DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

REBALANCING MANUFACTURING LINES BY META-HEURISTIC METHODS

by
Tolga ÇİMEN

February, 2023
İZMİR

REBALANCING MANUFACTURING LINES BY META-HEURISTIC METHODS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Industrial Engineering Program

by
Tolga ÇİMEN

February, 2023
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"REBALANCING MANUFACTURING LINES BY META-HEURISTIC METHODS"** completed by **TOLGA ÇİMEN** under the supervision of **PROF.DR. ADİL BAYKASOĞLU** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

…...…….

Prof. Dr. Adil BAYKASOĞLU

Supervisor

…...…….

Assoc. Prof. Dr. Ali Serdar TAŞAN

Thesis Committee Member

…...…….

Assoc. Prof. Dr. Kökten Ulaş BİRANT

Thesis Committee Member

…...…….

Assoc. Prof. Dr. Zeynel Abidin ÇİL

Examining Committee Member

…...…….

Assoc. Prof. Dr. İlker GÖLCÜK

Examining Committee Member

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude to my supervisor Prof. Dr. Adil BAYKASOĞLU, for believing in me and making it possible to participate in the Ph.D. program in industrial engineering as a computer engineer. This thesis would not exist if you did not give me a chance for an application. I would also like to thank my thesis committee members, Assist. Prof. Dr. Kökten Ulaş BİRANT and Assoc. Prof. Dr. A. Serdar TAŞAN, for participation in my Ph.D. journey. It was always a pleasure meeting with you each term in the thesis meetings.

Additionally, I am grateful to Şebnem DEMİRKOL AKYOL for her invaluable support and cooperation in developing the study. You were a great coworker, and I always enjoyed working together. Also, thanks to Nazan GÜNEY for always being very supportive and kind, which made me feel peaceful every time I visited you, even in difficult times. I could not imagine this department without you.

Last but most, I would like to thank my wife, Özlem, and my girls for their continuous support, understanding, patience, and love.

Tolga ÇİMEN

REBALANCING MANUFACTURING LINES BY META-HEURISTIC METHODS

ABSTRACT

A wide variety of approaches has been proposed since the 50s to solve the assembly line balancing problem. Almost all of them established an assembly line configuration from the beginning. However, in real life, a pre-balanced assembly line may have to be rebalanced very often for various reasons, such as changes in the cycle time or tasks' operation times, demand fluctuations, changes in product features or production technologies, etc. Therefore, it has attracted scientists' interest increasingly in recent years. Heuristic and meta-heuristic optimization methods are widely used to solve assembly line rebalancing problems (ALRBP) because they can be effective in finding good solutions in a reasonable amount of time, even for complex problems with a high number of variables, constraints, or multiple objectives. Although there are several research attempts on ALRBP in the relevant literature, only a few studies consider workers as unique individuals. This thesis aims to solve the double reassignment problem: tasks to workers and workers to stations with considering ergonomic risk factors. A randomized constructive rule-based heuristic approach is developed with the objectives of minimizing rebalancing costs, which include transportation of tasks and workers and minimizing ergonomic risk factors of stations. The rebalancing performance of the proposed method is evaluated with benchmark data, and obtained results are promising. Then, the proposed solution approach is implemented in a real-life case study in an automotive parts manufacturing plant. Moreover, a comprehensive review of the literature on ALRBP is presented.

Keywords: Assembly line rebalancing problem, rule-based randomized constructive heuristic, ergonomic risk assessment, OCRA, worker assignment

META-SEZGİSEL YÖNTEMLER İLE ÜRETİM HATLARININ YENİDEN DENGELENMESİ

ÖZ

Montaj hattı dengeleme problemini çözmek için 50'li yıllardan beri çok çeşitli yaklaşımlar önerildi. Bu yöntemlerin neredeyse tamamında en başından bir montaj hattı konfigürasyonu oluşturulması üzerinde çalışıldı. Ancak gerçek hayatta sıklıkla çevrim süresi veya işlerin sürelerindeki değişiklikler, sipariş dalgalanmaları, ürün özelliklerindeki veya üretim teknolojilerindeki değişiklikler vb. çeşitli nedenlerle önceden dengelenmiş bir montaj hattının yeniden dengelenmesi gerekebilir. Bu nedenle, montaj hattı yeniden dengeleme problem son yıllarda araştırmacıların giderek daha fazla ilgisini çekmektedir. Sezgisel ve meta-sezgisel optimizasyon yöntemleri, çok sayıda değişken ve kısıt içeren ya da çok amaçlı olan karmaşık problemler için bile makul bir sürede iyi çözümler bulmada etkilidirler. Bu nedenle montaj hattı yeniden dengeleme problemlerini çözmek için de yaygın olarak kullanılmaktadırlar. Literatürde montaj hattı yeniden dengeleme problemi ile ilgili birçok yayın olmasına rağmen, yalnızca birkaç çalışmada, işçiler, farklı yetkinlik ve kapasitede benzersiz bireyler olarak kabul edilmiştir. Bu tezin amacı, görevlerin işçilere ve işçilerin istasyonlara atanmasını içeren ikili yeniden atama probleminin, ergonomik risk faktörlerini göz önünde bulundurarak çözülmesidir. Problemin çözümü için, kural tabanlı rassal uyarlamalı çözüm kurucu sezgisel bir yöntem geliştirilmiştir. Bu yöntem ile, görevlerin ve çalışanların taşıma maliyetlerinin sebep olduğu yeniden dengeleme maliyetlerini ve montaj istasyonlarındaki ergonomik risk faktörlerini en aza indirmek amaçlanmıştır. Önerilen yöntemin yeniden dengeleme performansı literatürdeki bilinen test problemleri ile değerlendirilmiş ve başarılı sonuçlar elde edilmiştir. Sonrasında, önerilen çözüm yaklaşımı, otomotiv kablo ağı üretimi yapan bir fabrikada gerçek hayattan bir vaka çalışmasında uygulanmıştır. Ayrıca, montaj hattı yeniden dengeleme problemi ile ilgili literatürün kapsamlı bir incelemesi sunulmuştur.

Anahtar kelimeler: Montaj hattı yeniden dengeleme problemi, kural tabanlı rassal uyarlamalı çözüm kurucu sezgisel, ergonomik risk değerlendirme, OCRA, işçi atama

CONTENTS

Page

Ph.D. THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZ v
LIST OF FIGURES ix
LIST OF TABLES xi
LIST OF SYMBOLS xii

CHAPTER 1 - INTRODUCTION 1

CHAPTER 2 - LITERATURE REVIEW ON ASSEMBLY LINE REBALANCING PROBLEMS..... 5

2.1 Material Collection Method 5
2.2 Descriptive Analysis 8
2.3Assembly Line Rebalancing Problem Classification 10
2.3.1 The Type of Product Mix in The Assembly Line 13
2.3.2 Line Layout 14
2.3.3 Variability of Task Times 15
2.3.4 Additional Problems That Are Incorporated in The ALRBP 15
2.4 Objective Functions 17
2.5 Problem Constraints 22
2.6 Causes of Rebalancing 26
2.7 Solution Methods …..... 30
2.7.1 Genetic Algorithms (GA) 31
2.7.2 Simulated Annealing (SA) 32
2.7.3 Differential Evolution (DE) 33
2.7.4 Tabu Search and Local Search 33
2.7.5 Multi-Objective Optimization (MOO) 34
2.7.6 Particle Swarm Optimization (PSO) 35

2.7.7 Ant Colony Optimization (ACO) 36
2.7.8 Simulation-Based Approaches..... 36
2.8 Test Data Types for Solution's Performance Evaluation 38
2.9 Overview of ALRBP Papers 39

CHAPTER 3 - ASSEMBLY LINE REBALANCING AND WORKER ASSIGNMENT PROBLEM CONSIDERING ERGONOMIC RISK FACTORS (ERGOALWARBP) 54

3.1 Assumptions 54
3.2 Objective Functions 55
3.3 Constraints 58

CHAPTER 4 - THE PROPOSED ALGORITHM FOR ERGOALWARBP 60

4.1 The Rule-Based Task Selection Algorithm 60
4.2 The Rule-Based Worker Selection Algorithm 61
4.3 The Developed Preemptive Goal Programming Algorithm for Fitness Evaluation 62

CHAPTER 5 – COMPUTATIONAL STUDY 66

5.1 Computational Study Results……..... 66

CHAPTER 6 – INDUSTRIAL CASE STUDY 69

6.1 Operational Problem Information……..... 69
6.2 Analysis of the Current Assembly Line 72
6.3 ErgoALWARBP Decision Support Software: Inputs, Outputs, and Solution Environment 74
6.4 Industrial Case Study Implementation, Results, and Discussion 75

CHAPTER 7 – CONCLUSION 82

REFERENCES 84

APPENDICES 95

LIST OF FIGURES

Page

Figure 2.1 Distribution of the publications per year (33 articles, eight proceedings) 8
Figure 2.2 Distribution of the publications based on their country of origin 9
Figure 2.3 Distribution of publications based on journals (34 articles: 2005-2022) 9
Figure 2.4 Assembly line rebalancing problems classification 13
Figure 2.5 Distribution of publications based on the type of product mix in the assembly line 16
Figure 2.6 Distribution of publications based on the type of line layout 16
Figure 2.7 Distribution of publications based on the variability of task times 16
Figure 2.8 Distribution of publications based on solved problems 16
Figure 2.9 Distribution of publications based on objective functions 18
Figure 2.10 Distribution of publications based on the number of objective functions 18
Figure 2.11 Distribution of publications based on problem constraint 23
Figure 2.12 Distribution of publications based on causes of rebalancing 27
Figure 2.13 Distribution of publications based on solution methods 38
Figure 2.14 Distribution of publications based on test data types for solution evaluation 39
Figure 4.1 Pseudo code of the rule-based task selection algorithm 61
Figure 4.2 Pseudo code of the rule-based worker selection algorithm 62
Figure 4.3 Pseudo code of the preemptive goal programming algorithm (for Goal 1) 64
Figure 4.4 Pseudo code of Calculating the best solution for all goals (changing order of objective functions) 65
Figure 6.1 Flowchart of the proposed heuristic solution procedure 71
Figure 6.2 Precedence relations of the assembly line 73
Figure 6.3 Precedence diagram of task 1 73
Figure 6.4 Line efficiency & line smoothness index values of the feasible solutions 76

Figure 6.5 The distribution of the feasible solutions in terms of total rebalancing cost 79

LIST OF TABLES

Page

Table 2.1 Search results from databases for selected keywords 7
Table 2.2 Web of Science (WoS) and EBSCO advanced search strings with the returned number of results 7
Table 2.3 Assembly line rebalancing problem classification of the referenced works 11
Table 2.4 Objective functions of the referenced works 19
Table 2.5 Problem-specific constraints of the referenced works 24
Table 2.6. Causes for the rebalancing of the referenced works 28
Table 2.7. Solution methods and test data types for solution's performance evaluation of the referenced studies 37
Table 5.1 Benchmark test problems and the results of the computational study 68
Table 6.1 Task transportation costs and task performing times of the workers in the industrial case study 73
Table 6.2 Initial assignment’s and the proposed solutions’ station-task and station-worker assignments of the industrial case-study 77
Table 6.3 Computational results of the industrial case study 79

LIST OF SYMBOLS

The notation used in the rest of the paper is as follows (Celik et al., 2014; Gamberini et al., 2006; Miralles et al., 2008; Yang et al., 2013)

i, j	: task index, for $i, j = 1, \dots, n$
a, b	: station index, for $a, b \in S$
TC_{rb}	: the total cost of rebalancing
$cost_{open}$	: opening costs of a new station
$cost_{close}$	: cost of closing an existing station
$cost_{run}$	: operating cost of a station for a month
$cost_i$	: transportation cost of task *i*
N_{rb}	: set of tasks that were transported for rebalancing
m	: number of stations
m^0	: initial station count
m^+	: number of new stations that were opened after rebalancing
m^-	: number of stations that were closed after rebalancing
MSF	: mean similarity factor of the line
SF_i	: similarity factor of task *i*, for $i \epsilon N$
TIB_i	: set of tasks allocated to the same station as task *i* in the initial balancing, other than task *i*, for $i \epsilon N$
TNB_i	: set of tasks allocated to the same station as task *i* in the new balancing, other than task *i*, for $i \epsilon N$
n	: number of tasks
n^0	: number of tasks in the initial balancing
$WorkerMSF$	: mean similarity factor of the workers
$WorkerSF_w$	: similarity factor of worker *w*, for $w \in W$
w	: worker index, for $w \in W$
W	: set of workers
N_w	: set of tasks assigned to worker *w*, for $w \in W$
N_w^0	: initial set of tasks assigned to worker *w*, for $w \in W$
$madOCRA$	: mean absolute deviation of OCRA index values

$avgOCRA$: OCRA average

$OCRA_a$: OCRA index value of station a

d_i^- : negative deviation

d_i^+ : positive deviation

n_{rb} : number of tasks that were transported for rebalancing

NS_a^0 : initial set of tasks assigned to station a, for $a = 1, \dots, m^0$

NS_a : set of tasks assigned to station a, for $a = 1, \dots, m$

LE : line efficiency

SX : similarity index

c : cycle time of the line

Tt_a : total task processing time of station a (station time)

S : set of stations

N : set of tasks

y_{aw} : binary variable equal to 1, only if worker w is assigned to station a

x_{iaw} : binary variable equal to 1, only if task i and worker w are assigned to station a

t_{iw} : processing time of task i when worker w executes it

IP_j : immediate predecessors of task j

CHAPTER 1
INTRODUCTION

An assembly line (AL) is a manufacturing process in which a product is built by moving it through a series of stations, each of which performs a specific task. The product moves from station to station on a conveyor belt or other transportation system, and workers at each station add or assemble components or perform other tasks to complete the product. The goal of an assembly line is to increase efficiency and productivity by breaking down the manufacturing process into smaller, more specialized tasks that can be completed quickly and accurately. The assembly line is a key component of mass production, as it allows companies to produce large quantities of products quickly and efficiently. It was first implemented by Henry Ford in the early 20th century and has since become a common method of manufacturing in many industries.

A critical aspect of designing and managing an assembly line is the assembly line balancing problem (ALBP), which problem is a classic problem in operations research and industrial engineering. ALBP involves determining the optimal number and arrangement of workstations and tasks to minimize the overall production time and maximize efficiency. This involves considering factors such as the time required to complete each task, the dependencies between tasks, and the availability of resources and materials. A balanced AL is important because an imbalanced AL can lead to bottlenecks, delays, and increased costs.

Salveson wrote the first paper on ALBP in 1955 (Salveson, 1955), and the subject received growing interest from researchers over time. For over 65 years, various algorithms and methods have been suggested for solving ALBP. In his pioneer study, Scholl (1999) classified ALBPs according to fundamental characteristics of ALs, such as product mix, line layout, and objective. Based on the objective function, two main forms of ALBP are Type 1 and Type 2. Type 1 tries to minimize the number of workstations while maintaining a constant cycle time. Type 2 tries to minimize the cycle time with a fixed amount of workstations (Karp, 1972). Moreover, line

productivity is tried to be increased by minimizing the number of workstations and the cycle time concurrently on Type E problems.

Since the fundamental goal of early ALs was primarily based on producing a particular product in large quantities, the first research attempts on ALBP were focused on single-model, straight-line ALs with deterministic task operation times. Over the years, the variety of customer demands has been enhanced, and consequently, ALs have shifted to mixed-model and multi-model lines to produce customized products (Boysen et al., 2007). Although single-objective problems such as Type 1 and Type 2 problems were solved initially, multi-objective ALBPs were considered by adding additional objectives in the following years. Moreover, additional problem-specific constraints have been considered recently, such as different types of line layouts, stochastic task operation times, worker assignment, and ergonomic risk factors assessments. As the problem's complexity increases, solution approaches have evolved to approximate methods such as heuristics and meta-heuristics instead of exact solution techniques (Battaïa & Dolgui, 2013). Lately, the ALBP has orientated toward flexible assembly paths due to technological developments such as digitization, autonomous robots, and Internet-of-Things technology (Boysen et al., 2022).

The use of assembly lines in manufacturing has been a prevalent practice for many years to increase efficiency and productivity. In ALBP, a completely new assembly line configuration is designed from the beginning. However, the dynamic nature of the production process often leads to imbalances in the flow of work, resulting in bottlenecks and inefficiencies. Traditional methods for addressing these imbalances, such as manual adjustments or fixed schedules, are limited in their ability to adapt to changing demands and conditions. As a result, the assembly line rebalancing problem (ALRBP) occurs. ALRBPs are a special case of traditional ALB problems. The tasks are already assigned to the workstation, and the line must be rebalanced due to industrial conditions. The most usual reasons can be listed as; changes in cycle time due to demand fluctuations (Ağpak, 2010; Belassiria et al., 2018; Celik et al., 2014; Corominas et al., 2008; Mokhtari & Mozdgir, 2015; Serin et al., 2019), changes in product design and features (Altemeier et al., 2010; Karas & Ozcelik, 2021; Makssoud

et al., 2013; Yang & Gao, 2016), changes in task times (Y. Li & Boucher, 2017; Sikora et al., 2017; Zha & Yu, 2014), changes in task precedence relations (Gamberini et al., 2006; Y. Zhang et al., 2018a), and machine breakdowns (Ishigaki, 2018; Sancı & Azizoğlu, 2017). The reasons that cause rebalancing will be explained in the following sections. In recent years, the number of studies on rebalancing ALs have increased. As the manufacturing environment has become more dynamic and the requirement for customized products has increased, the rebalancing problem has become a hot topic for researchers. Many researchers are interested in the ALRBPs, and currently, there are several solution approaches to deal with them.

Many of the current AL rebalancing literature studies aim to rebalance the AL to reduce the number of changes to be made on the initial line. Moreover, worker assignment is barely studied in AL rebalancing literature (Çimen et al., 2022b). However, the assumption that all workers are identical does not represent a real-life case. The assembly line rebalancing problem, when combined with worker assignment, involves adjusting the workstations and tasks in an assembly line and determining the optimal assignment of workers to these tasks to improve efficiency and productivity. This can be a complex problem, as it involves considering both the technical requirements of the tasks and the skills and capabilities of the workers.

Moreover, the mentioned rebalancing and worker assignment problems generally occur in manual ALs. Workers on manual ALs can be at risk of developing occupational diseases due to the repetitive and potentially hazardous tasks they perform. Workers who perform the same task repeatedly, such as assembling components or assembling products, may be at risk of developing injuries such as tendonitis, carpal tunnel syndrome, and other musculoskeletal disorders. In addition, if the workstations and equipment are not designed or set up in a way that is ergonomically correct, workers may be at risk of developing injuries due to poor posture or awkward body positions. It takes a long time to treat certain occupational diseases, and this causes a loss of productivity and workforce. Musculoskeletal conditions account for a significant portion of occupational illnesses in many European countries. In Turkey, approximately 40 percent of all occupational diseases are

expected to be associated with musculoskeletal disorders (Berk et al., 2011). It is vital for employers to implement measures to minimize the risk of occupational diseases on manual assembly lines, such as providing appropriate training and equipment, enforcing safety regulations, and ensuring that workstations are designed to be ergonomically correct. Workers should also be aware of the potential risks and take steps to protect their own health and safety. This would eliminate occupational diseases and increase productivity. Although the influence of occupational exposure and ergonomics in production are becoming more and more important in practice, to our best knowledge, there is only one research that considers ergonomic aspects in the AL rebalancing literature (Çimen et al., 2022a). This study presents an assembly line rebalancing and worker assignment problem by considering ergonomic risk factors to narrow this distance between the literature and real life (ErgoALWARBP). The presented ErgoALWARBP is interesting and novel since none of the previous work in the relevant literature handled the rebalancing of ALs under ergonomic and worker assignment constraints.

The remainder of this study is organized as follows. In Section 2, a bibliometric analysis of the ALRBP papers is provided, the structure for analyzing the reviewed publications is defined, the literature is reviewed according to the previously specified focuses, and the justification of this research is stated. In Section 3, the novel ErgoALWARBP is detailly defined in mathematical formulations and various additional constraints. In Section 4, the randomized constructive rule-based heuristic approach is introduced. In section 5, the proposed heuristic is tested using benchmark test problems, and the results of the experiments are discussed, followed by the application of the developed approach on a real-life case in Section 6, and section 7 concludes the article.

CHAPTER 2
LITERATURE REVIEW ON ASSEMBLY LINE REBALANCING PROBLEMS

2.1 Material Collection Method

Our literature review covers conference proceedings, peer-reviewed review papers, and journal articles in English, indexed by the selected databases. We collected papers on ALRBPs through a detailed search from online databases: ISI-Web of Science, Scopus, ScienceDirect, Taylor & Francis Online, Wiley Online Library, EBSCO, and Google Scholar, using various keywords. These keywords were assembly line rebalancing (ALRB), "assembly line rebalancing" ("ALRB"), assembly line re-balancing (ALR-B), and "assembly line re-balancing" ("ALR-B"), and different versions of these keywords to improve the search performance. The search string was applied to all fields for each database. The search results for these keywords are shared in Table 2.1. The first column of Table 2.1 indicates the database such as EBSCO, Web of Science, Scopus, etc. The search string is given in the second column. For example, all fields in the EBSCO database were searched, including full text, abstract, keywords, topic, and title. The remaining columns depict the number of results obtained by searching ALRB, "ALRB", ALR-B, and "ALR-B". For example, the number of results obtained by searching the phrases assembly line rebalancing, "assembly line rebalancing", assembly line re-balancing, and "assembly line re-balancing" in the EBSCO database are 7763, 62, 855, and 54, respectively.

If there are title, keyword, and abstract search options, we run additional queries to improve the result's performance, as shown in Table 2.2.

Table 2.2 states that a deep search in the Web of Science (WoS) database was done to obtain results that are articles, proceedings papers, or review papers that are indexed in SCI-Expanded and Conference proceedings citation index Science (CPCI-S).

- Thirty-six results were obtained by searching assembly line rebalancing in all fields (full text, title, abstract, topic, and keyword).
- Thirty-five results were gathered by searching any of rebalance/rebalancing/re-balance/re-balancing (by using regular expression re$balanc*) and (assembly line) in the topic.
- Twenty-seven results were found by searching any of rebalance/rebalancing/re-balance/re-balancing and "assembly line" (as an exact phrase) in the topic, excluding the "balancing" keyword in the topic.
- Thirteen results were obtained by searching the third search string in the topic but excluding the "balancing" keyword in the title.

