Journal of Production Research*, *50*(18), 5063–5076.

Montreuil, B. (1999). Fractal layout organization for job shop environments. *International Journal of Production Research*, *37*(3), 501–521.

Müller, M., Lehmann, M., & Kuhn, H. (2020). Measuring sequence stability in automotive production lines. *International Journal of Production Research*, *4*, 1–21.

Muriel, A., Somasundaram, A., & Zhang, Y. (2006). Impact of partial manufacturing flexibility on production variability. *Manufacturing & Service Operations Management*, *8*(2), 192–205.

Nagi, R., Harhalakis, G., & Proth, J.-M. (1990). Multiple routeings and capacity considerations in group technology applications. *International Journal of Production Research*, *28*(12), 1243–1257.

Olsen, T. L., & Tomlin, B. (2020). Industry 4.0: Opportunities and challenges for operations management. *Manufacturing & Service Operations Management*, *22*(1), 113–122.

Papaioannou, G., & Wilson, J. M. (2010). The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research. *European Journal of Operational Research*, *206*(3), 509–521.

Pil, F. K., & Holweg, M. (2004). Linking product variety to order-fulfillment strategies. *Interfaces*, *34*(5), 394–403.

Rajamani, D., Singh, N., & Aneja, Y. P. (1990). Integrated design of cellular manufacturing systems in the presence of alternative process plans. *International Journal of Production Research*, *28*(8), 1541–1554.

Ribeiro, C. C., Aloise, D., Noronha, T. F., Rocha, C., & Urrutia, S. (2008). A hybrid heuristic for a multi-objective real-life car sequencing problem with painting and assembly line constraints. *European Journal of Operational Research*, *191*(3), 981–992.

Ribeiro, G. M., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, *39*(3), 728–735.

Roberts, S. D., & Villa, C. D. (1970). On a multiproduct assembly line-balancing problem. *AIIE Transactions*, *2*(4), 361–364.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, *40*(4), 455–472.

Saliby, E. (1990). Descriptive sampling: A better approach to Monte Carlo simulation. *Journal of the Operational Research Society*, *41*(12), 1133–1142.

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. (1993). Data of assembly line balancing problems. *Schriften zur Quantitativen Betriebswirtschaftslehre - Technische Hochschule Darmstadt*.

Simchi-Levi, D., Wang, H., & Wei, Y. (2018). Increasing supply chain robustness through process flexibility and inventory. *Production and Operations Management*, *27*(8), 1476–1491.

Sofianopoulou, S. (1999). Manufacturing cells design with alternative process plans and/or replicate machines. *International Journal of Production Research*, *37*(3), 707–720.

Solimanpur, M., Vrat, P., & Shankar, R. (2004). A multi-objective genetic algorithm approach to the design of cellular manufacturing systems. *International Journal of Production Research*, *42*(7), 1419–1441.

Solnon, C. (2008). Combining two pheromone structures for solving the car sequencing problem with ant colony optimization. *European Journal of Operational Research*, *191*(3), 1043–1055.

Solnon, C., van Cung, D., Nguyen, A., & Artigues, C. (2008). The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. *European Journal of Operational Research*, *191*(3), 912–927.

Taube, F., & Minner, S. (2018). Resequencing mixed-model assembly lines with restoration to customer orders. *Omega*, *78*, 99–111.

Tomlin, B., & Wang, Y. (2005). On the value of mix flexibility and dual sourcing in unreliable newsvendor networks. *Manufacturing & Service Operations Management*, *7*(1), 37–57.

Urban, T. L., Chiang, W.-C., & Russell, R. A. (2000). The integrated machine allocation and layout problem. *International Journal of Production Research*, *38*(13), 2911–2930.

van Mieghem, J. A. (2007). Risk mitigation in newsvendor networks: Resource diversification, flexibility, sharing, and hedging. *Management Science*, *53*(8), 1269–1288.

van Mieghem, J. A., & Rudi, N. (2002). Newsvendor networks: Inventory management and capacity investment with discretionary activities. *Manufacturing & Service Operations Management*, *4*(4), 313–335.

Wagner, S. M., & Silveira-Camargos, V. (2012). Managing risks in just-in-sequence supply networks: Exploratory evidence from automakers. *IEEE Transactions on Engineering Management*, *59*(1), 52–64.