Search strings related to language, document type, and index choices were kept only in the first query for information and removed from others because of space constraints. Merging the four queries resulted in 38 papers after removing the duplicates. After weeding out the papers in the manual screening, 30 ALRBP papers were left from the WoS result set. Similarly, in the EBSCO database:

- Thirty-five results were gathered by searching any of rebalance/rebalancing/re-balance/re-balancing and "assembly line" (as an exact phrase) in all fields.
- Also, 16 results were obtained by searching the same string in the title.
- 439 results were found by searching the same string in the full text.
- Finally, 322 results were obtained by searching the same string in the full text, excluding the "balancing" keyword in the title.

Since only two of the databases have appropriate features for this detailed study, we only shared these two databases, Web of Science and EBSCO, in Table 2.2. In our search from the ScienceDirect database, *the search query* was applied in all fields, which resulted in 592 papers. The modified string *"assembly line rebalancing"* returned 23 papers, included in the first result set. After weeding out the papers, 32 ALRBP papers were left from the ScienceDirect result set.

Table 2.1 Search results from databases for selected keywords

Database	Keyword Search String	ALRB	"ALRB"	ALR-B	"ALR-B"	Total
EBSCO	All fields	7763	62	855	54	8734
Web of Science	All fields	32	15	19	9	75
Scopus	Article title, abstract, keywords	40	21	20	14	95
Science Direct	Find articles with these terms	592	23	50435	17	51067
Taylor & Francis Online	Anywhere	670	11	64404	9	65094
Wiley Online Library	Anywhere	581	0	86411	0	86992
Google Scholar	Anywhere	25600	199	4150	260	30209
Total		35278	331	206294	363	242266

Table 2.2 Web of Science (WoS) and EBSCO advanced search strings with the returned number of results

	Keyword Search String	Results
WoS	ALL= (assembly line rebalancing) AND LANGUAGE: (English) AND DOCUMENT TYPES: (Article OR Proceedings Paper OR Review) Indexes=SCI-EXPANDED, CPCI-S Timespan=All years	36
	TS= (assembly line) AND TS=(re$balanc*)	35
	(TS= ("assembly line" AND ("rebalance" OR "re-balance" OR "rebalancing" OR "re-balancing" NOT balancing)))	27
	(TS= ("assembly line" AND ("rebalance" OR "re-balance" OR "rebalancing" OR "re-balancing")) NOT TI=(balancing))	13
EBSCO	((rebalancing OR re-balancing OR rebalance OR re-balance) AND "assembly line")	35
	TI ((rebalancing OR rebalancing OR rebalance OR rebalance) AND "assembly line")	16
	TX ((rebalancing OR re-balancing OR rebalance OR re-balance) AND "assembly line")	439
	TX ((rebalancing OR rebalancing OR rebalance OR re-balance) AND "assembly line") NOT TI balancing	322

The Google Scholar search engine makes it possible to search for keywords anywhere in the title or all articles. As dash char (-) is used to exclude keywords and wildcards are not supported, it was harder to find articles containing "*re-balancing"* and "*re-balance"* keywords as it returns a larger dataset compared to WoS and Scopus. We also used the title search function to find out more accurate results. The modified strings *"assembly line rebalancing"* and *allintitle: "assembly line rebalancing"* returned 199 and 22 results.

Some studies deal with the rebalancing of a current line (Dar-El & Rubinovitz, 1991; Gel et al., 2002; Sculli, 1979, 1984; Van Oyen et al., 2001) without tackling the ALRBP by considering task reassignment as an objective function to minimization (Gamberini et al., 2006). Thus, we only include the publications in our review which considered reassignment in the objective function and compared the solutions with a pre-balanced assembly line. We merged the results from EBSCO, ScienceDirect, WoS, Scopus, Taylor & Francis Online, Wiley Online Library, and Google Scholar. After manually screening results and removing duplicate and unrelated works, 41 papers (peer-reviewed papers and conference proceedings) were left in our final database.

2.2 Descriptive Analysis

Figure 2.1 shows the annual number of articles reported in the research literature on ALRBP. The first publication occurred in 2005 (Yılmaz & Erol, 2005). Although there is a slight increase between 2010 and 2019 in terms of publication number, it has never passed five publications per year, and it is still deficient compared to publications on ALBP.

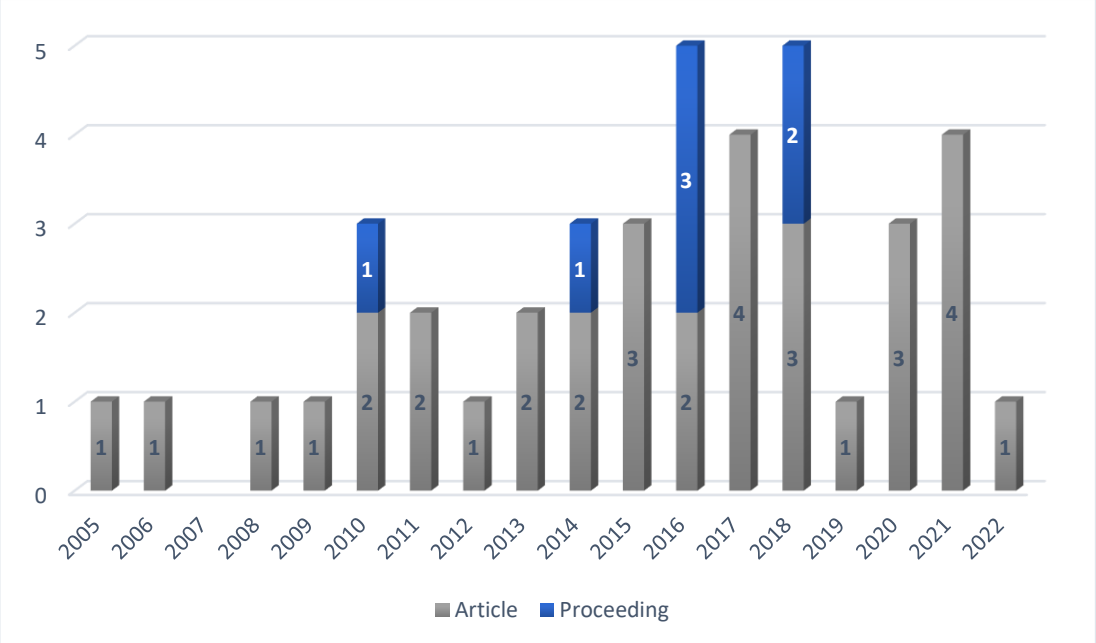


Figure 2.1 Distribution of the publications per year (33 articles, eight proceedings)

In Figure 2.2, the country of origin of these publications is presented. 26,9% of the works originated from China, followed by Turkey and France with 25,0% and 9,6%. The remaining 38,5% of the papers are from the other 13 countries. While 33 of these papers were published in peer-reviewed journals, eight were conference proceedings.

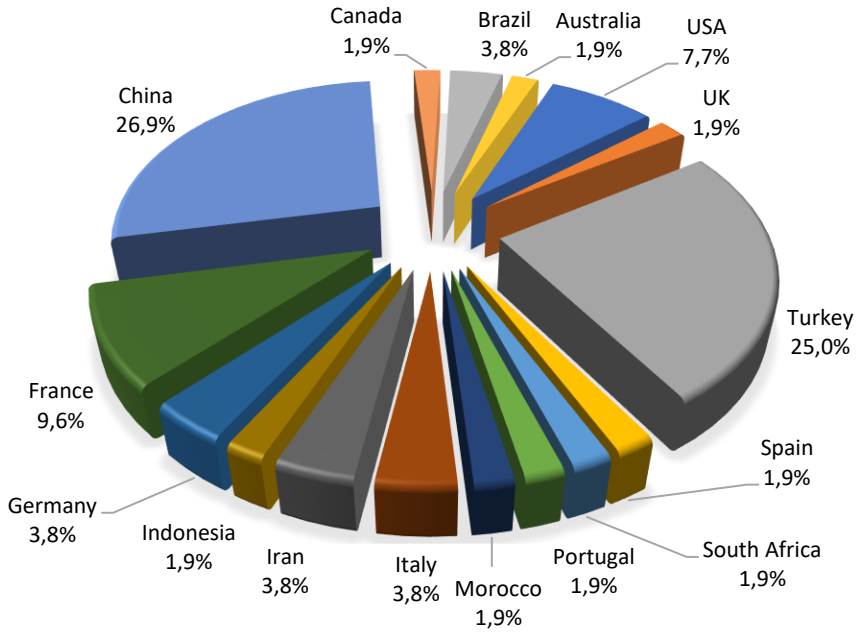


Figure 2.2 Distribution of the publications based on their country of origin

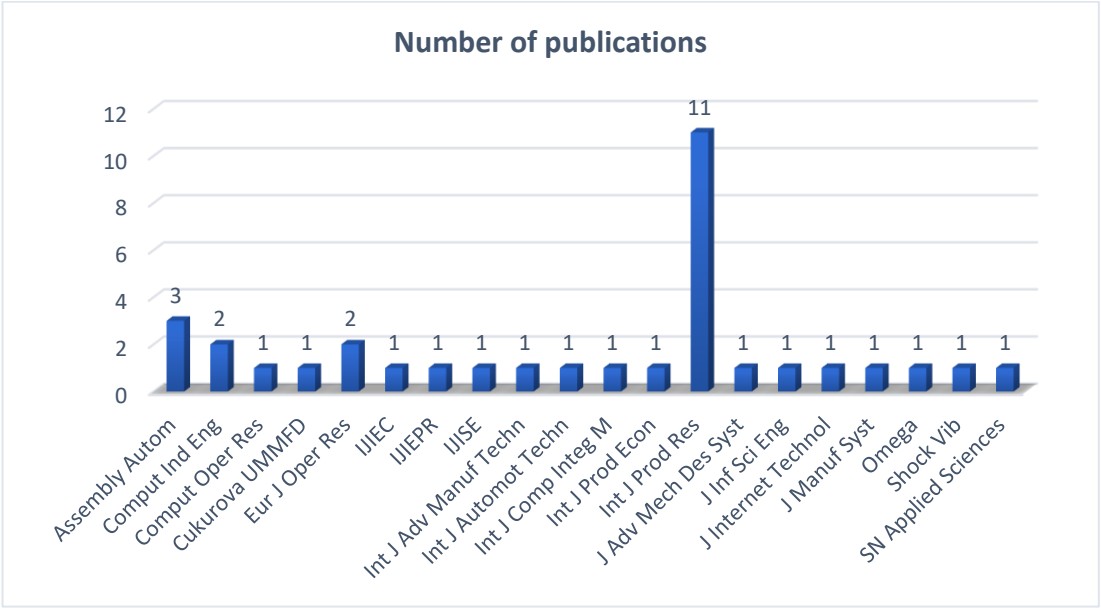


Figure 2.3 Distribution of publications based on journals (33 articles: 2005-2022)

The distribution of the 33 articles based on the journals is presented in Figure 2.3. 33,3% of the articles were published in the International Journal of Production Research (IJPR), followed by Assembly Automation, the European Journal of Operational Research, and Computers and Industrial Engineering. Furthermore, the eight proceedings were published at eight conferences between 2010 and 2018.

It can be concluded from the bibliometric analysis that although ALBP has been studied abundantly for many decades, ALRBP is a relatively new topic for researchers. There is a growing trend in rebalancing studies, which helps close the gap between the current literature and industrial assembly systems. Moreover, it is seen that a considerable part of the ALRBP studies is published in the IJPR.

2.3 Assembly Line Rebalancing Problem Classification

We analyzed and classified the ALRBP papers under the four topics below. ALRBP classification details for each referenced work are presented in Table 2.3. The first column of Table 2.3 gives the related reference. The second column depicts the problem type according to the product mix, such as single-model, mixed-model, and multi-model. Types of ALRBPs are similar to the types of ALBPs. In single-model ALs, one homogenous product is produced continuously in the line. However, customer-oriented production strategies necessitate product groups with different characteristics to be assembled. Consequently, ALs have evolved into the mixed model. In mixed-model ALs, different models of a standard product with similar features are produced simultaneously. Finally, in multi-model lines, similar products, namely several versions of a basic product, are assembled in batches on the same line incorporating set-up times.

The third column of Table 2.3 represents the type of line layout. In straight ALs, workstations are lined up consecutively. When a worker finishes executing a task in the workstation, they pass the task to the next workstation in the line. In U-shaped ALs (U-lines), the layout of the line is like the letter “U”, and the entry and exit points of the line are located at both ends of the “U”. The most important advantage of the U-

lines is that workers can execute adjacent tasks, as well as tasks on both sides of the U-line. In two-sided ALs, different tasks are operated simultaneously on the left and right sides of the AL. These lines are appropriate for producing high-volume large-sized standardized products.

ALRBPs could be divided into three categories according to the variability of task times. Deterministic task times indicate that the processing time of a task is assumed to be a fixed number. However, especially in manual ALs in real life, the operation time of a particular task could vary due to the worker. Many factors might affect the task time, such as the worker's fatigue level, period of the day, day of the week, etc. As a remedy, stochastic task times are introduced to the literature. Stochastic task times use a distribution function for the task operation time instead of a fixed number. Moreover, the operation time of a task might have changed during the rebalancing by using real-time information. Thus, dynamic task times occur (Scholl, 1999). The task times are categorized in the fourth column of Table 2.3.

Finally, the last column of Table 2.3 states if there exists an additional problem accompanying the rebalancing problem. For instance, some papers study worker reassignment problems together with the rebalancing problem (Budak & Chen, 2020; Çimen et al., 2022a; Corominas et al., 2008; Karas & Ozcelik, 2021; Michels & Costa, 2022; Sikora et al., 2017; Yang et al., 2013; Yang & Gao, 2016); while some others incorporate ergonomic aspects into the rebalancing problem (Çimen et al., 2022a; Otto & Scholl, 2011). AL rebalancing problems classification, which is summarized above, is illustrated in Figure 2.4.

Table 2.3 Assembly line rebalancing problem classification of the referenced works

Author (Year)	Type of product mix	Line layout	Variability of task times	Additional problems in the objective function
Yılmaz & Erol (2005)	Single-model	Straight line	Deterministic	
Gamberini et al. (2006)	Single-model	Straight line	Stochastic	
Corominas et al. (2008)	Single-model	Straight line	Deterministic	Worker reassignment
Gamberini et al. (2009)	Single-model	Straight line	Stochastic	
Ağpak (2010)	Single-model	Straight line & U-line	Deterministic	

Table 2.3 Continues

Author (Year)	Type of product mix	Line layout	Variability of task times	Additional problems in the objective function
Altemeier et al. (2010)	Mixed-model	Straight line	Deterministic	
Rahman (2010)	Single-model	Straight line	Deterministic	
Grangeon et al. (2011)	Mixed-model	Two-sided line	Deterministic	
Otto & Scholl (2011)	Single-model	Straight line	Deterministic	Ergonomic risk assessment
Oliveira et al. (2012)	Mixed-model	Straight line	Deterministic	
Makssoud et al. (2013)	Single-model	Straight line	Deterministic	
Yang et al. (2013)	Mixed-model	Straight line	Deterministic	Worker reassignment
Antoine et al. (2014)	Mixed-model	Straight line	Deterministic	
Celik et al. (2014)	Single-model	U-line	Stochastic	
Zha & Yu (2014)	Single-model	U-line	Deterministic	
Lai et al. (2015)	Mixed-model	Unspecified	Deterministic	
Makssoud et al. (2015)	Single-model	Straight line	Deterministic	
Mokhtari & Mozdgir (2015)	Single-model	Straight line	Deterministic	
Antoine et al. (2016)	Single-model	Unspecified	Deterministic	
Fattahi & Samouei (2016)	Single-model & Mixed-model	Straight line	Stochastic	Worker reassignment
Samadhi & Sumihartati (2016)	Single-model	Straight line	Deterministic	
Yang & Gao (2016)	Mixed-model	Straight line	Dynamic	Worker reassignment
D. Z. Zhang et al. (2016)	Mixed-model	Two-sided line	Deterministic	
Y. Li & Boucher (2017)	Single-model	Straight line	Dynamic	
Y. Li (2017)	Single-model	Straight line	Stochastic	
Sancı & Azizoğlu (2017)	Single-model	Straight line	Deterministic	
Sikora et al. (2017)	Single-model	Straight line & U-line	Deterministic	Worker reassignment
Belassiria et al. (2018)	Single-model	Straight line	Deterministic	
Ishigaki (2018)	Mixed-model	Straight line	Deterministic	Deciding item processing sequences
Y. Zhang et al. (2018a)	Single-model	Two-sided line	Deterministic	
Y. Zhang et al. (2018b)	Single-model	Two-sided line	Deterministic	
Y. Zhang et al. (2018c)	Single-model	Two-sided line	Deterministic	
Serin et al. (2019)	Single-model	U-line	Stochastic	
Budak & Chen (2020)	Single-model	Straight line	Dynamic	Worker reassignment
Hu et al. (2020)	Single-model	Two-sided line	Deterministic	
Y. Zhang et al. (2020)	Single-model	Two-sided line	Deterministic	
Karas & Ozcelik (2021)	Single-model	Straight line	Deterministic	Worker reassignment
Girit & Azizoğlu (2021)	Single-model	Straight line	Deterministic	
Y. Li et al. (2021)	Single-model	Straight line	Stochastic	
Michels & Costa (2021)	Single-model	Straight line	Deterministic	Worker reassignment
Çimen et al. (2022a)	Single-model	Straight line	Deterministic	Worker reassignment, Ergonomic risk assessment

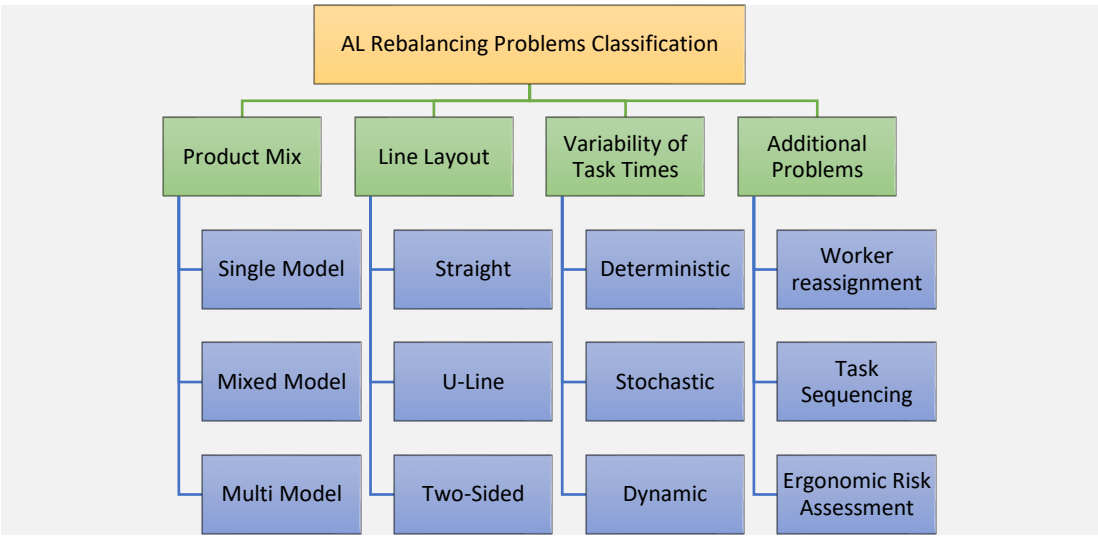


Figure 2.4 Assembly line rebalancing problems classification

2.3.1 The Type of Product Mix in The Assembly Line

According to the product mix, ALs are commonly distinguished as single, mixed, and multi-model (Scholl, 1999). A single-product model is manufactured in single-model ALs (Becker & Scholl, 2006). The AL was considered as single-model in 75% of the works in ALRBP literature (Antoine et al., 2016; Belassiria et al., 2018; Celik et al., 2014; Çimen et al., 2022a; Corominas et al., 2008; Gamberini et al., 2006, 2009; Karas & Ozcelik, 2021; Y. Li, 2017; Y. Li & Boucher, 2017; Makssoud et al., 2013, 2015; Mokhtari & Mozdgir, 2015; Otto & Scholl, 2011; Rahman, 2010; Samadhi & Sumihartati, 2016; Sancı & Azizoğlu, 2017; Serin et al., 2019; Sikora et al., 2017; Yılmaz & Erol, 2005; Zha & Yu, 2014; Y. Zhang et al., 2018a, 2018b, 2018c).

Different product models are manufactured in an arbitrarily mixed sequence (not in batches) in mixed-model ALs (Bukchin et al., 2002). 25% of the works were mixed-model in the reviewed ALRBP publications (Altemeier et al., 2010; Antoine et al., 2014; Grangeon et al., 2011; Ishigaki, 2018; Karas & Ozcelik, 2021; Lai et al., 2015; Oliveira et al., 2012; Yang et al., 2013; Yang & Gao, 2016; D. Z. Zhang et al., 2016).

In multi-model ALs, several product models are built in a series of model batches with intermediary set-up works (Becker & Scholl, 2006). There is no study on multi-model ALs in the reviewed papers.

The distribution of publications based on the type of product mix is presented in Figure 2.5. There are 41 papers in Figure 2.5 because Fattahi & Samouei (Karas & Ozcelik, 2021) worked on both single-model and mixed-model problems.

2.3.2 Line Layout

According to the line layout, serial (straight) lines and U-lines occur in ALBPs, which are partially predetermined by the flow of materials (Becker & Scholl, 2006). The most common layout type in the reviewed ALRBP studies is a straight line, a traditional AL layout in which workstations are sequentially arranged along a conveyor belt (Becker & Scholl, 2006).

In U-lines (Ağpak, 2010; Celik et al., 2014; Rahman, 2010; Serin et al., 2019; Sikora et al., 2017; Zha & Yu, 2014), the workstations are arranged along with a relatively narrow U, where both legs are close together. While some workers in a U-shape may alter the leg of the U on which they operate, some other workers do their work exclusively on one leg of the U. Thus, a product does not revisit a workstation, but a worker may do work on the product at two different workstations. This allows for a better balance of workloads with the increased number of worker-workstation combinations (Becker & Scholl, 2006).

A two-sided line (Grangeon et al., 2011; Hu et al., 2020; D. Z. Zhang et al., 2016; Y. Zhang et al., 2018a, 2018b, 2018c, 2020) is a serial line composed of two lines that are connected in parallel and have one opposite workstation on either side (left-hand side and right-hand side workstations (Bartholdi, 1993; Becker & Scholl, 2006).