Warwick, T., & Tsang, E. P. K. (1995). Tackling car sequencing problems using a generic genetic algorithm. *Evolutionary Computation*, *3*(3), 267–298.

Zhang, L., & Wong, T. N. (2015). An object-coding genetic algorithm for integrated process planning and scheduling. *European Journal of Operational Research*, *244*(2), 434–444.

A Appendices of Chapter 2

A.1 Instance generation scheme

Input : Single-model precedence graph from data set of Scholl (1993)
Output: 16 multi-model instances with different levels of structure (sh) and task time (tth) heterogeneity

Randomly generate demand d_m for five models;
Generate base case: Precedence graphs and task times of all five models are identical to data of Scholl (1993);
Determine overall workload per task in base case W_t;
for $sh \in [0\%, 10\%, 25\%, 50\%]$ **do**
  Randomly remove sh of task-model nodes in precedence graphs of the base case such that each task is needed by at least one model and that each model needs at least two real tasks;
  Calculate demand for each task: $\hat{d}_t = \sum_{m \in M | t \in T_m} d_m$;
  Calculate average task times: $\bar{q}_t = \frac{W_t}{\hat{d}_t}$;
  for $tth \in [0\%, 10\%, 25\%, 50\%]$ **do**
    Solve LP

$$\max \; Z^{LP} = \sum_{m \in M | t \in T_m} d_m \left| q_{m,t} - \bar{q}_t \right| \tag{A.1a}$$

s.t.

$$\sum_{m \in M | t \in T_m} d_m q_{m,t} = W_t \qquad \forall t \in T \tag{A.1b}$$

$$(1 - tth)\bar{q}_t \leq q_{m,t} \leq (1 + tth)\bar{q}_t \qquad \forall m \in M, t \in T_m \tag{A.1c}$$

    Decision variables $q_{m,t}$: Task time of task t for model m,
    Objective function (A.1a): Maximize task time heterogeneity,
    Constraints (A.1b): Ensure that overall workload for each task is identical to base case,
    Constraints (A.1c): Define heterogeneity range of task times;
    Retrieve task times of all models $q_{m,t}$ from LP solution;
    Save instance;
  end
end

Figure A.1: Pseudocode of instance generation scheme.

A.2 Benchmark mixed-model assembly line balancing model

We adapt the MMAL balancing model formulation by Y. Bukchin and Rabinowitch (2006). Table A.1 shows the required additional notation. We introduce a new index set $s \in S$ that represents the stations on the line. Parameter c denotes the cycle time. We add parameter $\alpha \geq 0$ to vary between closed and open stations. α represents the percentage of cycle time that workers are allowed to drift out into downstream stations. Next, we denote three sets of decision variables. The binary variable $\bar{X}_{t,m,s}$ shows whether task t for model m is assigned to station s. $\bar{Y}_{t,s}$ is a binary variable as well. It indicates whether task t is assigned to station s for any model. Last, the continuous variable $\bar{Z}$ represents the number of stations used and is to be minimized (A.2a).

Table A.1: Additional notation for MMAL balancing problem.

Index sets	
$s \in S$	Stations ($s = 1, \ldots, \lvert S \rvert$)
Parameters	
c	Cycle time
α	Drift factor: percentage of cycle time that workers are allowed to drift out into downstream stations
Decision variables	
$\bar{X}_{t,m,s}$	1 if task t of model m is assigned to station s, else 0
$\bar{Y}_{t,s}$	1 if task t of any model is assigned to station s, else 0
$\bar{Z}$	Number of stations to be used in LAL

Constraints (A.2b) force the workload of a task for a particular model to be assigned to a single station. Note again that splitting the workload of one model among task duplicates at different stations is not allowed in an LAL. Precedence relations are satisfied by Constraints (A.2c). Constraints (A.2d) limit the total processing time for each model at each station. For $\alpha = 0$, we evaluate closed stations, in which workers are not allowed to drift out into downstream stations. For $\alpha > 0$, workers are allowed to drift out into the neighboring stations by $\alpha\%$ of the cycle time. In order to make sure that the overall capacity of the station is not violated, we add Constraints (A.2e) to the model by Y. Bukchin and Rabinowitch (2006). The number of stations used is derived in Constraints (A.2f). Constraints (A.2g) link the two binary variables by checking whether a task is performed for any model at a certain station. As an extension to the model by Y. Bukchin and Rabinowitch (2006), we introduce Constraints (A.2h) that limit the maximum number of task duplicates. Without these constraints, the LAL solution would not be comparable to the FAL solution. Finally, in Constraints (A.2i) - (A.2k), we restrict the domains of the decision variables.