In Figure 2.6, we presented the distribution of the reviewed papers based on the type of line layout. There are 42 papers in Figure 4b because Ağpak (2010) and Sikora

et al. (2017) worked on two different line layouts. Two-sided lines and U-lines are equally studied in the ALRBP literature, following straight lines.

2.3.3 Variability of Task Times

Task times may vary depending on the type of tasks and operators. While the variance of the task times is expected to be very small for simple tasks, complex and failure-sensitive operations may have considerably varying execution times. Scholl (1999) divided production lines into three groups due to the uncertainty of work times: deterministic, stochastic, and dynamic. The distribution of articles based on the variability of task times is presented in Figure 2.7.

When task time variability is sufficiently small, the task processing times are assumed to be deterministic. Deterministic task times are used in 75,6% of the articles in ALRBP literature. If the human work rate, skill, and motivation result in variations in task times, they are accepted as stochastic. Gamberini et al. (2006, 2009), Celik et al. (2014), Fattahi & Samouei (Karas & Ozcelik, 2021), Y. Li (2017), Serin et al. (2019), and Y. Li et al. (2022) studied stochastic task times. The task times are deemed to be dynamic if processing times fluctuate over time as a result of learning effects or advancements in the production process. Yang & Gao (2016), Y. Li & Boucher (2017), and Budak & Chen (2020) used dynamic task times in their studies.

Moreover, task times may be worker-dependent which means that task times get different values according to an assigned worker who performs the tasks in the workstation. In the reviewed papers, worker-dependent task times occur in two studies (Çimen et al., 2022a; Karas & Ozcelik, 2021).

2.3.4 Additional Problems That Are Incorporated in the ALRBP

As a result of the analysis, we saw that some additional problems were solved together with ALRBP in eight articles, presented in Figure 2.8. While Ishigaki (2018)

tried to find the best item processing sequence, Otto & Scholl (2011) and Çimen *et al*. (2022a) solved the ergonomic risk assessment problem together with ALRBP.

Moreover, the worker assignment problem was also incorporated in the ALRBP in nine publications (Budak & Chen, 2020; Çimen et al., 2022a; Corominas et al., 2008; Karas & Ozcelik, 2021; Michels & Costa, 2022; Sikora et al., 2017; Yang et al., 2013; Yang & Gao, 2016).

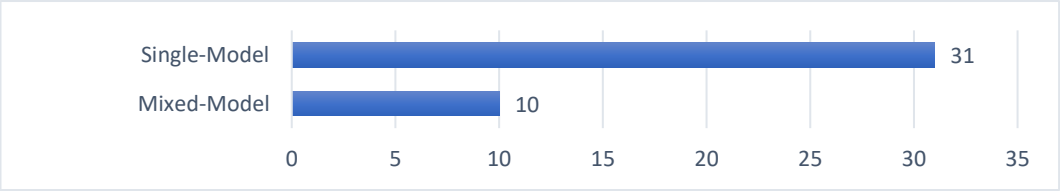


Figure 2.5 Distribution of publications based on the type of product mix in the assembly line

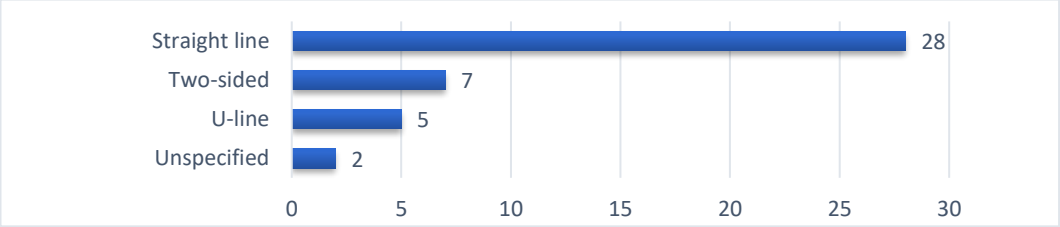


Figure 2.6 Distribution of publications based on the type of line layout

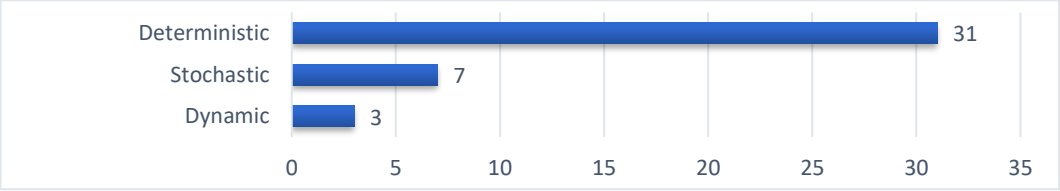


Figure 2.7 Distribution of publications based on the variability of task times

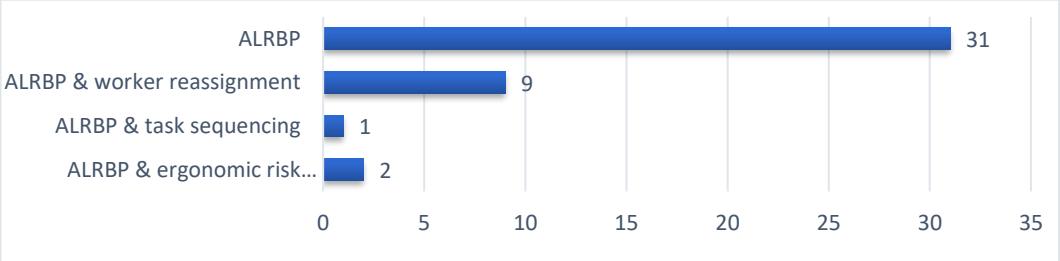


Figure 2.8 Distribution of publications based on solved problems

Corominas et al. (2008) tried to minimize the number of temporary (unskilled) workers who should work alongside skilled workers and cannot perform some tasks, leading to worker assignment restrictions. Yang et al. (2013) attempted to minimize the cost of worker retraining caused by assigning different tasks to the same worker in the rebalanced AL. Fattahi & Samouei (Karas & Ozcelik, 2021) had workers with different skill levels in their study who had different task processing times for each task. Yang & Gao (2016) had workers with different skills who could only perform the tasks in the rebalanced AL that had previously been assigned to them or their adjacent workstations. Sikora et al. (2017) tried to solve traveling worker ALRBP, in which a worker can operate in more than one workstation. More details about the papers were shared in section 3.6.

2.4 Objective Functions

An objective function is used to choose the best solution by evaluating its quality (Battaïa & Dolgui, 2013). As described in the introduction section, ALBP is widely categorized as Type 1 and Type 2 (see section 1) according to their objective function. In the ALRBP articles that we analyzed, various additional objectives were used. The most used objectives are minimizing the rebalancing costs, minimizing task relocation, and maximizing workload balance. The distribution of publications based on objective functions is presented in Figure 2.9. It can be concluded from Figure 2.9 that 13 of the research articles try to minimize rebalancing cost, 12 of them try to minimize the cycle time, 11 of them try to minimize task relocation, nine of them aim to minimize the number of workstations, 8 of them try to maximize the workload balance between workstations, and so on. Note that since the rebalancing problem is much more complicated than the traditional ALBP, many of the studies consider multiple objectives. This situation is presented in Figure 2.10. Combinations of two or more objective functions were used in 25 of the papers, while 39% of the papers have only one objective function (see Figure 2.10). Objective function information for each referenced work is presented in Table 2.4.

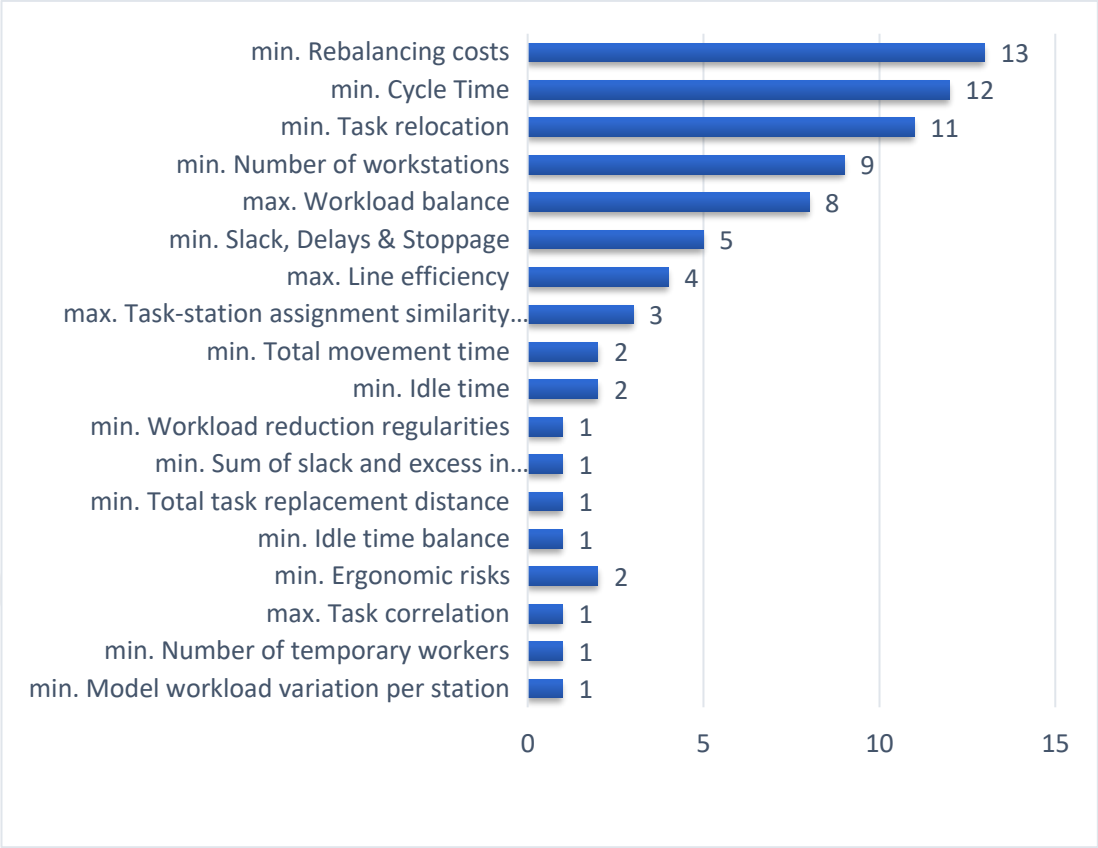


Figure 2.9 Distribution of publications based on objective functions

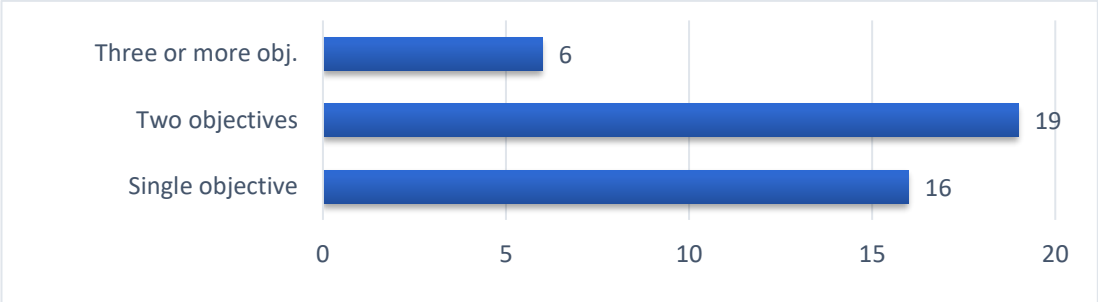


Figure 2.10 Distribution of publications based on the number of objective functions

Table 2.4 Objective functions of the referenced works

Author (Year)	Min. cycle time	Min. number of workstations	Max. line efficiency	Min. rebalancing cost	Min. task relocation	Max. workload balance	Other
Yılmaz & Erol (2005)				X			
Gamberini et al. (2006)				X			Maximize task-station assignment similarity with the initial solution
Corominas et al. (2008)							Minimize the number of temporary workers
Gamberini et al. (2009)				X			Maximize task-station assignment similarity with the initial solution
Ağpak (2010)		X					
Altemeier et al. (2010)			X	X			
Rahman (2010)			X				
Grangeon et al. (2011)		X			X	X	
Otto & Scholl (2011)		X					Minimize ergonomic risks
Oliveira et al. (2012)		X				X	
Makssoud et al. (2013)					X		
Yang et al. (2013)		X		X			Minimize workload variation at each station for different models
Antoine et al. (2014)							Minimize delay
Celik et al. (2014)				X			
Zha & Yu (2014)				X			Minimize total movement time
Lai et al. (2015)		X		X		X	Maximize task-station assignment similarity with the initial solution
Makssoud et al. (2015)		X			X		
Mokhtari & Mozdgir (2015)						X	Maximize task correlation
Antoine et al. (2016)							Minimize delay
Fattahi & Samouei (2016)	X						
Samadhi & Sumihartati (2016)	X	X		X			

Table 2.4 Continues

Author (Year)	Min. cycle time	Min. number of workstations	Max. line efficiency	Min. rebalancing cost	Min. task relocation	Max. workload balance	Other
Samadhi & Sumihartati (2016)				X			
Yang & Gao (2016)		X					
D. Z. Zhang et al. (2016)	X					X	
Y. Li & Boucher (2017)		X					Minimize idle time
Y. Li (2017)	X						Minimize idle time
Sancı & Azizoğlu (2017)	X				X		
Sikora et al. (2017)	X						Minimize total movement time
Belassiria et al. (2018)			X			Minimize idle time balance	
Ishigaki (2018)							Minimize line stoppage time
Y. Zhang et al. (2018a)	X				X		
Y. Zhang et al. (2018b)	X				X		
Y. Zhang et al. (2018c)	X				X		
Serin et al. (2019)				X			
Budak & Chen (2020)							Minimize the sum of slack and excess in production rates of all workstations
Hu et al. (2020)					X		
Y. Zhang et al. (2020)	X		X		X	X	
Karas & Ozcelik (2021)	X				X		
Girit & Azizoğlu (2021)	X				X	X	
Y. Li et al. (2021)				X			
Michels & Costa (2021)				X			Minimize workload reduction regularities
Çimen et al. (2022a)	X			X		X	Minimize ergonomic risks

The first column of Table 2.4 indicates the related reference. The second column states the papers whose objective is minimizing the cycle time (Çimen et al., 2022a; Girit & Azizoğlu, 2021; Karas & Ozcelik, 2021; Y. Li, 2017; Sancı & Azizoğlu, 2017; Sikora et al., 2017; D. Z. Zhang et al., 2016; Y. Zhang et al., 2018a, 2018b, 2018c, 2020). They studied the rebalancing problem of Type 2. The third column depicts the papers dealing with the Type 1 problem, in other words aiming to minimize the number of workstations (Ağpak, 2010; Grangeon et al., 2011; Lai et al., 2015; Y. Li & Boucher, 2017; Makssoud et al., 2015; Oliveira et al., 2012; Otto & Scholl, 2011; Yang et al., 2013; Yang & Gao, 2016). The fourth column of Table 2.4 shows the articles whose objective is to maximize the efficiency of the AL. This problem is referred as a Type E problem (Scholl, 1999) in the relevant literature. In Type E rebalancing problems, the efficiency of the assembly line is tried to be maximized by minimizing both the cycle time and the number of workstations simultaneously (Altemeier et al., 2010; Belassiria et al., 2018; Rahman, 2010; Y. Zhang et al., 2020). The fifth column represents the papers that try to minimize the rebalancing cost. The rebalancing cost might occur due to the relocation of tasks, worker movements, operators training, equipment installation and moving, etc. (Altemeier et al., 2010; Celik et al., 2014; Çimen et al., 2022a; Gamberini et al., 2006, 2009; Lai et al., 2015; Y. Li et al., 2022; Michels & Costa, 2022; Samadhi & Sumihartati, 2016; Serin et al., 2019; Yang et al., 2013; Yılmaz & Erol, 2005; Zha & Yu, 2014). The sixth column depicts studies that aimed at task relocation minimization. These studies try to assign tasks on the same workstation before rebalancing as much as possible (Grangeon et al., 2011; Makssoud et al., 2013, 2015; Sancı & Azizoğlu, 2017; Y. Zhang et al., 2018a, 2018b, 2018c). The seventh column gives the papers whose objective is workload balance maximization. The researchers try to assign tasks among workstations as smoothly as possible. In other words, they aim for each worker to work at workstations with nearly identical station times (Belassiria et al., 2018; Çimen et al., 2022a; Grangeon et al., 2011; Makssoud et al., 2013; Mokhtari & Mozdgir, 2015; D. Z. Zhang et al., 2016). Finally, the last column states if there is an additional objective.

2.5 Problem Constraints

ALBP-specific constraints such as precedence relations, the number of workstations, and cycle time are considered as general constraints. In some articles, researchers used additional constraints to make the problem better fit real-world ALRB problems, making them more complicated. Among all the ALRBP articles we analyzed, the most common constraints are task zoning and worker-related constraints.

All the constraints used in the analyzed papers and their distribution among the publications are presented in Figure 2.11. The list does not contain the ALBP-specific constraints, cycle time, task precedence, and the number of workstations. It can be interpreted from Figure 2.11 that constraints related to the number of workstations, workers, task-workstation allocation, and zoning are the most studied ones.

Problem-specific constraints of each referenced work are shown in Table 2.5. The first column of Table 2.5 indicates the related reference. The second column specifies articles that have a constraint related to the number of workstations (Budak & Chen, 2020; Çimen et al., 2022a; Girit & Azizoğlu, 2021; Hu et al., 2020; Karas & Ozcelik, 2021; Y. Li, 2017; Michels & Costa, 2022; Mokhtari & Mozdgir, 2015; Otto & Scholl, 2011; D. Z. Zhang et al., 2016). Task zoning constraints are categorized as negative zoning and positive zoning in the literature, which are used to forbid or force the assignment of specific tasks to the same workstation. The third and fourth columns represent positive and negative zoning constraints, respectively. Positive zoning constraints are considered in five papers (Belassiria et al., 2018; Corominas et al., 2008; Oliveira et al., 2012; Yılmaz & Erol, 2005; Y. Zhang et al., 2020), whereas negative zoning constraints are considered in seven papers (Belassiria et al., 2018; Corominas et al., 2008; Makssoud et al., 2013, 2015; Oliveira et al., 2012; Yılmaz & Erol, 2005; Y. Zhang et al., 2020). Note that all papers with positive zoning constraints also incorporate negative zoning constraints simultaneously. In some cases, a task should be operated or certainly not be operated on a particular workstation. This situation is referred as task-workstation allocation restrictions in the relevant literature; and is stated in the fifth column (Altemeier et al., 2010; Belassiria et al., 2018; Budak

& Chen, 2020; Karas & Ozcelik, 2021; Lai et al., 2015; Sikora et al., 2017; Yılmaz & Erol, 2005). Moreover, some studies consider workers-related constraints. The sixth column represents studies that consider workers as unique workforces (Çimen et al., 2022a; Corominas et al., 2008; Grangeon et al., 2011; Karas & Ozcelik, 2021; Michels & Costa, 2022; Sikora et al., 2017; Yang et al., 2013; Yang & Gao, 2016). Also, one article takes into consideration worker-workstation restrictions, which means that some workers should be assigned or certainly not be assigned to particular workstations during the rebalancing process (Budak & Chen, 2020).