$$\min \; Z^L = \bar{Z} \tag{A.2a}$$

s.t.

$$\sum_{s\in S} \bar{X}_{t,m,s} = 1 \qquad \forall m\in M, t\in T_m \tag{A.2b}$$

$$\sum_{s_1\in S} s_1 \cdot \bar{X}_{t_1,m,s_1} \leq \sum_{s_2\in S} s_2 \cdot \bar{X}_{t_2,m,s_2} \qquad \forall m\in M, t_1\in T_m, t_2\in V_{m,t_1} \tag{A.2c}$$

$$\sum_{t\in T_m} q_{m,t} \cdot \bar{X}_{t,m,s} \leq c \cdot (1+\alpha) \qquad \forall m\in M, s\in S \tag{A.2d}$$

$$\sum_{m\in M} d_m \cdot \sum_{t\in T_m} q_{m,t} \cdot \bar{X}_{t,m,s} \leq \tau \qquad \forall s\in S \tag{A.2e}$$

$$\bar{Z} \geq \sum_{s\in S} s \cdot \bar{X}_{t,m,s} \qquad \forall m\in M, t\in T_m \tag{A.2f}$$

$$\bar{Y}_{t,s} \geq \frac{1}{|M|} \cdot \sum_{m\in M|t\in T_m} \bar{X}_{t,m,s} \qquad \forall t\in T, s\in S \tag{A.2g}$$

$$\sum_{s\in S} \bar{Y}_{t,s} \leq n_t \qquad \forall t\in T \tag{A.2h}$$

$$\bar{X}_{t,m,s} \in \{0,1\} \qquad \forall m\in M, t\in T_m, s\in S \tag{A.2i}$$

$$\bar{Y}_{t,s} \in \{0,1\} \qquad \forall t\in T, s\in S \tag{A.2j}$$

$$\bar{Z} \geq 0 \tag{A.2k}$$

B Appendices of Chapter 3

B.1 Proof of Theorem 1

To proof Theorem 1, we first analyze possible schedules for the no flexibility (NF) and full flexibility (FF) configurations of the respective minimal case (cf. Section 3.1.1)

In the NF configuration, four scheduling cases may exist, depending on the instance (cf. Figure B.1).

Case 1. The vehicles neither interfere at station $L1$ nor at $L2$: $c \geq q_{1A} \wedge c \geq q_{1A} + q_{1B} - q_{2A}$.

Case 2. The vehicles interfere at station $L1$ but not at station $L2$: $c < q_{1A} \wedge q_{2A} \geq q_{1B}$.

Case 3. The vehicles interfere at station $L2$ but not at station $L1$: $c \geq q_{1A} \wedge c < q_{1A} + q_{1B} - q_{2A}$.

Case 4. The vehicles interfere at both station $L1$ and station $L2$: $c < q_{1A} \wedge q_{2A} < q_{1B}$.

Based on these cases, we derive the required segment cycle time for the NF configuration C^{NF} as the maximum of the required segment cycle times of both vehicles C_{V1}^{ReqNF}, C_{V2}^{ReqNF} , formally

$$\begin{aligned} C^{NF} &= \max\{C_{V1}^{ReqNF}; C_{V2}^{ReqNF}\} \\ &= \max\{q_{1A} + \sigma + q_{1B}; \max\{\max\{c; q_{1A}\} + q_{2A}; q_{1A} + q_{1B}\} + \sigma + q_{2B} - c\}. \end{aligned} \tag{B.1}$$

In the FF configuration, there always exists an optimal schedule in which both vehicles do not interfere (cf. Figure B.2), and we can formalize the required segment cycle time for the FF configuration as

$$C^{FF} = \max\{C_{V1}^{ReqFF}; C_{V2}^{ReqFF}\} = \max\{q_{1A} + q_{1B} + \sigma; q_{2A} + q_{2B} + \sigma\}. \tag{B.2}$$

We can now determine minimum and maximum bounds on flexibility benefits.

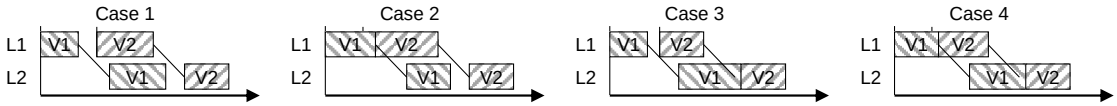


Figure B.1: Scheduling cases in NF configuration.