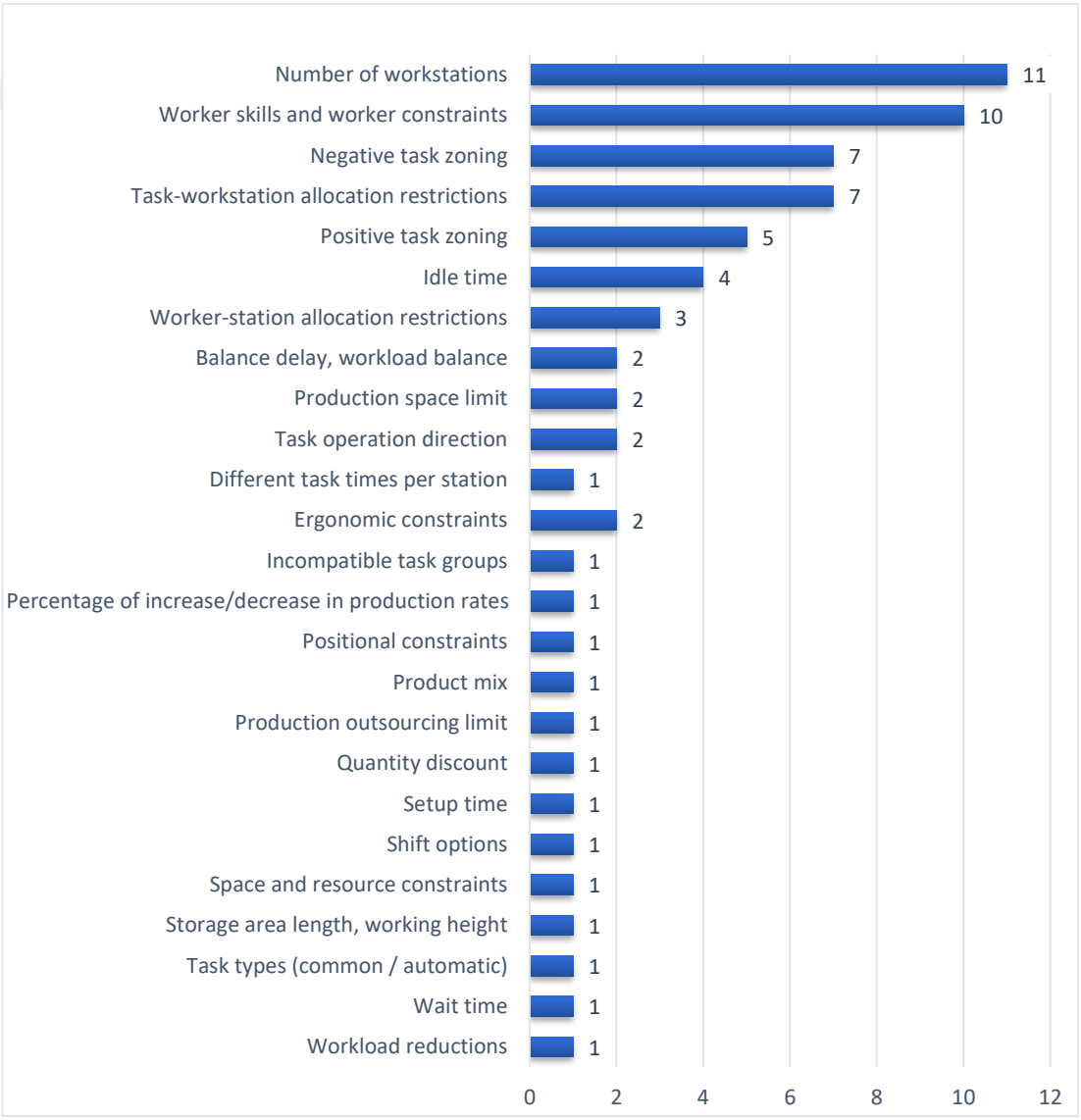


Figure 2.11 Distribution of publications based on problem constraint

Table 2.5 Problem-specific constraints of the referenced works

Author (Year)	Number of workstations	Positive task zoning	Negative task zoning	Task-Workstation allocation restrictions	Worker constraints	Other
Yılmaz & Erol (2005)		X	X	X		Balance delay
Gamberini et al. (2006)						
Corominas et al. (2008)		X	X		Worker skills	Positional constraints
Gamberini et al. (2009)						
Ağpak (2010)						
Altemeier et al. (2010)				X		
Rahman (2010)						
Grangeon et al. (2011)					X	Storage area length, Working height
Otto & Scholl (2011)	X					Ergonomic constraints
Oliveira et al. (2012)		X	X			
Makssoud et al. (2013)			X			
Yang et al. (2013)					X	
Antoine et al. (2014)						
Celik et al. (2014)						
Zha & Yu (2014)						
Lai et al. (2015)				X		
Makssoud et al. (2015)			X			
Mokhtari & Mozdgir (2015)	X					
Antoine et al. (2016)						
Fattahi & Samouei (2016)	X				Worker skills	Product mix
Samadhi & Sumihartati (2016)						Production space available; Maximum number of outsourcings; Quantity discount
Yang & Gao (2016)					Worker skills	

Table 2.5 Continues

Author (Year)	Number of workstations	Positive task zoning	Negative task zoning	Task-Workstation allocation restrictions	Worker constraints	Other
D. Z. Zhang et al. (2016)	X					Incompatible task groups
Y. Li & Boucher (2017)						
Y. Li (2017)	X					
Sancı & Azizoğlu (2017)						
Sikora et al. (2017)				X	Worker skills	Worker-Station allocation restrictions; Different task types (common /automatic); Different task processing times per station; Setup times
Belassiria et al. (2018)		X	X	X		
Ishigaki (2018)						
Y. Zhang et al. (2018a)						Task operation direction; Wait time; Idle time; Workload balance
Y. Zhang et al. (2018b)						Idle time
Y. Zhang et al. (2018c)						Task operation direction; Idle time
Serin et al. (2019)						
Budak & Chen (2020)	X			X	Worker-Workstation allocation restrictions	Percentage of increase/decrease in production rates
Hu et al. (2020)	X					
Y. Zhang et al. (2020)		X	X	X		Idle time, space, and resource constraints
Karas & Ozcelik (2021)	X				Worker skills	Worker-Station allocation restrictions
Girit & Azizoğlu (2021)	X					
Y. Li et al. (2021)						Delay
Michels & Costa (2021)	X				X	Workload reductions, shift options (8h, 6h, 4h)
Çimen et al. (2022a)	X				X	Ergonomic constraints

2.6 Causes of Rebalancing

There may be several reasons that break the line balance and cause the need for rebalancing. Disturbances that led to the occurrence of the rebalancing problem are given in Table 2.6. These disturbances are listed as: demand variations, changes in cycle time, changes in task precedence, adding tasks/removing tasks, changes in task times, changes in product design, and machine breakdown/shutdown. Product demand variations are foremost taken the lead as a rebalancing cause in the ALRBP literature. In today's competitive production environment, businesses must be adaptable and agile to fulfill client requirements. Lead times are becoming shorter, and products are becoming more specific with various options. Therefore, product demand changes affect the line balance and cycle time, or task time needs to be updated to rebalance the AL.

Moreover, technological developments are increasing rapidly in today's conditions. As a result of technological developments, some of the task times might be decreased, some tasks might be combined with others, and some tasks might be eliminated. Consequently, changes in task processing times and precedence relations might occur. Also, companies that have to cope with customer needs and requests in today's competitive environment should change their product design more often. Finally, one of the most common disturbances is machine breakdown. Although preventive maintenance is executed regularly, sudden machine breakdowns might occur in real-life manufacturing systems. These are the most common reasons for the need for rebalancing.

Demand or cycle time fluctuations caused rebalancing in 78% of the articles we reviewed. Product design changes are the following most observed rebalancing reason (Ağpak, 2010; Altemeier et al., 2010; Karas & Ozcelik, 2021; Lai et al., 2015; Otto & Scholl, 2011; Yang et al., 2013; Yang & Gao, 2016; Y. Zhang et al., 2018c, 2020), with nine papers. Other reasons are task time changes, machine breakdown/shutdown, adding/removing tasks, and task precedence changes. The encountered reasons and their distribution among ALRBP articles are presented in Figure 2.12. The causes for

the rebalancing of each referenced work are presented in Table 2.6. In Table 2.6, the first column gives the reference, and the remaining columns indicate the relevant disturbance. For example, Yılmaz & Erol (2005) studied the rebalancing problem, which occurs due to changes in cycle time and task processing times.

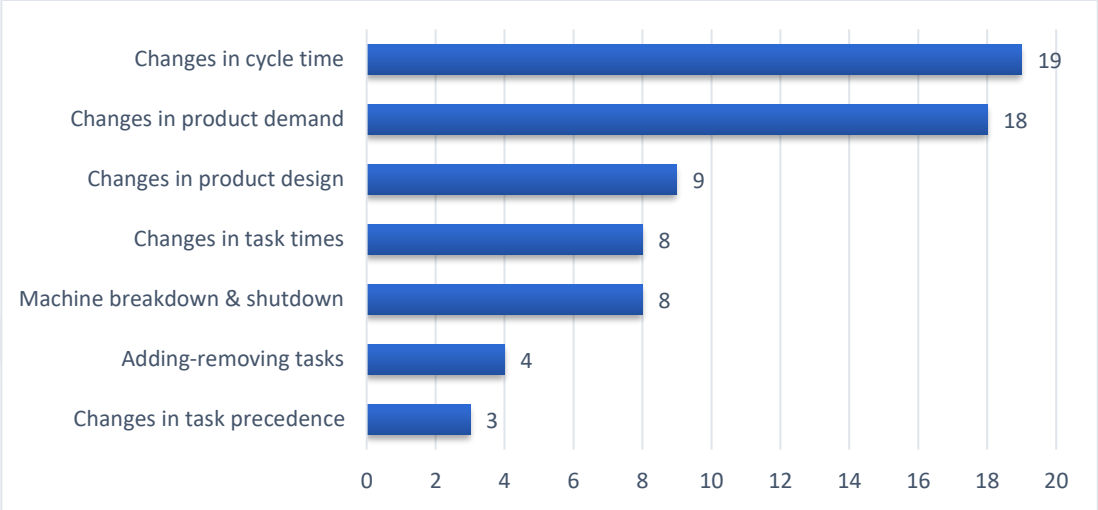


Figure 2.12 Distribution of publications based on causes of rebalancing

Table 2.6 Causes for the rebalancing of the referenced works

Author (Year)	Demand variations	Changes in cycle time	Changes in task precedence	Adding tasks / removing tasks	Changes in task times	Changes in product design	Machine breakdown/shutdown
Yılmaz & Erol (2005)		X			X		
Gamberini et al. (2006)		X	X				
Corominas et al. (2008)	X	X					
Gamberini et al. (2009)	X	X	X		X	X	
Ağpak (2010)		X					
Altemeier et al. (2010)	X					X	
Rahman (2010)	X						
Grangeon et al. (2011)	X						
Otto & Scholl (2011)						Ergonomic problems	
Oliveira et al. (2012)	X						
Makssoud et al. (2013)				X			
Yang et al. (2013)	X					Changes in product mix	
Antoine et al. (2014)							X
Celik et al. (2014)		X					
Zha & Yu (2014)	X	X					
Lai et al. (2015)	X			X	X	X	
Makssoud et al. (2015)				X			
Mokhtari & Mozdgir (2015)	X	X					
Antoine et al. (2016)							X
Fattahi & Samouei (2016)	X					X	
Samadhi & Sumihartati (2016)	X						
Yang & Gao (2016)	X					X	
D. Z. Zhang et al. (2016)		X					
Y. Li & Boucher (2017)					X		

Table 2.6 Continues

Author (Year)	Demand variations	Changes in cycle time	Changes in task precedence	Adding tasks / removing tasks	Changes in task times	Changes in product design	Machine breakdown/shutdown
Y. Li (2017)		X			X		
Sancı & Azizoğlu (2017)							X
Sikora et al. (2017)					X		
Belassiria et al. (2018)	X						
Ishigaki (2018)	X						X
Y. Zhang et al. (2018a)	X	X	X	X	X	X	
Y. Zhang et al. (2018b)		X					
Y. Zhang et al. (2018c)		X					
Serin et al. (2019)	X	X					
Budak & Chen (2020)		X					X
Hu et al. (2020)		X					
Y. Zhang et al. (2020)						X	
Karas & Ozcelik (2021)							X
Girit & Azizoğlu (2021)							X
Y. Li et al. (2021)		X			X		Disruptions
Michels & Costa (2021)	X	X					
Çimen et al. (2022a)		X					

2.7 Solution Methods

There are two main types of solution methods: exact and approximate. The most important advantage of exact methods is that they guarantee the global optimum result. However, exact methods are appropriate for small-sized problems, but they cannot solve large-size instances in a reasonable time. As the problem size increases, solution approaches such as constraint programming can be used in the relevant literature. Constraint programming still yields the best results for larger problems. If the size of the problem gets complicated, the constraint programming approach will no longer yield results in a meaningful time interval. Approximation methods such as heuristics and meta-heuristics have been developed as a remedy. Although these approximate approaches do not guarantee the global optimum solution, they give very successful results in a reasonable computational time. Moreover, simulation-based solution approaches also do not guarantee the optimum solution. However, they can incorporate stochastic data, such as task processing times, in order to represent real life. In summary, while approximate methods do not guarantee optimality, they can be preferred in large-scale and more complicated problems with several constraints.

The distribution of the solution methods used in the articles is presented in Figure 2.13. Oliveira et al. (2012), Antoine et al. (2016), Karas & Ozcelik (Karas & Ozcelik, 2021), and Girit & Azizoğlu (2021) used hybrid approaches (exact and approximate methods together) while solving ALRBPs. Thus, there are 45 publications in Figure 2.13 in total.

Mixed-integer linear programming and branch & bound methods are mostly preferred exact methods in the articles. Heuristic methods, ant colony optimization, differential evolution, genetic algorithms, and local search are the most preferred approximate methods. Solution methodologies, which were used in the reviewed studies, are presented in Table 2.7. It can be concluded from Table 2.7 that some articles deal with case studies (Altemeier et al., 2010; Antoine et al., 2014; Belassiria et al., 2018; Budak & Chen, 2020; Çimen et al., 2022a; Corominas et al., 2008; Gamberini et al., 2009; Grangeon et al., 2011; Lai et al., 2015; Y. Li & Boucher, 2017;

Michels & Costa, 2022; Oliveira et al., 2012; Rahman, 2010; Samadhi & Sumihartati, 2016; Sikora et al., 2017; D. Z. Zhang et al., 2016; Y. Zhang et al., 2018c, 2020); some articles study benchmark problems (Ağpak, 2010; Belassiria et al., 2018; Celik et al., 2014; Çimen et al., 2022a; Girit & Azizoğlu, 2021; Hu et al., 2020; Ishigaki, 2018; Karas & Ozcelik, 2021; Y. Li, 2017; Y. Li et al., 2022; Michels & Costa, 2022; Mokhtari & Mozdgir, 2015; Otto & Scholl, 2011; Sancı & Azizoğlu, 2017; Serin et al., 2019; Sikora et al., 2017; Yang et al., 2013; Yang & Gao, 2016; Zha & Yu, 2014; Y. Zhang et al., 2018a, 2018b, 2020); and some articles solve the randomly generated problems (Gamberini et al., 2006, 2009; Makssoud et al., 2013, 2015; Yang & Gao, 2016; Yılmaz & Erol, 2005; Zha & Yu, 2014).

Here are some examples of **heuristic** and **meta-heuristic** methods that have been applied to assembly line rebalancing problems.

2.7.1 Genetic Algorithms (GA)

Genetic algorithms are a type of heuristic optimization method that is inspired by the process of natural selection in biology. They are particularly well-suited to problems with a large number of variables or solutions that are difficult to represent explicitly.

In a GA, solutions to the optimization problem are represented as strings of symbols called "chromosomes." A population of chromosomes is initialized randomly and then evolves over time through a series of iterations, or "generations." At each generation, the chromosomes are evaluated using an objective function, and the best solutions are selected to produce the next generation.

The evolution of the chromosomes is guided by a set of genetic operators, such as crossover (recombination) and mutation, which can generate new solutions by combining the characteristics of existing solutions in different ways. The genetic operators are probabilistic, which means that they do not always produce an

improvement, but they can allow the GA to explore a larger portion of the search space and avoid getting stuck in local optima.

The following steps are typically followed to solve an assembly line rebalancing problem using GAs:

1. Define the problem and the objective function.
2. Initialize a population of candidate solutions, which can be done randomly or using heuristics.
3. Evaluate the fitness of each candidate solution using the objective function.
4. Select the best solutions from the population using selection criteria such as fitness proportionate selection or tournament selection.
5. Generate new solutions using crossover and mutation operators.
6. Repeat steps 3-5 until a satisfactory solution is found or a termination criterion is met.

In the context of assembly line rebalancing, GAs can be used to optimize the number and arrangement of workstations and tasks, as well as the assignment of workers to tasks. The objective function can be defined as minimizing production time, maximizing throughput, or minimizing cost, among other objectives.

2.7.2 Simulated Annealing (SA)

Simulated annealing is a heuristic method that is inspired by the annealing process used in metallurgy to strengthen metals. It involves starting with a random solution and gradually refining it through a series of small changes while also allowing for the occasional "bad" move in order to avoid getting stuck in local optima.

In SA, a solution to the optimization problem is represented as a "state" in the search space. The optimization process begins with a random initial state and then iteratively generates new states by making small random changes to the current state. The new

state is accepted or rejected based on its objective function value and a temperature parameter, which controls the probability of accepting worse solutions.

As the temperature decreases over time, the probability of accepting worse solutions decreases, and the optimization process becomes more deterministic. This allows the simulated annealing algorithm to gradually refine the solution and avoid getting stuck in local optima.

It is generally considered to be a robust and effective optimization method, but it can be sensitive to the choice of parameters and may require some tuning to achieve good performance.

2.7.3 Differential Evolution (DE)

Differential evolution is a heuristic optimization method that is used to solve complex optimization problems. It is a population-based method, which means that it involves representing solutions as a set of individuals (or "candidate solutions") that evolve over time through a series of iterations. DE works by generating new candidate solutions by combining the characteristics of existing solutions in different ways. It uses a "differential" component, which is a vector of differences between pairs of solutions, to guide the search for good solutions. The new candidate solutions are then evaluated using an objective function, and the best solutions are kept for the next iteration. It is generally considered to be a robust and effective optimization method, but it can be sensitive to the choice of parameters and may require some tuning to achieve good performance.

2.7.4 Tabu Search and Local Search

Tabu search and *local search* are heuristic optimization methods that involve iteratively making minor changes to a solution and evaluating the impact on the objective function, with the goal of finding a good or optimal solution.

Tabu search is a heuristic method that uses a "memory" of previously visited solutions to guide the search for good solutions. It involves maintaining a list of "tabu" solutions that are not allowed to be revisited in order to encourage the exploration of new areas of the search space. The tabu list is updated at each iteration, and solutions that were previously tabu may be allowed to be revisited after a certain period of time.

Local search is a heuristic method that involves making minor changes to a solution and evaluating the impact on the objective function, with the goal of finding a local optimum. Local search algorithms typically start with an initial solution and iteratively search the neighborhood of that solution for improvements.

Both tabu search and local search can be effective in solving assembly line rebalancing problems, but they may not always find the global optimum. It is important to carefully evaluate the trade-offs between solution quality and computational complexity when choosing an optimization technique for a particular problem.

2.7.5 Multi-Objective Optimization (MOO)

Multi-objective optimization is a meta-heuristic method that involves finding solutions that optimize multiple conflicting objectives simultaneously. This can be particularly useful in assembly line rebalancing problems, as there are often multiple objectives that need to be considered, such as minimizing production time, minimizing cost, and maximizing quality.

In multi-objective optimization, the optimization problem is defined in terms of a set of objective functions, each of which represents one of the objectives to be optimized. The solution to the optimization problem is represented as a set of points in the objective space, called the "Pareto front," which represents the trade-offs between the different objectives. A solution is said to be "non-dominated" if there is no other solution that is better in all objectives.

Multi-objective optimization algorithms search for the Pareto front by generating and evaluating a set of candidate solutions and selecting the non-dominated solutions for the next iteration. The search process can be guided by a variety of meta-heuristics, such as evolutionary algorithms, particle swarm optimization, or ant colony optimization, which can provide good exploration and exploitation of the search space.

Multi-objective optimization can be an effective approach to solving assembly line rebalancing problems, as it allows for the consideration of multiple objectives simultaneously. However, it can be computationally intensive and may require specialized software and expertise to implement.

2.7.6 Particle Swarm Optimization (PSO)

Particle swarm optimization is a meta-heuristic optimization method that is inspired by the behavior of swarms of animals, such as birds or fish. It involves representing solutions as "particles" that move through the search space and using techniques such as velocity updates and social interactions to guide the search for good solutions.

In PSO, the optimization problem is defined in terms of an objective function, which represents the measure of the quality of the solutions. The solution to the optimization problem is represented as a set of particles, each of which is associated with a position in the search space and a velocity.

At each iteration, the particles update their position and velocity based on their current position, the best position they have visited so far (called the "personal best"), and the best position visited by any particle in the swarm (called the "global best"). The position and velocity updates are governed by a set of equations that incorporate the influence of the personal best and the global best, as well as a set of control parameters that can be adjusted to control the behavior of the algorithm.

2.7.7 Ant Colony Optimization (ACO)

Ant colony optimization is a meta-heuristic optimization method that is inspired by the foraging behavior of ants. It involves representing solutions as "trails" of pheromones that are laid down by "ants" as they explore the search space and using techniques such as pheromone evaporation and reinforcement learning to guide the search for good solutions.

In ACO, the optimization problem is defined in terms of an objective function, which represents the measure of the quality of the solutions. The solution to the optimization problem is represented as a set of trails of pheromones, each of which corresponds to a possible solution.

At each iteration, a set of ants is released from a starting point in the search space, and they move through the space by following the trails of pheromones. As they move, they lay down additional pheromones on the trails, which reinforce the attractiveness of those trails to other ants. The strength of the pheromone decays over time through a process called pheromone evaporation, which encourages the ants to explore new areas of the search space.

2.7.8 Simulation-Based Approaches

Simulation-based approaches are optimization techniques that involve using computer simulations to model and analyze the production process in an assembly line. These techniques can be used to identify potential improvements and optimize the production process, including the assignment of tasks and workers.

Simulation-based approaches can be particularly useful in cases where it is difficult or impossible to obtain precise data on the production process or where the production process is complex and involves significant uncertainty. They can also be used to evaluate the performance of different design and operating scenarios and to identify bottlenecks or constraints in the production process.

There are a variety of simulation-based approaches that can be used to solve assembly line rebalancing problems, including discrete event simulation, agent-based simulation, and system dynamics simulation. These approaches can be used in combination with optimization techniques, such as heuristics or meta-heuristics, to identify the optimal solutions.

Simulation-based approaches can be time-consuming and computationally intensive and may require specialized software and expertise to implement. However, they can provide valuable insights and can be an effective tool for improving the efficiency and effectiveness of the production process.

Table 2.7 Solution methods and test data types for solution's performance evaluation of the referenced studies

		Test Data Type for Method Evaluation		
Author (Year)	**Solution Methodology**	**Case Study**	**Benchmark problems**	**Randomly generated**
Yılmaz & Erol (2005)	Branch and bound			X
Gamberini et al. (2006)	TOPSIS-based heuristic			X
Corominas et al. (2008)	Binary linear programming (LP)	X		
Gamberini et al. (2009)	Multiple single-pass heuristics, Multi-objective GA	X		X
Ağpak (2010)	COMSOAL-based heuristic		X	
Altemeier et al. (2010)	Mathematical optimization method	X		
Rahman (2010)	Least predecessor rule	X		
Grangeon et al. (2011)	Heuristic algorithm with three phases	X		
Otto & Scholl (2011)	Two-stage heuristic approach, based on SALOME and simulated annealing		X	
Oliveira et al. (2012)	Branch and bound, Heuristic method	X		
Makssoud et al. (2013)	Mixed-integer LP			X
Yang et al. (2013)	Multi-objective GA, Local search		X	
Antoine et al. (2014)	Discrete event systems theory, Reachability analysis, Communicating automata	X		
Celik et al. (2014)	ACO		X	
Zha & Yu (2014)	Filtered beam search, Hybrid ACO		X	X
Lai et al. (2015)	Improved dual-population GA	X		
Makssoud et al. (2015)	Mixed-integer LP			X
Mokhtari & Mozdgir (2015)	Adapted differential evolution, Taguchi method		X	
Antoine et al. (2016)	Integer LP, Iterated local search	X		
Fattahi & Samouei (2016)	Heuristic method		X	

Table 2.7 Continues

Author (Year)	Solution Methodology	Test Data Type for Method Evaluation		
		Case Study	Benchmark problems	Randomly generated
Samadhi & Sumihartati (2016)	Integer LP	X		
Yang & Gao (2016)	Branch, bound, and remember		X	X
D. Z. Zhang et al. (2016)	Heuristic method	X		
Y. Li & Boucher (2017)	Backward induction rule-based heuristic method	X		
Y. Li (2017)	Branch and bound		X	
Sanci & Azizoğlu (2017)	Mixed-integer LP, Branch and bound		X	
Sikora et al. (2017)	Mixed integer LP	X	X	
Belassiria et al. (2018)	Hybrid GA with a rule-based heuristic procedure	X	X	
Ishigaki (2018)	Heuristic method		X	
Y. Zhang et al. (2018a)	Modified nondominated sorting GA	X		
Y. Zhang et al. (2018b)	Constraint programming, Workstation-oriented heuristic		X	
Y. Zhang et al. (2018c)	Push rule-based heuristic method		X	
Serin et al. (2019)	GA		X	
Budak & Chen (2020)	Mixed-integer LP	X		
Hu et al. (2020)	Modified beam search algorithm		X	
Y. Zhang et al. (2020)	Improved imperial competition algorithm	X	X	
Karas & Ozcelik (2021)	Mixed-integer LP, Artificial bee colony algorithm		X	
Girit & Azizoğlu (2021)	Mixed-integer LP, tabu search algorithm		X	
Y. Li et al. (2021)	Heuristic method		X	
Michels & Costa (2021)	Mixed-integer LP	X	X	
Çimen et al. (2022a)	Heuristic method	X	X	

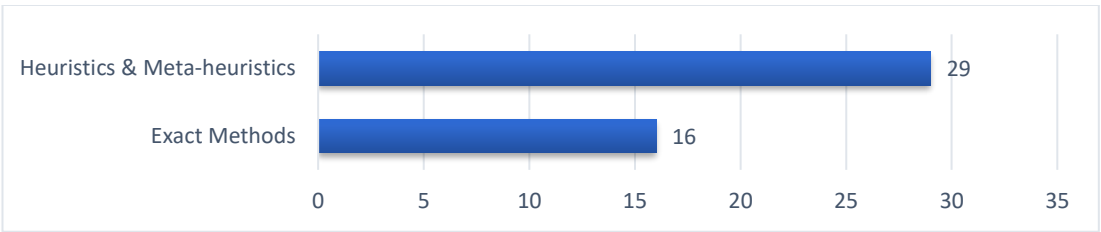


Figure 2.13 Distribution of publications based on solution methods

2.8 Test Data Types for Solution's Performance Evaluation

Experiments are made to validate the results of the proposed methods, better visualize and describe the problem to the readers and compare the proposed method

with other solution approaches. We have categorized experimental data into three categories: case studies, benchmark problems, and randomly generated test problems. In case studies, actual data from a real-world production environment is used by researchers. In randomly generated test problems, initial problem instances and all necessary variables are developed according to the defined problem environment. Lastly, benchmark problems are problem-specific, well-defined, and formatted data that were generated and shared with the research community to help researchers compare the proposed solution method with previous results. In Figure 2.14, the distribution of experimental settings used in the reviewed articles is presented. Some researchers used two different experimental data while testing their methods (Belassiria et al., 2018; Gamberini et al., 2009; Michels & Costa, 2022; Sikora et al., 2017; Yang & Gao, 2016; Zha & Yu, 2014; Y. Zhang et al., 2020). Thus, there are 49 publications in Figure 2.14 in total. Test problem types, which were used in the reviewed studies, are presented in Table 2.7.

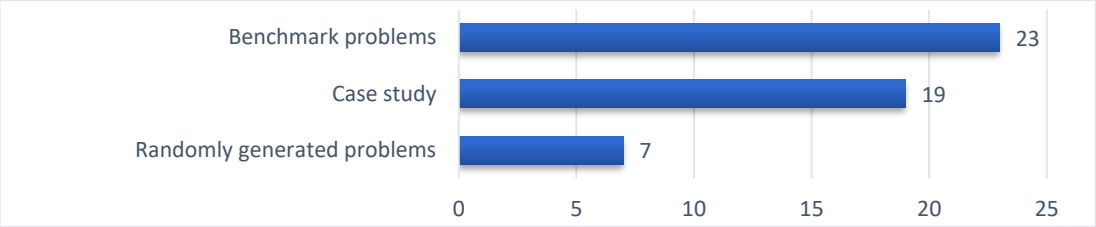


Figure 2.14 Distribution of publications based on test data types for solution evaluation

2.9 Overview of ALRBP Papers

Several studies dealt with ALRBP for single-model, straight assembly lines (Belassiria et al., 2018; Corominas et al., 2008; Gamberini et al., 2006, 2009; Karas & Ozcelik, 2021; Y. Li, 2017; Y. Li et al., 2022; Y. Li & Boucher, 2017; Makssoud et al., 2013, 2015; Mokhtari & Mozdgir, 2015; Rahman, 2010; Sancı & Azizoğlu, 2017; Yılmaz & Erol, 2005).

Belassiria et al. (2018) proposed a GA that was hybridized with a heuristic priority rule-based procedure for solving single-model ALRBP type-E on straight lines with deterministic task times. In this study, product demand changes were considered as

causes for re-balancing. The objectives were maximizing line efficiency (type E), maximizing workload balance, and minimizing cost subject to task precedence relations, positive & negative task zoning and assignment restrictions were additional constraints. The authors evaluated the results with a real-life case study on the harness manufacturing company and used benchmark data from Scholl (1999).

Corominas et al. (2008) developed a binary linear programming procedure for solving a single-model deterministic rebalancing problem in a motorcycle-assembly plant. Demand changes in the spring and summer months caused the need for increasing production (decreasing cycle time) and re-balancing the line. For that purpose, the company hired temporary (unskilled) workers who needed more time to implement the tasks and could not perform some of the tasks. The objective was to minimize the number of unskilled workers while considering cycle time, task precedence, task zoning constraints, and worker assignment constraint, which skilled workers could not work without a skilled worker alongside. The proposed method improved the current solution by decreasing the need for temporary.

Fattahi & Samouei (Karas & Ozcelik, 2021) developed a heuristic-simulation algorithm for solving single and mixed model ALRBP type-2 on straight lines together with worker assignment. In this study, product demand variations triggered the AL rebalancing. Researchers used stochastic task times and considered task precedence and worker skill constraints while trying to minimize cycle time on a fixed number of workstations (and workers). The problem was modeled, and benchmarked test problems were solved to compare the results of the proposed method.

Gamberini et al. (2006) considered task reassignment as an objective function and offered a single-pass heuristic with a multi-attribute decision-making system based on the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). Researchers focused on solving a single-model stochastic rebalancing problem on straight lines. Due to the changes in cycle time and precedence relations, a rebalancing problem occurred. The authors stated an index to quantify the similarity of the task

assignment between original and rebalanced lines, called the similarity factor. The objective was to minimize task reassignment by minimizing the similarity factor.

Gamberini et al. (2009) proposed a multiple single-pass heuristic algorithm for solving single-model stochastic rebalancing problems on straight lines. Changes in product design, task precedence, task times, and cycle time caused the need for re-balancing in the current problem. The objective was to maximize the similarity of task assignments and to minimize the total expected assembly cost subject to cycle time and task precedence constraints. In comparison with the proposed heuristic, a multi-objective genetic algorithm (moGA) was developed. However, in terms of similarity optimization, moGA was not able to achieve consistent results, and better results were produced by the proposed heuristic. A randomly generated test problem was solved to evaluate the proposed method, and finally, an illustrative case study was provided by the authors that demonstrated the advantages of utilizing the suggested multiple single-pass heuristics.

Y. Li (2017) studied single-model ALRBP type-2 on straight lines with stochastic task times where AL was operated by robots. The author proposed the ENCORE algorithm, which is a B&B method. In this study, cycle time fluctuations and changes in task times after task learning triggered the rebalancing. Task precedence relations were considered as an additional constraint. Benchmark datasets were used to validate the performance of the ENCORE algorithm. Finally, the results were compared with the branch and bound remember algorithm (BBR) and SALOME2.

Y. Li et al. (2022) studied the AL rebalancing problem due to sudden disruption. They considered stochastic operation times, and the goal was minimizing total cost. The authors compared two solution approaches: i) periodic and ii) data-driven rebalancing policies. The results show that a data-driven rebalancing policy is better than a periodic rebalancing policy.

Y. Li & Boucher (2017) used line balancing algorithms of Scholl & Becker (2006) as a solution method for solving single-model ALRBP type-1 on dynamic automated

lines where task times were dynamically changing because of collaborative learning. Task time changes were the cause of rebalancing, and the objectives were reducing the total number and total idle time of workstations. A hypothetical case study was solved for the evaluation of the method.

Makssoud et al. (2013) proposed a mixed integer programming approach for solving single-model ALRBP with deterministic task times on a straight AL. The objective was to minimize the number of changes while considering task precedence relations and negative task zoning constraints. Researchers have solved an illustrative case study where rebalancing was triggered by product modification by adding & removing tasks. The model was coded in C++, and results showed that the proposed model could solve problems with up to 40 tasks.

Makssoud et al. (2015) developed a new mathematical model and exact solution method for solving single-model ALRBP type-1 on straight lines with deterministic task times. In their study, adding and removing tasks triggered the rebalancing, and the objectives were minimizing the number of workstations and the number of modifications while considering negative task zoning constraints. Researchers solved an illustrative case study to test the proposed exact method. The experiments showed that the model could solve up to 34 task problems in a reasonable time.

Mokhtari & Mozdgir (2015) suggested a differential evolution algorithm (DEA) for solving single-model ALRBP in which a new demand-related cycle time was needed (CT). Researchers tried to optimize two objectives while re-scheduling the tasks to reduce the current CT: (i) minimization of the incurred costs and (ii) minimization of the non-smoothing reconfigured line. It is assumed that the assembly line was straight shaped with no buffer space, containing one family of products that all had deterministic processing times while formulating the model. To set control parameters of DEA without time-consuming manual operations, a detailed statistical experiment based on Taguchi experimental design was performed. Researchers considered two approaches to address infeasibility that occurred during the solving procedure: (i) Omitting infeasible solution and (ii) Considering the penalty for the objective function

of the infeasible solution. After testing with benchmark data, the authors found that the penalty method performed superior to the omitting method in terms of speed of convergence and quality of solutions.

Rahman (2010) focused on straight single-model assembly lines. The objectives of his case study were to investigate the performance of AL, to identify the opportunity of rebalancing the AL, and to consider alternative optimal solutions. The author defined the most significant problems faced while managing assembly lines are boredom, absenteeism, personnel turnover, and dissatisfaction among workers. He gave operation sequence, task times, task precedence requirements, efficiency, work capacity, and demand and evaluated the AL in detail. The line was re-balanced by moving tasks using the least predecessor rule. It has also mentioned that, while making re-balancing, it is important to consider other facts, whether the system is capital-intensive or labor-intensive. According to different scenarios, solution alternatives were suggested, such as team assembly, multiple, U-shaped, or mixed-model assembly lines to provide flexibility to changes.

Sancı & Azizoğlu (2017) stated a GA that was hybridized with a priority rule-based heuristic procedure for solving ALRBP. The objectives of their study were maximizing line efficiency and workload balance while minimizing total cost. The first method was a mixed-integer linear programming (MILP) algorithm and the second was Branch and Bound (B&B). Researchers proposed a sequential approach (SA) that used successive solutions of MILP and B&B to find the exact set of nondominated objective vectors. In their study, rebalancing was triggered when there was a workstation breakdown or shutdown. Task precedence relations were considered as an additional constraint, and the method was tested with benchmark datasets of Scholl (1999).

A particular branch & bound (B&B) technique has been developed by Yılmaz & Erol (2005) to solve ALRBP with deterministic task times in single model straight lines. Cycle time and task time changes were considered as causes for re-balancing. The objective was to minimize the total cost of moving tasks between workstations subject to; cycle time, task precedence, task zoning, moving, and balance delay

constraints. Researchers applied some stopping and special fathoming rules to reduce total solution time. To evaluate the suggested solution procedure, randomly generated test samples with varying sizes were solved.

Few studies deal with the single model U-shaped ALRBP (Ağpak, 2010; Celik et al., 2014; Serin et al., 2019; Sikora et al., 2017; Zha & Yu, 2014). Workers can work on one of each sub-line or both of them simultaneously in U-shaped assembly lines (Ramezanian & Khalesi, 2021).

Ağpak (2010) concentrated on re-balancing U-shaped and straight single-model assembly lines with deterministic task times while considering task precedence constraints. In the considered problem, the initial line balance was broken because of cycle time fluctuations. The author aimed to find a fixed task sequence that enabled assembly line balance with a minimum number of workstations (NS). A heuristic method called REB was proposed, which consists of two stages. In the first stage of REB, a balancing module was run, and then in the second stage, a COMSOAL heuristic was used to regenerate the task precedence diagram. Several well-known test problems from ALBP literature (Scholl & Klein, 1997) were solved to test the method's efficiency. As no method can achieve optimal results, the comparisons were made between U-type and straight lines. The provided method found a fixed task sequence in which optimal NS for different cycle times could be achieved. It was illustrated that good line balances were attainable for different cycle times with no need to relocate the equipment.

Unlike other U-shaped studies, Celik et al. (2014) studied ALRBP with stochastic task times. To reduce the overall cost of rebalancing, the authors used an ant colony optimization (ACO) algorithm. The objectives were to minimize the total cost of rebalancing, which consists of task transportation costs, workstation opening/closing costs, and workstation operating costs while considering the probability of incompletion cost constraints. Changes in product demand and cycle time fluctuations made the line unbalanced. To evaluate the proposed ACO, researchers used relationships and deterministic task times from 11 well-known line balancing

problems taken from Talbot et al. (1986) and used them to generate rebalancing problems. COMSOAL heuristic developed by Arcus (1966) has been used for generating initial U-line balances. The transportation cost of half of the tasks was assumed to be zero, and the remaining half were generated between $100 and $2000 randomly. Workstation opening/closing costs were selected as $500 and $250, respectively. Workstation operating cost was assumed to be $2000/month. The algorithm was implemented by Visual Basic 6.0 and tested with 330 test problems. As there were no previous U-line ALRBP solutions proposed in the literature, researchers could not compare the results with other methods but shared the results for all test problems, which were solved for cases of increased demand and decreased demand.

Serin et al. (2019) proposed a GA for solving U-type ALRBP with stochastic task times. They used the same problem instances, objectives, and constraints with their previous study (Celik et al., 2014), in which they proposed an ACO algorithm. They compared the results of the proposed GA with the ACO algorithm and got better results for total rebalancing cost, the number of workstations, and computational times.

Sikora et al. (2017) studied single model ALRBP type-2 on U-shaped and straight lines with deterministic task times together with worker assignment. In this study, task time changes caused rebalancing. While researchers were trying to minimize the cycle time of the AL, they both tried to minimize total movement time and solved traveling worker ALBP which allowed a worker to perform tasks in more than one workstation. A mixed-integer linear programming method was proposed, and combinatorial optimization was used for pre-processing. Task zoning, capacity, human and robotic workers assignment restrictions (equipment & worker ability), automatic tasks (worker initialize, robot do), common tasks (more than one worker is required), different processing time per workstation, and fixed time per workstation (set-up times) were considered as additional constraints. Finally, the model was tested with adapted benchmark datasets of Scholl (1999), and three different case studies were solved.

Zha & Yu (2014) presented a hybrid ant colony optimization (ACO) and filtered beam search (FBS) for solving single-model U-line balancing and rebalancing problems with deterministic task times. In their study, researchers considered multiple objectives, which were minimizing the number of workstations, minimizing the moving cost of machines, minimizing labor cost, and minimizing the walking time of operators while not violating task precedence and worker walking time constraints. The need for rebalancing was triggered when product demand varied. For testing the performance of the proposed algorithm on the balancing problem, the U-line balancing benchmark problems ignoring walking time in the literature were used. FBS-ACO was compared with RPW (rank position weight algorithm), ACO proposed by Baykasoğlu & Dereli (2009), ULINO proposed by Scholl & Klein (1999), and LB (theoretical lower bound). In 25 test problems, it is stated that better solutions were found with a lesser number of workstations than RPW in 8 problems, ACO in 11 problems, and ULINO in 3 problems. FBS-ACO was also demonstrated with an example and yielded optimal solutions.

A two-sided assembly line is another widely used line type where there are two stations on two sides of a conveyor belt. The number of studies on two-sided ALRBP is limited (Grangeon et al., 2011; Liu et al., 2012; Y. Zhang et al., 2018a, 2018c, 2020).

Liu et al. (2012) studied the rebalancing problem focused on two-sided assembly lines. The authors studied the problem of how to reconfigure the conveyor assembly line to serus. Seru is a Japanese word for cell and usually refers to an assembly cell. The authors gave a mathematical model that addressed two issues: the number of serus that should be established and the number of workers that should be assigned to each seru. They stated that the computational results were promising.

Y. Zhang et al. (2018a) modified non-dominated sorting GA for solving a real-life multi-objective ALRBP of type-2 on two-sided lines. The objectives of their problem were minimizing cycle time while simultaneously minimizing rebalancing cost and minimizing similarity. Production demand modifications, changes in the line's structure, and production processes were considered as causes for re-balancing. New

cycle time, idle time, and balance constraints were considered by the authors as additional constraints. A real-life case study from a Chinese construction machinery manufacturing firm was solved for testing the proposed method.

Y. Zhang et al. (2018b) proposed a workstation-oriented heuristic for solving single-model two-sided ALRBP type-2 with deterministic task times. They took cycle time fluctuations as a rebalancing trigger. The objective was minimizing cycle time and rebalancing costs. Instead of calculating the rebalancing cost, the authors considered minimizing task movements. Task precedence relations and workstation idle time were taken as additional constraints. An example was solved by using a 24-task problem of Lee et al. (2001). An initial assignment that is not optimal was used from the study, and their proposed method also reached the best-known cycle time. They have calculated total task movements as rebalancing costs.

Several product versions are manufactured simultaneously in some instances, which is called mixed-model (Mardani-Fard et al., 2020) ALRBP. Altemeier et al. (2010) defined a mathematical optimization method for solving mixed model ALRBP in the context of auto production. They considered; task precedence restrictions, workplace limits (average workload on a workstation), permitted workplaces (each process receives a list of allowed workstations), and fixed processes (processes that could not be moved to other workstations) as validity constraints. Researchers tried to minimize production costs by reassigning and shifting tasks between workstations. The reconfiguration costs which were consisted of retraining of workers, shifting tools, and parts were summed with the operating cost for regular workers and floaters (special and more expensive workers) to calculate the total production cost. It was indicated that exact or heuristic solution methods could be developed to solve this model however in a real-world manufacturing line; it is difficult to obtain the big amount of data that the model requires and there are numerous additional practical restrictions, so no model-based approach has been implemented as a practical solution in the industry. Therefore, researchers proposed a concept of decision support procedure based on the model, and the planner was integrated into the decision process with his knowledge. The procedure contains four steps: analysis, design, selection, and

implementation. The manual reconfiguration part of the decision support approach was complemented by a local search heuristic. The proposed approach was evaluated on two case studies from a major car manufacturer. The results were compared with the current solution which was made without a decision support system. The proposed method got better results than the manual decision and drastically reduced the occurrence of overload situations.

Grangeon et al. (2011) proposed three heuristics for solving mixed-mode ALRBP type-1 on straight lines with deterministic task times where monthly product demand changes caused the need for rebalancing. Researchers tried to minimize task relocation and workload smoothing while considering operation precedence, capacity (operator time), working height, and storage area length constraints. They divided lines into sections. Each operation had a working height, and only a working height was allowed at a section. The storage length was required on the section to store the necessary parts and tools used for the operation. The proposed heuristics were used for the transfer of the operations. The first heuristic, feasible load-based (FLB), tried to transform the initial unfeasible balancing into a feasible one. The second one, maximum load-based (MLB), tried to improve a feasible solution by reducing the number of workstations. The third one, Smoothed load-based (SLB), tried to improve a feasible solution by smoothing the workload of the workstations. They used this method to solve a real-life industrial instance and compare the best results obtained by the preliminary work of Boutevin (as quoted in Grangeon et al. (2011)). Researchers have developed software with a graphical interface to manage data, make necessary modifications, and view the results of the solution methods.

Ishigaki (2018) addressed a mixed-model straight ALRBP with deterministic processing times which the processing time depended on the item and workstation. After demand fluctuation, a need for line rebalancing occurred. A heuristic method was proposed to minimize the line stopping time. The proposed heuristic was used for deciding task assignments and item processing sequences. The author used numerical experiments with a benchmark to validate the proposed method.

Several studies dealt with AL balancing and worker assignment. Even though they did not consider the rebalancing process, they gave initiation for rebalancing and worker assignment simultaneously (Borba & Ritt, 2014; Chaves et al., 2007; Costa & Miralles, 2009; Guo et al., 2008; Miralles et al., 2008; Moreira et al., 2015; Zacharia & Nearchou, 2016). However, published literature on ALRBPs often only focuses on the rebalancing of the assembly line. There is limited work that considers worker assignment together with the rebalancing problem (Corominas et al., 2008; Girit & Azizoğlu, 2021; Karas & Ozcelik, 2021; Liu et al., 2012; Sikora et al., 2017; Yang et al., 2013; Yang & Gao, 2016). Corominas et al. (2008) solved ALRBP in a motorcycle plant with a linear programming procedure while minimizing the number of temporary workers. Temporary workers needed more time to implement the tasks and could not perform some of them.

Yang et al. (2013) solved mixed-model ALRBP with a multi-objective genetic algorithm while minimizing workload smoothness, rebalancing cost, and the number of stations. Yang & Gao (2016) offered a branch, bound, and remember algorithm for solving mixed-model ALRBP of type-1. A model for adjacent workforce cross-training policy assembly lines was presented, in which two employees in neighboring stations could learn the skills of each other. When there was a change in product demand or product mix, the tasks could only be shifted to adjacent stations to rebalance the line, where there was a worker who had previously learned the skills to process the task. The objective was minimizing the number of workstations under task precedence and cycle time constraints where task times were deterministic. The proposed method was tested on 450 problems which were generated on 25 well-known benchmark problems, and researchers could find optimal solutions for 388 instances in under 100 seconds. According to the results, it was suggested that using the cross-training policy could effectively reduce the number of workstations when especially demand variation is significant.

Girit & Azizoğlu (2021) rebalanced AL with two objectives: i) maximizing workload balance and ii) minimizing total replacement distance. They offered a

fairness index to penalize higher workloads and smooth the work among stations. However, they assumed that all workers were identical.

Otto & Scholl (2011) solved ALRBP together with ergonomic risk assessment. They proposed a two-stage heuristic approach based on the exact balancing procedure SALOME and simulated annealing after ergonomic problems caused an imbalance in the AL. The objectives are to minimize the number of workstations while minimizing ergonomic risk measures, which makes it the only study in the ALRBP literature that considers ergonomics in the objective function. In their study, cycle time, the number of workstations, and task precedence relations are considered as problem constraints while rebalancing single-model straight assembly lines with deterministic task times. They used the well-known benchmark dataset (Scholl, 1999) for ALBP and found good and stable results in short computational times.