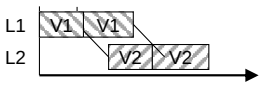


Figure B.2: Scheduling case in FF configuration.

Minimum bounds

For Case 1 in the NF configuration, the vehicles do not interfere, and hence, the WIP and the utilization are the same in the NF and FF configurations. The same holds if vehicle $V1$ sets the segment cycle time in the NF configuration, i.e., $C_{V1}^{ReqNF} \geq C_{V2}^{ReqNF}$.

Maximum bounds

In order to conservatively approximate a maximum bound, we study Case 4 of the NF configuration and note that Cases 2 and 3 are special variants of Case 4. We make the following four observations:

(i) We can only benefit through flexibility if vehicle $V2$ sets the segment cycle time in the NF configuration, i.e., $C_{V2}^{ReqNF} \geq C_{V1}^{ReqNF}$.

(ii) Regarding the FF configuration, we can realize the highest flexibility benefits if vehicle $V2$ sets the segment cycle time, i.e., $C_{V2}^{ReqFF} \geq C_{V1}^{ReqFF}$. We then know that $q_{2A} + q_{2B} \geq q_{1A} + q_{1B}$.

(iii) We can realize the highest flexibility benefits if both transportation times ($\sigma \to 0$) and the arrival time of vehicle $V2$ ($c \to 0$) are negligible.

(iv) We can define q_{2B} depending on q_{2A}, formally, $q_{2B} = \Psi q_{2A}$, with $\Psi \in \mathbb{R}^{\geq 0}$.

With observations (i) and (ii), we refine Equation (B.1) to (B.3) and Equation (B.2) to (B.4).

$$C^{NF} \overset{(i)}{=} C_{V2}^{ReqNF} = q_{1A} + q_{1B} + \sigma + q_{2B} - c \tag{B.3}$$

$$C^{FF} \overset{(ii)}{=} C_{V2}^{ReqFF} = q_{2A} + q_{2B} + \sigma \tag{B.4}$$

Now, we derive the changes in WIP and utilization as shown in Equations (B.5) and (B.6) respectively.

$$\begin{aligned}
\Delta^{WIP} &= \frac{WIP^{FF}}{WIP^{NF}} = \frac{\frac{C^{FF}}{c}}{\frac{C^{NF}}{c}} = \frac{C^{FF}}{C^{NF}} = \frac{q_{2A}+q_{2B}+\sigma}{q_{1A}+q_{1B}+\sigma+q_{2B}-c} \\
&\overset{(ii)}{\geq} \frac{q_{2A}+q_{2B}+\sigma}{q_{2A}+q_{2B}+\sigma+q_{2B}-c} = \frac{q_{2A}+q_{2B}+\sigma}{q_{2A}+2q_{2B}+\sigma-c} \\
&\overset{(iii)}{\geq} \frac{q_{2A}+q_{2B}}{q_{2A}+2q_{2B}} \\
&\overset{(iv)}{=} \frac{q_{2A}+\Psi q_{2A}}{q_{2A}+2\Psi q_{2A}} = \frac{1+\Psi}{1+2\Psi} \Rightarrow \Delta^{WIP} \in [0.5; 1.0]
\end{aligned} \tag{B.5}$$

$$\begin{aligned}
\Delta^{U} &= \frac{U^{FF}}{U^{NF}} = \frac{\frac{q_{1A}+q_{1B}+q_{2A}+q_{2B}}{2(c+C^{FF})}}{\frac{q_{1A}+q_{1B}+q_{2A}+q_{2B}}{2(c+C^{NF})}} = \frac{c+C^{NF}}{c+C^{FF}} = \frac{c+q_{1A}+q_{1B}+\sigma+q_{2B}-c}{c+q_{2A}+q_{2B}+\sigma} = \frac{q_{1A}+q_{1B}+q_{2B}+\sigma}{c+q_{2A}+q_{2B}+\sigma} \\
&\overset{(ii)}{\leq} \frac{q_{2A}+q_{2B}+q_{2B}+\sigma}{c+q_{2A}+q_{2B}+\sigma} = \frac{q_{2A}+2q_{2B}+\sigma}{c+q_{2A}+q_{2B}+\sigma} \\
&\overset{(iii)}{\leq} \frac{q_{2A}+2q_{2B}}{q_{2A}+q_{2B}} \\
&\overset{(iv)}{=} \frac{q_{2A}+2\Psi q_{2A}}{q_{2A}+\Psi q_{2A}} = \frac{1+2\Psi}{1+\Psi} \Rightarrow \Delta^{U} \in [1.0; 2.0]
\end{aligned} \tag{B.6}$$

Then, (B.5) and (B.6) verify Theorem 1 and conclude the proof. □

B.2 NP-hardness proof

In the following, we proof that Problem 2 is NP-hard, by proving the NP-hardness of Problem 3, which entails the hardness result for Problem 2. To proof the NP-hardness of Problem 3, we use a transformation from the job shop scheduling problem (JSP), which is known to be NP-hard in the strong sense (Lenstra, Rinnooy Kan, & Brucker, 1977). The JSP in its feasibility version is defined as follows:

Job shop scheduling problem: We consider n jobs $J_1, J_2, \ldots, J_n$, each having a release date μ_j and a due date υ_j. Each job consists of a sequence of operations that require a certain processing time on a predefined machine. Every job may visit a machine at most once and no transportation times between machines occur. W.l.o.g., we consider due dates that equal the jobs' release dates plus a constant Φ. If a job's completion time χ_j is higher than its due date, job j is late, formally $\Lambda_j = \chi_j - \upsilon_j$. A schedule of the JSP assigns all jobs to machines such that only one

job is assigned to a machine at a time. In this setting, we ask whether a schedule satisfies $\Lambda^{max} = \max_j \Lambda_j$ for a given Λ^{max}.

We transform an instance I of the JSP as described above into an instance I' of Problem 3 in polynomial time as follows: for any job in the JSP, we create one vehicle in Problem 3. The release dates represent the vehicles' arrival times. For any machine in the JSP, we create a station in Problem 3. The vehicle's tasks correspond to the job's operations and have equal processing times. We set transportation times to zero and neglect operation and routing flexibility.

If I is feasible to the JSP with maximum lateness Λ^{max}, choosing the same schedule for the corresponding vehicles in Problem 3 as for the jobs in the JSP leads to a solution to I' with $C_s \leq \Phi + \Lambda^{max}$. Vice versa, any solution to I' of Problem 3 with $C_s \leq \Phi + \Lambda^{max}$ is also a feasible solution to I for the JSP with maximum lateness Λ^{max}, which completes the proof. □

B.3 Preselection problem

In the following, we formalize the *preselection problem* to predetermine task sequences and/or task-to-station assignments for the NF, OF, and RF configurations. Here, we select task sequences and task-to-station assignments such that the transportation distances are minimal and such that the workload is equally distributed among the stations.

Let $m \in \mathcal{M}$ be the set of models. The set $r \in \hat{\mathcal{R}}_m$ comprises all routes for model m. In contrast to the problem formulation in Section 3.3.2, the routes here only encode the sequence of visited stations and performed tasks but exclude the timing of the operations. The parameter d_{mr} indicates the transportation distance of route r for model m, and u_{mrl} shows the aggregated workload at station l for model m if route r is selected. We encode the average workload per station in $\hat{u}$ and use two types of decision variables. Binary variables X_{mr} indicate whether route r is selected for model m ($X_{mr} = 1$) or not ($X_{mr} = 0$). Continuous variables G_l represent the positive workload deviation from the average workload at station l.

Problem 5

$$\min \sum_{m \in \mathcal{M}} \sum_{r \in \hat{\mathcal{R}}_m} d_{mr} X_{mr} \tag{B.7a}$$

$$\min \sum_{l \in \mathcal{L}} G_l \tag{B.7b}$$

s.t.

$$\sum_{r \in \hat{\mathcal{R}}_m} X_{mr} = 1 \qquad \forall m \in \mathcal{M} \qquad \text{(B.7c)}$$

$$G_l \geq \sum_{m \in \mathcal{M}} \sum_{r \in \hat{\mathcal{R}}_m} u_{mrl} X_{mr} - \hat{u} \qquad \forall l \in \mathcal{L} \qquad \text{(B.7d)}$$

$$X_{mr} \in \{0, 1\} \qquad \forall m \in \mathcal{M}, r \in \hat{\mathcal{R}}_m \qquad \text{(B.7e)}$$

$$G_l \geq 0 \qquad \forall l \in \mathcal{L} \qquad \text{(B.7f)}$$

We formalize the preselection problem using a lexicographic objective. The Primary Objective (B.7a) minimizes the transportation distances of the selected routes, while the Secondary Objective (B.7b) equally distributes the workload across all stations. Therefore, we minimize the sum of positive workload deviations from the average workload at all stations. Constraints (B.7c) select one route for each model. In Constraints (B.7d), we derive the positive workload deviation from the average workload at all stations. Finally, Constraints (B.7e) - (B.7f) define the domains of our decision variables.