Oliveira et al. (2012) focused on solving mixed model ALRBP type-1 with deterministic task times on straight assembly lines. Researchers tried to balance workstation times while trying to minimize the number of workstations. The initial balance was broken with changes in product demand, and AL was tried to be rebalanced without violating task precedence and positive/negative task zoning constraints. ALBP was solved by the formulation of Gökçen & Erel (1998), and a binary integer programming model by branch and bound (BIP-BB) methodology was proposed to optimize that model. For model validation, researchers compared the proposed method with the heuristic method that was suggested by Slack et al. (1995), and these methods were evaluated based on performance indicators such as the number of workstations created, average workload, and level of unbalance. The methodology was illustrated through a real-life case study in the automotive industry. The solution method of Gökçen & Erel (1998) got better results than the method of Slack et al. (1995), while it also improved the performance of industrial AL by reducing the number of workstations, improving the imbalance rate, and reducing the costs.

Antoine et al. (2014) proposed a method that is based on the reachability analysis of an automata network for solving mixed-model ALRBP on manual straight lines

with deterministic and dynamic task times. They introduced online dynamic rebalancing for a short time horizon when disturbances occurred without violating task precedence relationships of the tasks. To keep the lines balanced, researchers proposed task time modifications. Their aim was quickly reacting to situations and minimizing delays even though new balancing is not optimal. It was stated that real-time information must be available from an intelligent manufacturing system to initiate this problem. A case study was solved, and the results were described to visualize the proposed method.

Lai et al. (2015) proposed an improved dual-population GA for solving mixed-model AL with deterministic task times. They tried to minimize the number of workstations, rebalancing costs, in-workstation, and between-workstation load indexes, and maximize task-workstation assignment similarity with the initial solution while considering task precedence and task-workstation allocation restrictions. An industrial case-study was solved to evaluate the method's performance after initial AL became imbalanced due to changes in product demand, adding-removing tasks, changes in task times, and product design. They reduced adjustment costs while other indicators were essentially unchanged, which generated economic benefits for enterprises.

Antoine et al. (2016) studied single-model dynamic ALRBP with deterministic task times. Researchers formalized the problem as an integer linear programming model and solved it with an iterated local search (ILS) heuristic approach. The objectives were minimizing the cycle time and generating solutions fast enough to adapt while considering task precedence relations. An industrial case study from the Trane Company was used for validating the ILS method. It is stated that the results have confirmed the effectiveness of the ILS approach, but real-life constraints and the size of the problem might lead to many issues for the proposed method.

Samadhi & Sumihartati (2016) suggested an integer programming model for solving single-model ALRBP type-1 on a straight line with deterministic task times in the garment industry. In this study, product demand changes were considered as the

cause of rebalancing. The objective of the model is to minimize the total cost of the line, which consists of operating costs, machine cost, adding capacity cost, losses cost due to idle capacity, and outsourcing costs. Cycle time, task precedence, available space, and outsourcing company capacity constraints are considered in the problem. Researchers solved an industrial case in a company and determined a balance between outsourcing and the addition of new sewing machines to the line to meet the shortage of production capacity after demand increased with minimum total cost.

D. Z. Zhang et al. (2016) tried to minimize the cycle time while maximizing workload smoothness in mixed-model two-sided lines with deterministic task times. The proposed heuristic method considers the number of workstations, task precedence, and a new constraint between tasks that are introduced by the authors, called *incompatible task groups (ITG)*. ITG concept limits performing the tasks existing in the same ITG group on either side of the line (D. Z. Zhang et al., 2016), which is different from negative zoning constraints. An industrial case study from an automotive company was solved, and after rebalancing the current line, cycle time was reduced by 22.4%.

Y. Zhang et al. (2018c) solved the same problem they published in 2018 (Y. Zhang et al., 2018b). In this study, they proposed a heuristic method based on three heuristic rules which they have named as the push rule-based heuristic (PRH) method. They used the same problem set to validate their results and got worse solutions compared to Y. Zhang et al. (2018b).

Note that none of the above-mentioned work considered workers as a unique resource. Although they considered the workforce while rebalancing, none of them considered varying operation times of tasks due to workers. Since the motivation of the worker assignment problem introduced by Miralles et al. (2008) is to address every worker as a unique resource, in this study, each task has a specific processing time depending on the worker. In the AL literature, labor uniqueness is typically underrated. However, the levels of skill, ability, experience, and physical conditions of workers directly affect the operation times of tasks. Limited work considers the importance of

unique worker skills in the manufacturing environment (Fichera et al., 2017; Tóth & Kulcsár, 2021). Fichera et al. (2017) tackled the worker assignment problem in a cellular manufacturing plant by considering the different ability levels of workers and the learning effect. Tóth & Kulcsár (2021) studied worker-dependent processing times in flexible manufacturing systems. The authors proposed a solution procedure by using simulation and demonstrated the effect of heterogeneous workers. Although these papers pay regard to unique workers and varying operation times of tasks, they study manufacturing environments rather than ALs. It is understood from the literature survey that there is a gap in the relevant literature that none of the rebalancing papers considers varying operation times of tasks depending on unique workers, which is one of the essential contributions of this paper.

Moreover, note that none of the above-mentioned research studies consider ergonomic risk factors of the working environment. The limited number of ALRBP studies with worker assignments mentioned above do not consider ergonomic issues as a constraint or in the objective function. Ergonomic aspects are becoming increasingly crucial in real-life scenarios, especially in manual assembly lines. Manuel assembly lines cover a major part of manufacturing in Turkey. So, this is the primary motivation of this research. This research study is a pioneer because while tackling ergonomic risks of the manufacturing environment, the presented method reassigns workers to stations and reassigns tasks to workers simultaneously. The motivation of this work is to fill the gap in the relevant literature.

CHAPTER 3
ASSEMBLY LINE REBALANCING AND WORKER ASSIGNMENT PROBLEM CONSIDERING ERGONOMIC RISK FACTORS (ERGOALWARBP)

As stated in the previous section, ALRBPs have barely been studied compared to ALBPs. Also, most of the rebalancing studies accept that all workers are identical in the relevant literature. However, this assumption is not realistic because workers have unique characteristics (Miralles et al., 2008). Especially in manual ALs, workers must be considered unique to represent the real-life manufacturing environment better. Moreover, it is essential to assign tasks as equally as possible among workers when rebalancing an existing line. Such assignment enables smoothening the strain levels of workers among stations (Otto & Scholl, 2011).

For ergonomic risk assessment in ALs, many methods have been used, such as Occupational Repetitive Action (OCRA)(Occhipinti, 1998), The European Assembly Worksheet (EAWS) (Schaub et al., 2013), the National Institute for Occupational Safety and Health (NIOSH) (Waters, 1994), Quick Exposure Check (QEC)(G. Li & Buckle, 1999), Rapid Upper Limb Assessment (RULA)(McAtamney & Nigel Corlett, 1993), and Rapid Entire Body Assessment (REBA)(Hignett & McAtamney, 2000). The OCRA method is used for estimating risk factors for tasks done by upper limbs repetitively with high frequency. In manual assembly lines, especially in automotive harness production, workers use their upper limbs repetitively at high frequency. Therefore, the OCRA method is used for ergonomic risk calculations in this study. Further details about ergonomic risk assessment and the OCRA method are described by Akyol & Baykasoğlu (2019a).

3.1 Assumptions

The following assumptions are taken into account for the problem:

- It is considered that: the line was previously balanced before the disruptive event occurred.

- Each product model's operating times and precedence diagrams are given.
- A station can perform any operation, and a task can be allocated to one station.
- Workers' travel times are neglected.
- The number of workers available for reassignment is greater than the number of workers in the initial assignment to fit the worker needs in case of opening additional stations while rebalancing the AL.

3.2 Objective Functions

It is benefited from several previous studies in this study while defining the seven objective functions and their formulations. (Celik et al., 2014; Gamberini et al., 2006; Yang et al., 2013; Zacharia & Nearchou, 2016). The objectives are listed below:

$$\boldsymbol{min}.\, TC_{rb} = (cost_{open} * m^{+}) + (cost_{close} * m^{-}) + (cost_{run} * (m - m^{0})) + \sum_{i \in N_{rb}} cost_i \tag{3.1}$$

$$m^{+} = m - m^{0}, \quad m^{-} = 0;\ if\ m > m^{0} \tag{3.1.1}$$

$$m^{-} = m^{0} - m, \quad m^{+} = 0;\ if\ m < m^{0} \tag{3.1.2}$$

$$\boldsymbol{max}.\, MSF = \left(\sum_{i=1}^{n} SF_i\right)/n \tag{3.2}$$

$$SF_i = \frac{Cardinality(TIB_i \cap TNB_i)}{Cardinality(TIB_i)} \tag{3.2.1}$$

$$TIB_i = \{\, j \in NS_a^0 \mid i \in NS_a^0 \ and\ j \neq i \,\}, \quad for\ i = 1, \dots, n^0 \tag{3.2.2}$$

$$TNB_i = \{\, j \in NS_a \mid i \in NS_a \ and\ j \neq i \,\}, \quad for\ i = 1, \dots, n \tag{3.2.3}$$

$$\boldsymbol{max}.\, WorkerMSF = \left(\sum_{w \in W} WorkerSF_w\right)/m \tag{3.3}$$

$$WorkerSF_w = \frac{Cardinality(N_w^0 \cap N_w)}{Cardinality(N_w^0)}, \quad \forall w \in W \tag{3.3.1}$$

$$\boldsymbol{min}.\, madOCRA = \left(\sum_{a=1}^{m} |avgOCRA - OCRA_a|\right)/m \tag{3.4}$$

$$avgOCRA = \left(\sum_{a=1}^{m} OCRA_a\right)/m \tag{3.4.1}$$

$$\boldsymbol{min}.\, n_{rb} = \sum_{a=1}^{m} \left(Cardinality(NS_a^0) - Cardinality(NS_a^0 \cap NS_a)\right) \tag{3.5}$$

$$\boldsymbol{max}.\, LE = \left(\sum_{a=1}^{m} Tt_a\right) \Big/ (m * c) \tag{3.6}$$

$$\boldsymbol{min}.\, SX = \sqrt{\sum_{a=1}^{m} (c - Tt_a)^2} \tag{3.7}$$

As Celik et al. (2014) stated, some tasks may be transferred between stations to rebalance the assembly line, and a new line balance may need more or a smaller number of stations. In the current study, the first objective (3.1) is minimizing total rebalancing costs (TC_{rb}) which is calculated as stated in equations (3.1), (3.1.1), and (3.1.2) (Celik et al., 2014).

TC_{rb} is calculated by summing up station opening/closing/operating and task transportation costs.

$cost_i$ arise if certain tasks are assigned from their current positions to new positions (Celik et al., 2014).

$cost_{open}$ happens due to the recruiting of new workers and the procurement of new equipment as the number of stations rises in the new line balance (3.1.1).

$cost_{close}$ occurs because of equipment disposal and the dismissal of obsolete staff when the number of stations in the new line balance decreases (3.1.2).

$cost_{run}$ is calculated by dividing the sum of operating expenses such as rent, equipment, labor, and inventory costs by a period (days, months, production hours). Therefore, the value changes (the duration that the new line balance will operate. If $m^+ > 0$, then the total operating cost is positive, and if $m^- > 0$, the total operating cost is negative.

Gamberini et al. (2006, 2009) presented an objective called mean similarity factor (MSF), which is the similarity between the rebalanced AL and the initial AL. The second objective (3.2) in the current study is to minimize the MSF. To evaluate MSF, TIB_i, and TNB_i are introduced in equations (3.2.2) and (3.2.3), respectively. SF_i, which is calculated using equation (3.2.1), that is obtained by dividing the number of tasks assigned to the same station as $task\ i$ in the initial and new AL, by to the number of tasks assigned to the same station in the initial AL.

The third objective (3.3) is stated for the first time in the literature to maximize the similarity of the tasks performed by workers between the initial and new AL (*WorkerMSF*), which evaluates the degree of similarity between the initial and the new task assignment of each worker. The similarity factor of the generic $worker\ w$ ($WorkerSF_W$), which is calculated using equation (3.3.1), is obtained by dividing the number of tasks assigned to $worker\ w$ both in the initial and in the new AL by the number of tasks assigned to the $worker\ w$ in the initial AL.

For example, let us assume that $worker\ 1$ is performing the tasks {1, 2, 3, 4} in $station\ 1$ in the initial AL. After the rebalancing, $worker\ 1$ is assigned to station 2 and starts performing the tasks {3, 4, 5, 6}. The tasks {3, 4} are performed by $worker\ 1$ in both the initial and the new AL; therefore, $WorkerSF_1$ is equal to *Cardinality(3,4) / Cardinality(1,2,3,4) = 2 / 4 = 0,5.* After applying this procedure to all assigned workers, $WorkerMSF$ value is calculated by summing up the $WorkerSF$ values divided by the number of stations in the rebalanced AL as stated in equation (3.3).

In this study, the ergonomic risk levels of stations are determined by using OCRA (Occhipinti, 1998) ergonomic risk assessment method (for more information, see Appendix; Akyol & Baykasoğlu (2019a)). The fourth objective (3.4) is minimizing the mean absolute deviation of stations' OCRA index values (*madOCRA*), which was offered to control and monitor the ergonomic risk level of the stations and to smooth the workload between workers in terms of ergonomic risks. To calculate *madOCRA*, the average of OCRA index values (*avgOCRA*) is needed, which is evaluated using equation (3.4.1).

Task transportation cost is an essential part of the total rebalancing cost. Therefore, as our fifth objective (3.5), we also try to minimize the number of tasks that are transported for rebalancing, n_{rb}. Although two solutions can have the same MSF value, they can have a different number of reassigned tasks.

The sixth objective (3.6) is maximizing *LE*, and the seventh objective (3.7) is minimizing the *SX* (Zacharia & Nearchou, 2016), which is a measure to distribute the work equally between the stations.

3.3 Constraints

Constraints of the mathematical model for the ALRBP with worker assignment can be stated as follows (Gamberini et al., 2006; Miralles et al., 2008):

$$\sum_{\mathrm{w}\in W}\sum_{\mathrm{a}\in S} x_{iaw} = 1, \qquad \forall i \in N, \tag{3.8}$$

$$\sum_{\mathrm{a}\in S} y_{aw} \leq 1, \qquad \forall \mathrm{w} \in W, \tag{3.9}$$

$$\sum_{\mathrm{w}\in W} y_{aw} \leq 1, \qquad \forall a \in S, \tag{3.10}$$

$$\sum_{w \in W} \sum_{a \in S} (s * x_{iaw}) \leq \sum_{w \in W} \sum_{a \in S} \left(s * x_{jaw}\right), \quad \forall i, j / i \in IP_j, \tag{3.11}$$

$$\sum_{i \in N} \left(t_{iw} * x_{iaw}\right) \leq c, \qquad \forall w \in W, \qquad \forall a \in S, \tag{3.12}$$

$$\sum_{i \in N} x_{iaw} \leq (m * y_{aw}), \qquad \forall w \in W, \qquad \forall a \in S, \tag{3.13}$$

with

$$y_{aw} \in \{0, 1\}, \qquad \forall a \in S, \qquad w \in W,$$

$$x_{iaw}, x_{jaw} \in \{0, 1\}, \qquad \forall a \in S, \qquad w \in W, \qquad i, j \in N$$

Constraint (3.8) ensures that each $task\ i$ is assigned to a single $worker\ w$ and a single $station\ a$. Constraint sets (3.9) and (3.10) express that a worker can work at only one station, and there must only be one assigned worker in each station, respectively. The precedence relationships between task *i* and task *j* are specified in the constraints set (3.11), in which task *i* is the predecessor of task *j*. Constraint sets (3.12) and (3.13) guarantee that when cycle time *c* is not reached, any worker *w* assigned to the station *a* can process more than one task.

Nevertheless, the proposed problem is an extension of the ALRBP with worker assignment because it includes ergonomic risk assessment. So, the OCRA index value ($OCRA_a$) calculations can be included in the model implicitly. Equation (3.14) demonstrates that risk analysis for all the stations is smoothed in accordance with the average OCRA index value ($avgOCRA$).

$$OCRA_a + d_i^+ - d_i^- = avgOCRA \tag{3.14}$$

Note that the above constraints are given to define the proposed problem formally. Because of the complexity, the novel ErgoALWARBP cannot be solved by exact solution methods. A randomized constructive rule-based heuristic approach is developed to tackle the problem.

CHAPTER 4
THE PROPOSED ALGORITHM FOR ERGOALWARBP

In this study, task and worker selection rules which were described by Akyol and Baykasoğlu (2019b), are applied to pick appropriate tasks for workers. A randomized constructive rule-based heuristic approach is developed to find a feasible solution. Firstly, 39 tasks and four worker priority rules are set. Then, the heuristic prioritizes the tasks, and a task is chosen via roulette wheel selection. The heuristic reassigns tasks sequentially to the stations. Later, after applying worker selection rules, a worker among prioritized workers is selected using roulette wheel selection and reassigned to that station. When cycle time is exceeded, a new station is opened. In this way, the developed heuristic constructs a new solution step by step and includes rule-based prioritized randomness (for more information, see Akyol & Baykasoğlu, (2019b)). After obtaining a newly rebalanced AL, the new line is compared with the old one regarding the MSF of tasks and workers.

4.1 The Rule-Based Task Selection Algorithm

In this study, 39 task priority rules (Akyol & Baykasoğlu, 2019b) are applied, and assignable tasks are prioritized by considering different task execution times depending on the operators. First, on lines 5 to 7, one point is added to each task that can be assigned to the station. Then, on lines 9 to 11, each task selection rule is called one by one. The rules loop list of assignable tasks and add point(s) to the appropriate one(s). These task selection rules are detailly described in Akyol & Baykasoğlu (2019b). Then, a task is chosen among assignable tasks such that the task with the higher point has a higher probability of selection on line 12. In the literature, this method is called *roulette wheel selection*. On line 13, the selected task by roulette wheel selection is returned. If there is only one assignable task to the station, the rules are not applied, and the task is directly returned on line 15. The overall procedure of the rule-based task selection method is outlined in Figure 4.1 as follows:

```
1   METHOD: Rule-based task selection for the station
2   INPUT: List of assignable tasks, workers with task processing times
3   OUTPUT: Selected task
4
5   FOR each task in the station's assignable tasks list DO
6     Add 1 point to the task
7   END FOR
8   IF station's assignable tasks count > 1 THEN
9     FOR each rule in 39 task rules DO
10      CALL rule //add point(s) to the appropriate task(s)
11    END FOR
12    SelectedTask ← CALL Task_Roulette_Wheel_Selection (list of assignable tasks)
13    RETURN SelectedTask
14  ELSE IF assignable tasks count = 1 THEN
15    RETURN the only assignable task
16  END IF
```

Figure 4.1 Pseudo code of the rule-based task selection algorithm

4.2 The Rule-Based Worker Selection Algorithm

In this study, four rules for the selection of workers are applied (Akyol & Baykasoğlu, 2019b):

- RND - Random priority
- GNTE - Greatest no. of tasks executed
- GNTEMT - Greatest no. of tasks executed in minimum time
- MU - Maximum utilization

First, on lines 5 to 7, one point is added to each worker that can be assigned to the station. If there is more than one assignable worker, then these four rules are called on lines 9 to 16. Each rule adds two points to the appropriate worker. By utilizing the above four rules, assignable workers are prioritized. Then, by applying the *roulette wheel selection* described in section 4.1, a worker is chosen among assignable workers. On line 18, the selected worker by roulette wheel selection is returned. If there is only one assignable worker to the station, the rules are not applied, and the worker is directly

returned on line 20. The rule-based worker selection method is outlined in Figure 4.2 as follows:

```
1   METHOD: Rule-based worker selection for the station
2   INPUT:  Station's assigned tasks, list of assignable workers with task processing times
3   OUTPUT: Selected worker for the station
4
5   FOR each worker in the list of assignable workers DO
6     Add 1 point to the worker
7   END FOR
8   IF assignable workers count > 1 THEN
9     //rule-1 (RND)
10    Add 2 points to the randomly selected worker from the assignable workers list
11    //rule-2 (GNTE)
12    Add 2 points to the worker who is capable of processing max. number of tasks
13    //rule-3 (GNTEMT)
14    Add 2 points to the worker who can process max. number of tasks in min. time
15    //rule-4 (MU)
16    Add 2 points to the worker who can execute this station's tasks fastest
17    SelectedWorker ← CALL Worker_Roulette_Wheel_Selection (list of assignable
18  workers)
19    RETURN SelectedWorker
20  ELSEIF assignable workers count = 1
21    RETURN the only assignable worker
22  END IF
```

Figure 4.2 Pseudo code of the rule-based worker selection algorithm

4.3 The Developed Preemptive Goal Programming Algorithm for Fitness Evaluation

There are several ways to evaluate solutions, such as comparing cycle time (Karas & Ozcelik, 2021; Y. Li, 2017; Liu et al., 2012; Sancı & Azizoğlu, 2017; Sikora et al., 2017; Y. Zhang et al., 2018a, 2018c, 2020), total cost (Lai et al., 2015; Serin et al., 2019; Zha & Yu, 2014), line efficiency (Altemeier et al., 2010; Belassiria et al., 2018; Rahman, 2010), workload balance (Grangeon et al., 2011; Lai et al., 2015; Mokhtari & Mozdgir, 2015; Oliveira et al., 2012) or similarity index (Zacharia & Nearchou,