We solve the preselection problem to obtain the predetermined task sequences and task-to-station assignments. For the NF and OF configurations, we then fix the task-to-station assignments for all models, while we fix the task sequences of all models for the NF and RF configurations.

B.4 Mixed-model sequencing

In order to quantify the utilization and output levels in an LAL segment, we determine the optimal vehicle sequence for the LAL segment based on a mixed-model sequencing problem formulation.

Let $o \in \mathcal{O}$, $l \in \mathcal{L}^{line}$, and $m \in \mathcal{M}$ be the sets of sequence positions, stations, and models. The parameter ν_m denotes the occurrence of model m, while $\tilde{q}_{ml}$ is the processing time of model m at station l, and ϑ indicates the length of all stations on the line. We define three types of decision variables. W_{lo} quantifies the work overload induced by the o^{th} vehicle at station l, and S_{lo} represents the worker start position at station l for the o^{th} vehicle. Binary variables $\tilde{X}_{mo}$ indicate whether model m is produced at sequence position o ($\tilde{X}_{mo} = 1$) or not ($\tilde{X}_{mo} = 0$). The mixed-model sequencing problem can then be formulated as follows:

Problem 6

$$\min \sum_{l \in \mathcal{L}^{line}} \sum_{o \in \mathcal{O}} W_{lo} \tag{B.8a}$$

s.t.

$$\sum_{o \in \mathcal{O}} \tilde{X}_{mo} = \nu_m \qquad \forall m \in \mathcal{M} \tag{B.8b}$$

$$\sum_{m \in \mathcal{M}} \tilde{X}_{mo} = 1 \qquad \forall o \in \mathcal{O} \tag{B.8c}$$

$$S_{lo} \geq S_{l,o-1} + \sum_{m \in \mathcal{M}} \tilde{q}_{ml} \tilde{X}_{m,o-1} - c - W_{l,o-1} \qquad \forall l \in \mathcal{L}^{line}, o \in \mathcal{O} \tag{B.8d}$$

$$S_{lo} + \sum_{m \in \mathcal{M}} \tilde{q}_{ml} \tilde{X}_{mo} - W_{lo} \leq \vartheta \qquad \forall l \in \mathcal{L}^{line}, o \in \mathcal{O} \tag{B.8e}$$

$$S_{l0} = 0, S_{l,|\mathcal{O}|+1} = 0 \qquad \forall l \in \mathcal{L}^{line} \tag{B.8f}$$

$$S_{lo} \geq 0, W_{lo} \geq 0 \qquad \forall l \in \mathcal{L}^{line}, o \in \mathcal{O} \tag{B.8g}$$

$$\tilde{X}_{mo} \in \{0, 1\} \qquad \forall m \in \mathcal{M}, o \in \mathcal{O} \tag{B.8h}$$

Objective (B.8a) minimizes work overloads. Constraints (B.8b) ensure that all models are produced in the correct amount, and Constraints (B.8c) state that only a single model is assigned to a sequence position. We derive the worker start positions and work overloads in Constraints (B.8d) - (B.8e). Constraints (B.8f) set the workers to their initial position at the beginning and end of the sequence to be planned. The variable domains are defined in Constraints (B.8g) - (B.8h).

B.5 Identification of sampling parameters

In the following, we give evidence to our choice of sampling parameters. We first explain how we determined the required number of sample sequences, before we detail how we selected a sufficient sequence length.

Number of sample sequences: To determine the number of sample sequences, we analyze the dependency between the coefficient of variation of the segment cycle time and the number of sample sequences. Let C_ψ denote the segment cycle time considering ψ sample sequences. For any $|\mathcal{S}| \geq 10$, we compute the coefficient of variation in C_ψ among all $\psi \in [|S| - 9, |S|]$. Figure B.3 shows the boxplots across all instances and flexibility configurations for a feasibility target of 90%. As can be seen, the coefficient of variation stabilizes for sample sizes above 30. Accordingly, we choose $|\mathcal{S}| = 50$ as a sufficient number of sample sequences.