2016). We made preferences among the seven objectives and put them in an initial order (Goal 1), as in Figure 4.3. The preemptive goal programming algorithm for Goal 1 is outlined in Figure 4.3, that the highest priority is finding the solution with the minimum total rebalancing cost. Firstly, we initialize the *BestSolution* variable on line 5. For the objectives that are trying to be minimized, the properties in the *BestSolution* are initialized with a very high number H. Likewise, for the objectives that are trying to be maximized, the properties in the *BestSolution* are initialized with a very low number L. By this way, the first feasible solution that is compared with the *BestSolution* is always better and is assigned to the *BestSolution*. By looping all feasible solutions on line 6, we compare solution's total cost, MSF, MSF Worker, mean absolute deviation of OCRA index values, line efficiency, and line smoothness index values with the *BestSolution*'s appropriate values respectively on lines 7 to 33. When the compared property of the solution is better than *BestSolution*'s value, the solution is assigned to the BestSolution. The calculation details of the relevant objectives were shared in section 3.3.

```
1   METHOD: Return best solution with goal programming
2   INPUT:  Feasible solutions, Goal's objective order
3   OUTPUT: The best solution
4
5   INIT BestSolution
6   FOR each solution in feasible solutions DO
7     //obj.1 Total cost
8     IF solution's Total Cost < BestSolution's Total Cost THEN
9       BestSolution ← solution
10    ELSE IF solution's Total Cost = BestSolution's Total Cost THEN
11      //obj.2 Mean similarity factor (MSF)
12      IF solution's MSF > BestSolution's MSF THEN
13        BestSolution ← solution
14      ELSE IF solution's MSF = BestSolution's MSF THEN
15        //obj.3 MSF Worker
16        IF solution's WorkerMSF > BestSolution's WorkerMSF THEN
17          BestSolution ← solution
18        ELSE IF solution's WorkerMSF = BestSolution's WorkerMSF THEN
19          //obj.4 Mean absolute deviation of OCRA index values
```

```
20          IF solution's madOCRA < BestSolution's madOCRA THEN
21            BestSolution ← solution
22          ELSE IF solution's madOCRA = BestSolution's madOCRA THEN
23            //obj.5 Re-assigned tasks count
24            IF solution's MovedTasksCount < BestSolution's MovedTasksCount THEN
25              BestSolution ← solution
26            ELSE IF solution's MovedTasksCount = BestSolution's MovedTasksCount THEN
27              //obj.6 Line efficiency
28              IF solution's LE > BestSolution's LE THEN
29                BestSolution ← solution
30              ELSE IF solution's LE = BestSolution's LE THEN
31                //obj.7 Line smoothness index
32                IF solution's SX < BestSolution's SX THEN
33                  BestSolution ← solution
34                END IF //obj.7
35              END IF //obj.6
36            END IF //obj.5
37          END IF //obj.4
38        END IF //obj.3
39      END IF //obj.2
40    END IF //obj.1
41  END FOR
42  RETURN BestSolution
```

Figure 4.3 Pseudo code of the preemptive goal programming algorithm (for Goal 1)

The priority of the objectives may change according to current conditions. To fit those varying needs, the initial order of the objectives, which is defined in Figure 4.3, was changed using insert, remove, and shift operations. In this way, six more objective orders (goals) were generated inside our solution calculation method, as outlined in Figure 4.4.

```
1   METHOD: Calculate goal results
2   INPUT:  Initial order of objective functions (list of obj. functions)
3   OUTPUT: The best solution for each goal (list of best solutions)
4
5   BestSolution ← CALL Algorithm_3 (Initial order of obj. functions, list of feasible solutions)
6   Insert BestSolution into the list of best solutions
7
8   FOR each objective in the list of obj. functions DO
9     firstObjective ← objective
10    Remove objective from the list of obj. functions
11    Shift the objectives before the removed objective to right
12    Insert firstObjective to the first position of the list of obj. functions
13
14    BestSolution ← CALL Algorithm_3(list of obj. functions, list of feasible solutions)
15    Insert BestSolution into the list of best solutions
16  END FOR
17  RETURN the list of best solutions
```

Figure 4.4 Pseudo code of Calculating the best solution for all goals (changing order of objective functions)

The objective order of the seven goals is listed below:

G1–min. TC_{rb}: $TC_{rb} > MSF > WorkerMSF > madOCRA > n_{rb} > LE > SX$

G2–max. MSF: $MSF > TC_{rb} > WorkerMSF > madOCRA > n_{rb} > LE > SX$

G3–max. WorkerMSF: $WorkerMSF > TC_{rb} > MSF > madOCRA > n_{rb} > LE > SX$

G4–min. madOCRA: $madOCRA > TC_{rb} > MSF > WorkerMSF > n_{rb} > LE > SX$

G5–min. n_{rb}: $n_{rb} > TC_{rb} > MSF > WorkerMSF > madOCRA > LE > SX$

G6–max. LE: $LE > TC_{rb} > MSF > WorkerMSF > madOCRA > n_{rb} > SX$

G7–min. SX: $SX > TC_{rb} > MSF > WorkerMSF > madOCRA > n_{rb} > LE$

The rule-based task and worker selection algorithms were described with an illustrative example in Akyol & Baykasoğlu (2019b). The proposed heuristic solution procedure for ErgoALWARBP is described with the flowchart that is shared in Figure 6.1.

CHAPTER 5
COMPUTATIONAL STUDY

We evaluated the rebalancing performance of our proposed heuristic with the well-known test problems of Scholl (1993, 1999). Our problem set contains 25 test problems in which the number of tasks is between 7 and 297, and cycle times are between 7 and 5853 seconds. In Table 5.1, the benchmark data family of the precedence graph, the number of tasks, the cycle time of the AL before rebalancing (**initial cycle time**), and the minimal number of stations for the new cycle time (**m***) are shown in the first four columns, respectively.

The initial balance is generated using the COMSOAL (Computer Method of Sequencing Operations of Assembly Lines) heuristic that was developed by Arcus (1966). For each test problem, 500 feasible solutions (12500 solutions in total) are calculated using COMSOAL, which randomly assigns tasks to stations considering precedence relations. Then, the solution with minimum cycle time and maximum line efficiency is returned as the initial solution for each problem.

Changes in customer demand led to an increase (or decrease) in the initial cycle time, which caused the rebalancing need. The updated cycle time values used for rebalancing are shared in the "New cycle time" column in Table 5.1. Each data family's new cycle time values are randomly chosen from the benchmark dataset. In this way, the reassignment results can be compared with **m***, which was calculated with exact algorithms and shared in the benchmark dataset. The developed heuristic runs for 1000 iterations for each test problem (25000 iterations in total).

5.1 Computational Study Results

The best and average results found for each test problem are shared in Table 5.1. Although the suggested heuristic solution method calculates results for seven different objectives, the benchmark problems were only solved to minimize the number of stations. Therefore, we can only measure the performance of the results in terms of the

number of stations, but the result set also contains CPU time, smoothness index, line efficiency, MSF, and the number of tasks moved for informational purposes.

A rebalanced solution with a line efficiency that is greater than 99% is found for Barthold, Gunther, Heskiaoff, and Mansoor. 34.5% of the tasks are moved on average to find a rebalanced solution with a minimum number of task moves for each test problem. The proposed heuristic found the minimum number of stations in 84% (21 of 25) of the test problems. For Barthol2, Lutz2, Scholl, and Warnecke, which can be defined as large-size problems with the number of tasks 148, 89, 297, and 58, respectively, a solution with one more station than the optimum solution is found.

In comparison with our 25 test problems from Sholl's dataset (1993, 1999), Belassiria et al. (2018) used and solved nine of the test problems. Both algorithms reached the minimal number of stations **m*** for Hahn, Heskiaoff, Lutz1, and Mukherje, but our heuristic method found a better solution (fewer stations) than the hybrid genetic algorithm of Belassiria et al. (2018) for Barthold, Gunther, Lutz2, Scholl, and Warnecke.

Table 5.1 Benchmark test problems and the results of the computational study

Test Problem					No. of Stations			CPU time (ms)		Smoothness Index		Line Efficiency		MSF		No. of Tasks Moved	
Precedence graph[1]	No. of tasks	initial cycle time (s)	New cycle time (s)	m*	h-GA[2]	Min.	Avg.	Avg.	Min.	Avg.	Min.	Avg.	Max.	Avg.	Max.	Avg.	Min.
Arcus1	83	5408	5824	14	-	14	14	169	110	2584	1413	92.3	95	0.39	0.59	43.8	26
Arcus2	111	6016	5755	27	-	27	28.7	185	127	4524	1680	91.1	96.8	0.38	0.55	75.2	54
Barthold	148	513	564	10	11	10a	11	1303	817	511	3	90	99.9	0.27	0.46	104	70
Barthol2	148	101	95	45	-	46	47.2	792	643	75.6	36	97	97	0.14	0.23	136.3	120
Bowman	8	19	20	5	-	5	5	0.06	0	10.8	6.63	79.5	88	0.38	0.75	2.9	2
Buxey	29	30	27	13	-	13	14.2	6.8	3	20.1	9	84.2	92	0.26	0.72	21.9	8
Gunther	35	49	54	9	10	9a	10	13.4	8	24.6	14	89	99.4	0.28	0.63	20.6	11
Hahn	53	3507	2806	6	6	6	6	52.8	37	1917	1327	83.2	85	0.60	0.92	19.7	2
Heskiaoff	28	256	205	5	5	5	6	5.42	3	138	1	83.3	99	0.4	0.8	16.1	4
Jackson	11	9	7	8	-	8	8.5	0.17	0	5.1	4	73	82	0.71	0.82	6	4
Jaeschke	9	8	10	4	-	4	4.3	0.06	0	2.4	1	88.5	92.5	0.48	0.56	2.2	2
Kilbridge	45	111	92	6	-	6	7	36.6	24	74.3	39	85.3	88	0.35	0.74	25.5	8
Lutz1	32	1768	2020	8	8	8	11.9	8.51	4	1181	459	84.5	91	0.52	0.94	9.9	2
Lutz2	89	21	20	25	27	26a	26.9	196	132	14.7	8	90.3	93	0.3	0.51	65.7	45
Lutz3	89	79	83	21	-	21	22.7	169	116	76.4	31	87.2	94	0.45	0.8	66	52
Mansoor	11	94	62	3	-	3	3.9	0.2	0	30.2	1	79.2	99	0.51	0.91	4.8	2
Mertens	7	6	7	5	-	5	5.2	0.1	0	3.1	3	81.2	82	0.43	0.57	3.7	2
Mitchell	21	18	15	8	-	8	8.8	2.13	1	13.3	6	80	87.5	0.47	0.9	8.8	2
Mukherje	94	301	281	16	16	16	16	627	443	133	90	93.6	94.6	0.22	0.4	67.4	50
Roszieg	25	14	18	8	-	8	8	3.67	2	10.9	5	86	91	0.61	0.96	10.4	2
Sawyer	30	33	30	12	-	12	12.4	8.18	4	17.6	9	87.2	93	0.36	0.71	19.6	8
Scholl	297	1422	1515	46	51	48a	49.2	17150	13456	1495	1207	93.5	95.8	0.22	0.29	236	211
Tonge	70	185	176	21	-	21	22.7	115	86	151	55	91.6	93.5	0.34	0.68	40.2	16
Warnecke	58	86	92	17	20	18a	19.2	90	64	87.3	33	87.7	94	0.31	0.65	37.1	23
Wee-Mag	75	30	29	63	-	63	63.8	101	71	50.5	44	80.4	82	0.18	0.27	69.7	62

[1] Benchmark data family of the precedence graph (Scholl, 1993), [2] Results of the hybrid genetic algorithm of Belassiria et al.(2018), **m***: The minimal number of stations for the new cycle time, [a] The proposed method found a better solution than h-GA (Belassiria et al., 2018)

CHAPTER 6
INDUSTRIAL CASE STUDY

A real-world case study with a sample of actual operational data is implemented to show the effectiveness of the suggested method. The company has 126 locations worldwide, producing several different products for the automotive sector. The industrial case is presented at a wiring harness production plant of the company in Turkey, which manufactures cable networks for the most prominent automotive brands in the world. Several components are required to assemble automotive wiring harnesses, such as wires, connectors, tubes, terminals, tapes, etc. As customer needs differ widely, more than four thousand components are used for harness production in the current plant.

6.1 Operational Problem Information

The manufacturing consists of four main processes: cutting, sub-assembly, main assembly, and electrical testing. In this study, the main assembly part is considered for ALRBP. The car harnesses are divided into smaller parts such as engine bay (EB), floor, roof, doors, and interior panel (IP) to ease the harness's production, logistics, and installation into the car. A harness is assembled in the main assembly area on vertically positioned boards which are specially designed and produced for each harness. These production boards are named *stations,* and AL contains *m* identical stations. The studied assembly line is O-shaped, and stations are moving with a constant speed that finishes its tour in *C* seconds. A worker is assigned to each station and positioned outside of the O-shape line and only works in the assigned station. This is contrary to U-shaped lines, where workers are positioned inside the assembly line and can work in multiple stations. Therefore, the described line characteristics of the O-shaped line make it equal to a straight assembly line.

Cars have optional features (options) such as several sensors, seat heating, sunroof, etc. These options are met by assembling additional parts to the harness, which are called different models of the harness. Sometimes, there are dozens of models of each

harness which makes it impossible to build a separate assembly line for each model because of production space constraints and high assembly line initial building costs. To overcome this problem, the station is designed for the harness model, which has all options available, and all other models of the harness are assembled on the same assembly line. Although the production is multi-model, the manufacturing engineering department (ME) solves (re)balancing problem as it is a single model AL because they solve the problem for the harness model with the highest penetration (the harness model with the highest demand).

The current line balance is broken because of several reasons, such as customer design change requests, material changes, ergonomic and production-related feedback from the production area, and changes in customer product demand. These changes may cause precedence diagram modifications, adding-removing tasks, changes in task processing times, and changes in cycle time. The proposed solution method and the software can rebalance the assembly line for all these changes, but in this study, its efficiency was tested by changing cycle times as it is the most common reason.

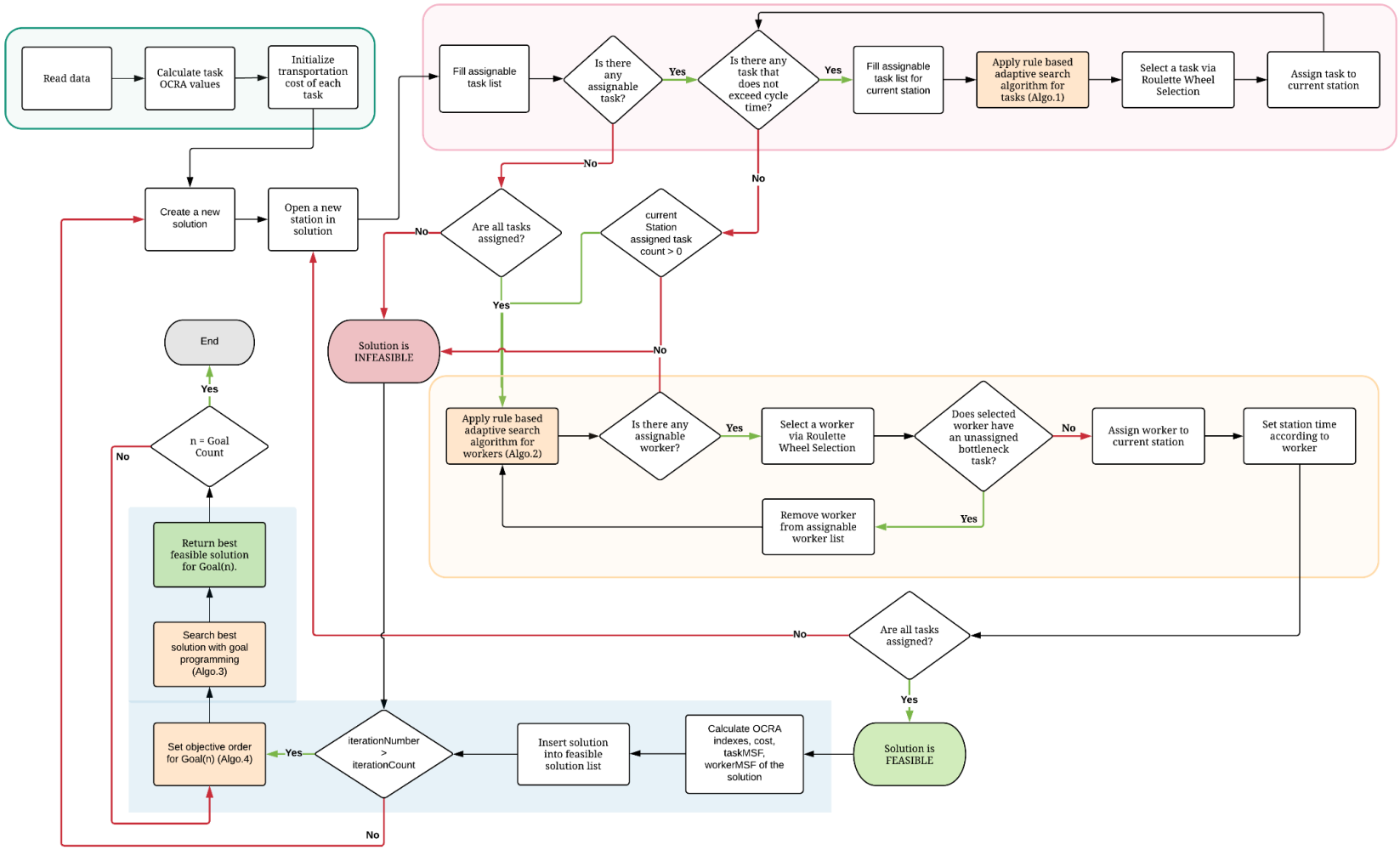


Figure 6.1 Flowchart of the proposed heuristic solution procedure

6.2 Analysis of the Current Assembly Line

The IP harness assembly line was selected for our test case, which was balanced with a cycle time of 170 seconds. The selected model consists of 34 tasks, and the task precedence relationships are shared in Figure 6.2. The diagram became extremely complicated when we showed all the relationships between tasks. To simplify the illustration, we combine the tasks into three groups and only show a relationship line between these groups. The first group contains tasks from 1 to 17. Each task in the first group is the predecessor of all tasks in the second group, which contains tasks from 18 to 29. Likewise, each task in the second group is the predecessor of all tasks in the third group, which contains tasks from 30 to 34. The task precedence relationship of task 1 is shared in Figure 6.3 as an example. Task 1 is the predecessor of tasks 18 to 29. Tasks 30 to 34 are the successors of tasks 18 to 29.

Nine workers are capable of working in the IP assembly line. The task processing times, shared in Table 6.1, differ from worker to worker. As some tasks need specific worker skills or training, they cannot be performed by some of the workers, which is stated with a dash in Table 6.1. The initial assignment of the assembly line is shared in Table 6.2.

Transporting tasks between stations may cause station modifications. This expense differs widely according to the line and task specifications. Task transportation cost is negligible for eight of the tasks, which are set to 0 EUR. For the remaining tasks, task transportation costs range from 134 EUR to 1889 EUR, which are shared in Table 6.1. The station operating cost was calculated as 2000 EUR for the current operation plan for eight weeks, including worker salary. Opening a new station needs ordering new equipment and hiring additional workers, which costs 3000 EUR, and closing a station costs 500 EUR.

The customer increased the demand for the harness, and the cycle time is needed to be decreased from 170 seconds to 158 seconds to meet the new demand. There are seven objectives, which are described in section 3.3. The constraints of the problem

are described in section 3.4. The software is set to stop after 500 iterations and run 10 times.

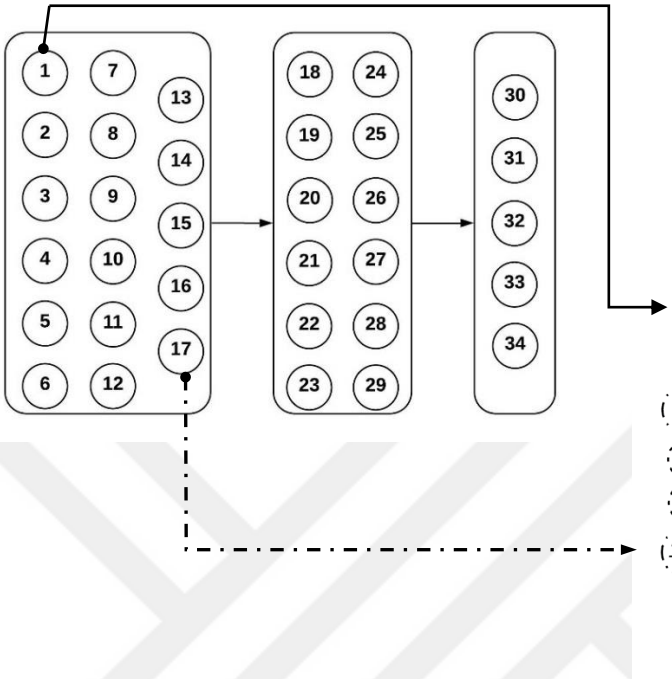


Figure 6.2 Precedence relations of the assembly line

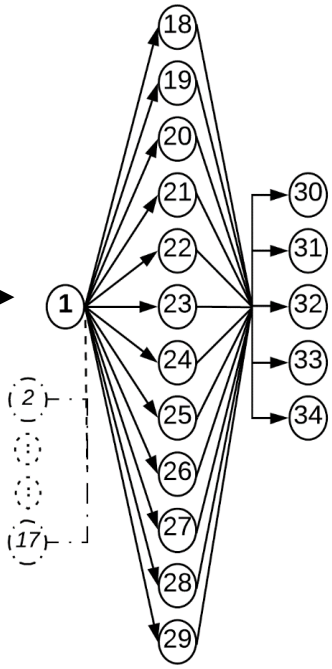


Figure 6.3 Precedence diagram of task 1

Table 6.1 Task transportation costs and task performing times of the workers in the industrial case study

Task number	Task transportation cost	Worker task times (seconds)								
		w1	**w2**	**w3**	**w4**	**w5**	**w6**	**w7**	**w8**	**w9**
1	668	-	20	**23**	22	25	20	24	26	21
2	239	15	**17**	15	-	16	16	19	20	16
3	443	14	**16**	14	19	15	17	13	18	-
4	1592	**28**	28	25	27	35	34	-	26	23
5	0	-	48	52	**50**	51	53	52	51	46
6	639	70	63	60	**64**	62	-	63	69	67
7	1537	8	**8**	-	11	7	10	5	9	10
8	0	**14**	-	16	16	12	18	-	13	11
9	0	31	38	30	**35**	38	-	31	-	39
10	598	53	-	**53**	52	52	56	52	56	51
11	720	**21**	25	19	20	-	20	22	20	21
12	524	**11**	-	14	12	10	10	8	14	9

Table 6.1 Continues

Task number	Task transportation cost	Worker task times (seconds)								
		w1	**w2**	**w3**	**w4**	**w5**	**w6**	**w7**	**w8**	**w9**
13	392	**12**	12	10	14	15	10	-	11	-
14	1889	**52**	53	54	51	54	50	55	49	50
15	1307	42	**40**	38	37	40	40	-	38	47
16	680	84	92	**86**	-	85	82	88	84	87
17	0	75	**77**	-	77	76	79	74	73	82
18	0	16	19	18	**17**	15	16	18	16	18
19	1229	32	36	34	36	**34**	31	36	33	-
20	629	46	47	50	50	**49**	52	45	-	52
21	983	31	31	32	32	**33**	36	32	37	-
22	632	32	-	29	30	**30**	29	29	23	37
23	1052	11	7	-	9	**9**	7	10	9	10
24	134	21	25	18	25	-	**22**	21	21	22
25	821	52	53	58	62	51	**57**	-	65	57
26	850	11	13	13	13	-	**14**	16	16	14
27	0	-	6	6	5	-	**5**	4	4	5
28	475	65	58	55	50	53	**56**	-	60	56
29	473	7	11	10	8	12	**10**	12	-	10
30	0	34	34	31	-	34	41	**34**	30	34
31	971	19	17	15	15	17	-	**17**	17	21
32	1079	40	44	48	-	44	48	**44**	48	36
33	0	29	26	-	26	25	-	**26**	25	24
34	1431	46	49	48	49	-	47	**49**	52	52

6.3 ErgoALWARBP Decision Support Software: Inputs, Outputs, and Solution Environment

The ErgoALWARBP software was developed with C# by Visual Studio 2015. The software can read the inputs from an Excel workbook that is filled in the pre-specified format. The five different input categories and their content are as follows:

- Initial data: Tasks (with names), workers, task precedence relations, and the task processing time of each worker for each task (task-worker processing time matrix) (Table 6.1).
- Work/time study: Task details (sub-tasks), including operation type and sub-task processing time information. Twelve operation types were defined and analyzed for specifying posture and force values of each operation that are used in OCRA ergonomic calculations (for OCRA input parameters, see Baykasoglu et al. (2017): 114-115).
- Kitting data: Technical move count variables and values of each task. These data are generated after a detailed analysis of technical drawings and assembly lines which are used in OCRA ergonomic calculations.
- Initial assignment: Assembly line stations, station task assignment, station worker assignment, station time, cycle time.
- New assembly line data: New cycle time, station operating/opening/closing costs, task transportation costs, operation duration (months), iteration, and run count (for calculations).