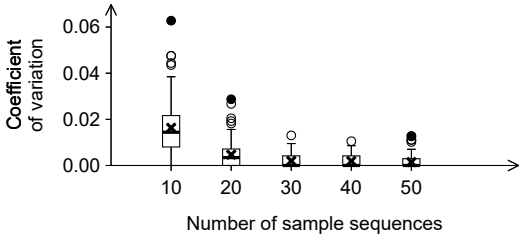


Figure B.3: Boxplot on the coefficient of variation in segment cycle time for different number of sample sequences.

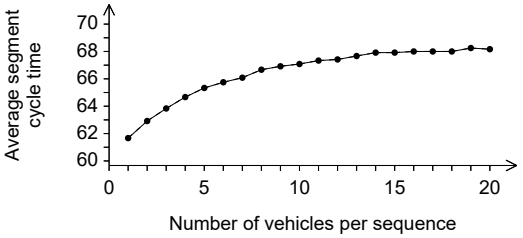


Figure B.4: Average segment cycle time in the FF configuration for different numbers of vehicles in the sequences.

Sequence length: Figure B.4 shows the average segment cycle time across all instances in the FF configuration for different sequence lengths. We note that the average segment cycle time increases degressively with longer sequences. As can be seen, the relative changes in the average segment cycle time remain marginal for sequence lengths of more than 15 vehicles. Accordingly, we choose a sequence length of 20 vehicles.

B.6 Mixed-model assembly line balancing

In our computational analyses, we compare FALs to conventional LALs. To construct LALs for comparison, we use a mixed-model assembly line balancing problem with the objective to minimize the number of stations.

Let $i \in \mathcal{I}$, $l \in \mathcal{L}^{line}$, and $m \in \mathcal{M}$ be the sets of tasks, stations, and models. The set $\mathcal{F}_i$ indicates the successor tasks of task i across all models. The demand-weighted average processing time of task i is $\bar{q}_i$. We define two types of decision variables. Binary variables V_{il} indicate if task i is assigned to station l ($V_{il} = 1$) or not ($V_{il} = 0$). The continuous variable N represents the index of the last opened station on the line. Then, the mixed-model assembly line balancing problem can be formulated as follows:

Problem 7

$$\min \; N \tag{B.9a}$$

s.t.

$$\sum_{l \in \mathcal{L}^{line}} V_{il} = 1 \qquad \forall i \in \mathcal{I} \tag{B.9b}$$

$$\sum_{l \in \mathcal{L}^{line}} l V_{i_1 l} \leq \sum_{l \in \mathcal{L}^{line}} l V_{i_2 l} \qquad \forall i_1 \in \mathcal{I}, i_2 \in \mathcal{F}_{i_1} \tag{B.9c}$$

$$\sum_{i \in \mathcal{I}} \bar{q}_i V_{il} \leq c \qquad \forall l \in \mathcal{L}^{line} \tag{B.9d}$$

$$N \geq l V_{il} \qquad \forall i \in \mathcal{I}, l \in \mathcal{L}^{line} \tag{B.9e}$$

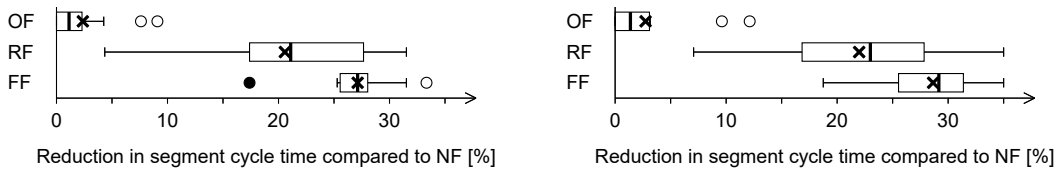
$$V_{il} \in \{0, 1\} \qquad \forall i \in \mathcal{I}, l \in \mathcal{L}^{line} \tag{B.9f}$$

Objective (B.9a) minimizes the number of opened stations. Constraints (B.9b) assign every task to exactly one station, and Constraints (B.9c) enforce precedence relations. Constraints (B.9d) ensure that the average workload at a station is below the cycle time c. We derive the objective value in Constraints (B.9e). Constraints (B.9f) state the binary variable domains.

B.7 Discussion on the number of task duplicates

In this section, we justify our choice on the number of task duplicates in FAL segments. We allow every task to be duplicated once such that it can be assigned to two stations at most. We study the effect of changing the number of task duplicates on the average segment cycle time for all flexibility configurations. Here, we found that the average segment cycle time is insensitive to the number of task duplicates for the NF and OF configurations, because the benefits of task duplicates cannot be exploited without routing flexibility. For the RF and FF configurations, we observe that changing the number of task duplicates from one to zero heavily increases the average segment cycle time by 28% and 38% respectively. Increasing the number of task duplicates from one to two, in contrast, yields insignificant changes. Accordingly, we fix the number of task duplicates to one such that each task exists at most twice.

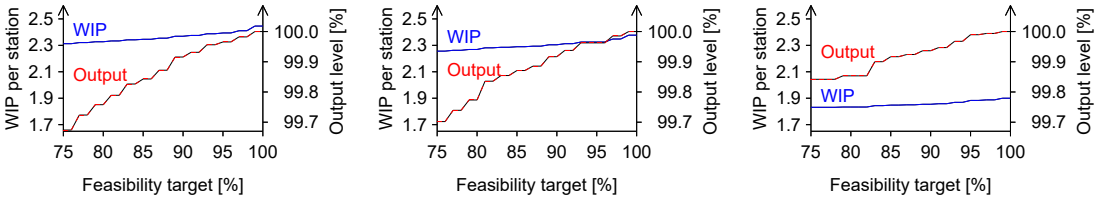
B.8 Effect of feasibility target on segment cycle time



(a) Feasibility target of 75%. **(b)** Feasibility target of 100%.

Figure B.5: Reduction in the segment cycle time (WIP) due to flexibility for different feasibility targets.

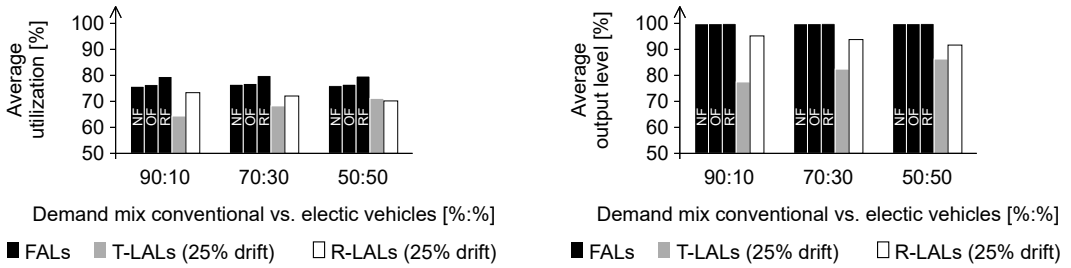
B.9 Effect of feasibility target on average WIP and output level



(a) NF configuration. **(b)** OF configuration. **(c)** RF configuration.

Figure B.6: Impact of feasibility target on average WIP and output level for NF, OF, and RF configurations.

B.10 Ramp-up result figures



(a) Utilization. **(b)** Output level.

Figure B.7: Performance of FALs (NF, OF, and RF configurations), T-LALs, and R-LALs during ramp-up.

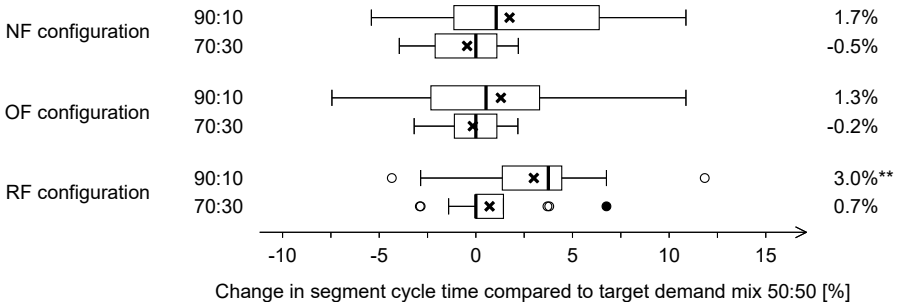


Figure B.8: Adjustments of the segment cycle time in FALs (NF, OF, and RF configurations) during ramp-up.