The software generates two outputs which are:

- Task and worker assignment with ergonomic results of the case study assembly line (Table 6.2).
- Solutions with objective value results (Table 6.3).

6.4 Industrial Case Study Implementation, Results, and Discussion

The software was run for 500 iterations on a Toshiba notebook with a 2.0 GHz AMD Turion Mobile TL-60 CPU. The station assignment details of the best solutions for each Goal are shared in Table 6.2. The solution names are given in the first column. In columns two and three, the station numbers of the assembly line and the assigned workers to each station are shared, respectively. In column four, the assigned tasks for each station are shared, which tasks are separated by commas. In columns five and six, each station's OCRA index and station time are shared, respectively. As an example of reading the results, there are seven stations in the initial assembly line, where worker

7(*w7*) was assigned to station 7(*s7*). Tasks 30, 31, 32, 33, and 34 were assigned to *s7*, and *w7* can process these tasks in 170 seconds (station time). The *OCRA index* for *s7* is calculated as 10.99. In solution Goal-1, *w7* will still operate in *s7* and implement tasks 30, 32, 33, and 34. So, while tasks 30, 32, 33, and 34 will stay in *s7*, task 31 will be moved from *s7* to *s6* in the rebalanced AL.

The same solution was found for min. *LE* (G6) and min. *SX* (G7). The *SX* values range between 4.12 and 25.44 (except for one solution with SX=33.08) for *LE* values higher than 95.75. These high-performance results are remarkably close to each other, which is better than the initial solution. For lower efficiencies, the SX dropped around 100 and ranged widely between 75 and 145, which became not dependent on *LE* values. The relation between SX and LE values of the 500 feasible solutions is shared in Figure 6.4.



Figure 6.4 Line efficiency & line smoothness index values of the feasible solutions

Table 6.2 Initial assignment's and the proposed solutions' station-task and station-worker assignments of the industrial case-study

Solutions	Station	Assigned Worker	Assigned Tasks	OCRA Index	Station Time
Initial Assignment	s1	w1	4, 8, 11, 12, 13, 14	9.66	138
	s2	w2	2, 3, 7, 15, 17	10.55	158
	s3	w3	1, 10, 16	9.99	162
	s4	w4	5, 6, 9, 18	14.63	166
	s5	w5	19, 20, 21, 22, 23	15.90	155
	s6	w6	24, 25, 26, 27, 28, 29	17.13	164
	s7	w7	30, 31, 32, 33, 34	10.99	170
Solution Goal-1 (min. Cost)	s1	w3	2, 9, 14, 13, 4, 11	11.20	153
	s2	w5	15, 12, 16, 3, 7	11.83	157
	s3	w4	10, 5, 1, 8	10.71	140
	s4	w2	6, 17, 29, 23	11.08	158
	s5	w1	21, 25, 20, 26, 18	24.65	156
	s6	w8	22, 19, 28, 27, 24, 31	11.90	158
	s7	w7	32, 33, 34, 30	10.74	153
Goal-2 (max. MSF)	s1	w6	10, 16, 1	10.25	158
	s2	w7	17, 5, 9	11.37	157
	s3	w1	12, 4, 13, 8, 14, 11, 2	9.49	153
	s4	w2	6, 3, 15, 7, 21	13.71	158
	s5	w4	22, 27, 29, 28, 18, 23, 26, 24	18.12	157
	s6	w5	25, 19, 20, 31	17.74	151
	s7	w8	33, 30, 34, 32	10.60	155
Goal-3 (max. worker MSF)	s1	w1	9, 4, 13, 11, 12, 3, 7, 8, 2	13.61	154
	s2	w8	17, 16	7.41	157
	s3	w3	6, 10, 1	10.68	136
	s4	w4	15, 5, 14, 18	12.44	155
	s5	w6	19, 22, 25, 27, 26, 24	20.57	158
	s6	w2	21, 28, 29, 20, 23	13.14	154
	s7	w7	34, 30, 32, 33	10.74	153
	s8	w9	31	10.77	21
Goal-4 (min. MAD of OCRA)	s1	w8	5, 16, 8, 7	13.04	157
	s2	w3	15, 6, 11, 9, 13	13.35	157
	s3	w5	4, 12, 1, 2, 10, 3	9.18	153
	s4	w1	14, 17, 26, 18	15.23	154
	s5	w4	24, 28, 19, 23, 21, 27	13.65	157
	s6	w9	22, 20, 29, 25	14.65	156
	s7	w7	32, 34, 30, 33	10.74	153
	s8	w2	31	13.31	17

Table 6.2 Continues

Solutions	Station	Assigned Worker	Assigned Tasks	OCRA Index	Station Time
Goal-5 (min. nr. of tasks moved)	s1	w3	13, 15, 9, 12, 3, 11, 1	10.78	148
	s2	w1	7, 17, 10, 2	11.98	151
	s3	w9	16, 4, 5	9.38	156
	s4	w5	8, 6, 14, 23, 18	12.30	152
	s5	w7	20, 19, 29, 27, 21, 22	15.52	158
	s6	w6	28, 25, 24, 26	17.58	149
	s7	w2	34, 32, 30, 31	12.65	144
	s8	w4	33	1.83	26
Goal-6 (max. line efficiency) & Goal-7 (min. smoothness index)	s1	w9	14, 10, 5, 12	10.84	156
	s2	w6	16, 4, 13, 1, 7	10.99	156
	s3	w1	2, 11, 17, 9, 8	11.60	156
	s4	w3	6, 3, 15, 24, 27, 18	15.62	154
	s5	w2	21, 29, 28, 20, 23	13.14	154
	s6	w4	25, 22, 19, 26, 31	18.62	156
	s7	w7	30, 33, 34, 32	10.74	153

Computational results of the best solutions for the seven goals, the average results, and the standard deviation values of all feasible solutions are shared in Table 6.3.

In the best solution for Goal-1 ($\boldsymbol{min}.TC_{rb}$), 15 tasks were transported, which had a 7471 EUR rebalancing cost. This cost is 54% less than solution Goal-2 (max. MSF), 57% less than solution Goal-4 ($min.madOCRA$), 42% less than solution Goal-5 (min. task movement), and 55% less than the average cost of all feasible solutions. Although it has better results in terms of the LE and SX in comparison with the initial assignment, there is a 13% increase in the OCRA result, which is the worst station ergonomics among the solutions. The distribution of solutions in terms of the total cost of rebalancing is shared in Figure 6.5.

Table 6.3 Computational results of the industrial case study

Solution	CPU time (ms)	No. of stations	Cycle time (s)	Total rebalancing cost (EUR)	Task movement cost (EUR)	No. of tasks moved	MSF	Worker MSF	OCRA MAD	Line efficiency	Smoothness index
Initial assignment	-	7	170	-	-	-	-	-	2.74	93.53	38.85
Goal1- min. rebalancing cost	209	7	158	**7471**	7471	15	0.31	0.18	3.10	97.20	19.47
G2- max. MSF	288	7	158	16333	16333	25	**0.56**	0.32	2.99	98.46	9.22
G3- max. worker MSF	207	8	158	17174	12383	16	0.44	**0.43**	2.52	86.08	139.00
G4- min. MAD of OCRA indexes	239	8	157	17346	12346	20	0.28	0.12	**1.47**	87.90	140.15
G5- min. no. of tasks moved	211	8	158	13022	8022	**13**	0.38	0.15	3.13	85.76	133.75
G6- max. line efficiency G7- min. smoothness index	196	7	156	10553	10533	20	0.30	0.16	2.33	**99.36**	**4.12**
Average of 500 feasible solutions	211.2	7.57	157.6	16879	6650	20.96	0.31	0.15	3.08	89.92	89.84
Standard deviation	15.47	0.46	0.72	3044	2500	2.87	0.05	0.08	0.54	5.01	50.59

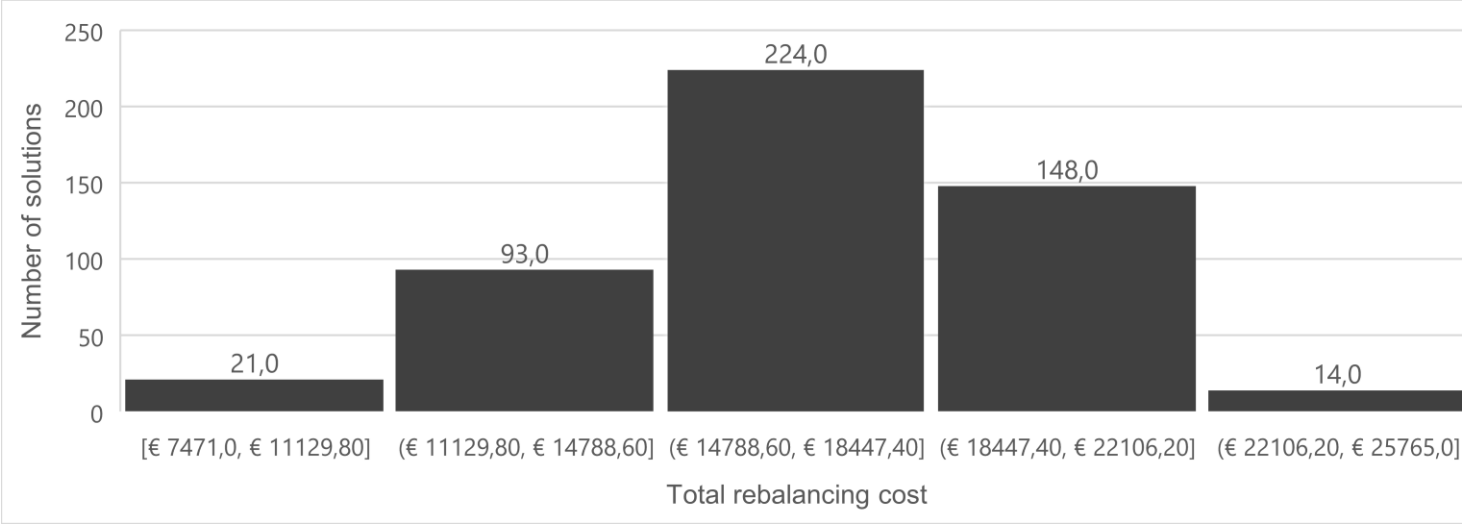


Figure 6.5 The distribution of the feasible solutions in terms of total rebalancing cost

The solution with the best ergonomics (G4) has 46% better ergonomics than the initial solution. However, it had the worst rebalancing cost with 17346 EUR and got worse LE and SX compared to the initial assignment. Also, the rebalancing cost will become higher because of the operational cost ($cost_{run}$) of its additional station if the planning period will be longer. This led the company not to choose this solution.

Although it costs 3K EUR more than the min. cost solution, the company chose to implement the best solution for G6 and G7. The solution results are analyzed as below:

- With a 6% improvement, the *LE* is close to the optimum efficiency value of 100%, with LE=99.36%.
- The *SX* is decreased by 89% to 4.12, close to the optimum SX value of 1.
- Although having a 36% worse OCRA result than the solution with the best ergonomics (G4), the solution improves line ergonomics by 14.9% compared to the initial assignment.
- The solution has remarkably close results to the minimum cost solution (G1) in terms of MSF (3% lower) and worker MSF (11% lower).

After implementing the suggested method in the real-world assembly line and analyzing the solutions, we can conclude that it is crucial to decide the order of importance of the objectives (the goals) while choosing the best solution. The planning period and the number of stations of the rebalanced line are essential factors while calculating the total cost of rebalancing. The operating cost ($cost_{run}$) changes in direct relation with the planning period. As it is shared in Eq. 1, the change in the no. of stations is the multiplier of $cost_{run}$. The total cost can change a lot in more extended planning periods if the number of stations is changed in the rebalanced AL. Also, the number of task moves has a low effect on total rebalancing cost if tasks with a low moving cost are selected for movement. Therefore, it is essential to define the planning period well if the cost has a high priority for the company.

With the ErgoALWARBP solution method and software, several benefits are gained in the company for solving the AL rebalancing problem. The most valuable gains can be summarized as below:

- Method engineers were manually solving the line rebalancing problem before the ErgoALWARBP software. Therefore, the problem-solving speed and the solution quality varied by the experience level of the engineers. The dependency on the experience level of engineers and the risk of making calculation errors were eliminated with the help of our solution method and the decision support software.
- We considered seven objectives in our solution method, described in section 3.3. Method engineers can manually solve the rebalancing problem in 4-6 weeks/engineer according to the scope of the change request and their experience level, although they can only consider a smaller number of objectives and constraints simultaneously. In a frequently changing agile work environment, the customer may request several product changes during the solution implementation, which causes a lot of time and material waste. The ErgoALWARBP software can calculate and propose several effective solutions that meet different goals in just seconds. Thus, the software improved the team's agility for the customer change requests by increasing the solution speed at least ten thousand times, decreasing the rebalancing cost, and eliminating the cost of waste.
- Manually calculating line ergonomics needs so much time and effort, which caused the team to ignore ergonomic criteria during the initial rebalancing. It was usually preferred to solve ergonomic problems according to feedback from the production area after solution implementation. With the help of ErgoALWARBP software, the team could find solutions with better ergonomics, considering all ergonomic criteria. This helped to decrease the costs that were related to additional ergonomic improvements and quickly improve the assembly line’s ergonomic conditions, which may cause musculoskeletal disorders.

CHAPTER 7
CONCLUSION

Many manufacturing-based businesses face assembly line rebalancing problems (ALRBP) in competitive markets. They are constantly obliged to adjust product characteristics and are sensitive to sales volume fluctuations. This causes a need for rebalancing their assembly lines (ALs), which requires redefining workstation workloads and reassigning tasks to operators (Gamberini et al., 2009). Although the assembly line balancing problem (ALBP) has been studied since 1955 (Salveson, 1955), ALRBP is a relatively new problem in the academic area. In this study, the current literature has been reviewed to understand the ALRBPs comprehensively.

We classified the reviewed ALRBP papers according to six aspects: ALRBP classification, objective functions, problem constraints, causes for rebalancing, solution methods, and test data types for solution evaluation. ALRBPs are classified into four aspects: the type of product mix in the AL, the line layout, variability of task times, and additional problems incorporated in the ALRBP.

Heuristic and meta-heuristic approaches were widely used in the reviewed studies. To deal with NP-hard ALRBP. 18% of the studies also considered worker assignment problems together with ALRBP, which is also another critical aspect. Since every worker has their ability, skill, and experience, the operation time of a particular task might vary from one worker to another. In many configurations, operators should be executing different tasks due to rebalancing. Some workers might be more successful at executing specific tasks, while others might not. So, worker assignments should be considered to distribute the workload as smoothly as possible.

In this thesis, a novel rebalancing problem about workers' specific skills is studied under ergonomic aspects, the ErgoALWARBP. A formal definition of the problem is stated, which is multi-objective by nature. The objectives are maximizing the similarity of the assigned tasks to the stations, the similarity of performed tasks by the workers, and the line efficiency while minimizing the total cost of rebalancing, line

smoothness index, ergonomic objective, and the number of relocated tasks. To deal with the novel ErgoALWARBP, a randomized constructive rule-based heuristic approach is developed, and a preemptive goal programming approach is applied. To evaluate the performance of the developed rebalancing method, it is tested on well-known benchmark data. Experimental results showed the efficiency of the developed algorithm. Then, the solution method is applied by a company that manufactures automotive harnesses by using the developed decision support software. The generated solutions were evaluated by the engineers. The most fitting solution was chosen and implemented on the assembly line among the suggested alternatives. Specifically, the proposed systematic approach improves problem-solving speed, assembly line ergonomics, line efficiency, and line smoothness index with less rebalancing cost. The developed ErgoALWARBP software improved the organization's agility for frequent changes, which may lead to the need for rebalancing, and decreased the dependency of the solution quality on the engineer’s experience level. Future researchers can extend the ErgoALWARBP and the heuristic solution method to solve U-type/two-sided line configurations, multi/mixed-model assembly lines, and stochastic worker availability/task times.

REFERENCES

Ağpak, K. (2010). An approach to find task sequence for re-balancing of assembly lines. *Assembly Automation*, *30*(4), 378–387. https://doi.org/10.1108/01445151011075834

Akyol, S. D., & Baykasoğlu, A. (2019a). ErgoALWABP: a multiple-rule based constructive randomized search algorithm for solving assembly line worker assignment and balancing problem under ergonomic risk factors. *Journal of Intelligent Manufacturing*, *30*(1), 557–573. https://doi.org/10.1007/s10845-016-1246-6

Akyol, S. D., & Baykasoğlu, A. (2019b). A multiple-rule based constructive randomized search algorithm for solving assembly line worker assignment and balancing problem. *Journal of Intelligent Manufacturing*, *30*(2), 557–573. https://doi.org/10.1007/s10845-016-1262-6

Altemeier, S., Helmdach, M., Koberstein, A., & Dangelmaier, W. (2010). Reconfiguration of assembly lines under the influence of high product variety in the automotive industry - A decision support system. *International Journal of Production Research*, *48*(21), 6235–6256. https://doi.org/10.1080/00207540903252290

Antoine, M., El-Haouzi Hind, B., Cherif-Khettaf Wahiba, R., & Mohand Lounes, B. (2016). Iterated Local Search for dynamic assembly line rebalancing problem. *IFAC-PapersOnLine*, *49*(12), 515–519. https://doi.org/10.1016/j.ifacol.2016.07.679

Antoine, M., Hind, B. E.-H., André, T., & Jean-François, P. (2014). Dynamic Rebalancing of an Assembly Line with a Reachability Analysis of Communicating Automata. *International Conference on Advances in Production Management Systems*, *438*, 597–604. https://doi.org/10.1007/978-3-662-44739-0_73

Arcus, A. L. (1966). COMSOAL A Computer Method of Sequencing Operations for

Assembly Lines. *International Journal of Production Research*, *4*(4), 259–277. https://doi.org/10.1080/00207546508919982

Bartholdi, J. J. (1993). Balancing two-sided assembly lines: A case study. *International Journal of Production Research*, *31*(10), 2447–2461. https://doi.org/10.1080/00207549308956868

Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, *142*(2), 259–277. https://doi.org/10.1016/j.ijpe.2012.10.020

Baykasoglu, A., Tasan, S. O., Tasan, A. S., & Akyol, S. D. (2017). Modeling and solving assembly line design problems by considering human factors with a real-life application. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *27*(2), 96–115. https://doi.org/10.1002/hfm.20695

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. https://doi.org/10.1016/j.ejor.2004.07.023

Belassiria, I., Mazouzi, M., ELfezazi, S., Cherrafi, A., & ELMaskaoui, Z. (2018). An integrated model for assembly line re-balancing problem. *International Journal of Production Research*, *56*(16), 5324–5344. https://doi.org/10.1080/00207543.2018.1467061

Berk, M., Önal, B., & Güven, R. (2011). *Meslek Hastalıkları Rehberi*. Matsa Basımevi.

Borba, L., & Ritt, M. (2014). A heuristic and a branch-and-bound algorithm for the Assembly Line Worker Assignment and Balancing Problem. *Computers & Operations Research*, *45*, 87–96. https://doi.org/10.1016/j.cor.2013.12.002

Boysen, N., Fliedner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, *183*(2), 674–693. https://doi.org/10.1016/j.ejor.2006.10.010

Boysen, N., Schulze, P., & Scholl, A. (2022). Assembly line balancing: What happened in the last fifteen years? *European Journal of Operational Research*, *301*(3), 797–814. https://doi.org/10.1016/j.ejor.2021.11.043

Budak, G., & Chen, X. (2020). Pace making and worker reassignment for assembly line rebalancing. *SN Applied Sciences*, *2*(9), 1–15. https://doi.org/10.1007/s42452-020-03368-9

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. https://doi.org/10.1016/S0360-8352(01)00065-1

Celik, E., Kara, Y., & Atasagun, Y. (2014). A new approach for rebalancing of U-lines with stochastic task times using ant colony optimisation algorithm. *International Journal of Production Research*, *52*(24), 7262–7275. https://doi.org/10.1080/00207543.2014.917768

Chaves, A. A., Lorena, L. A. N., & Miralles, C. (2007). Clustering search approach for the assembly line worker assignment and balancing problem. *Proceedings of the 37th International Conference on Computers and Industrial Engineering*, 1469–1478.

Çimen, T., Baykasoğlu, A., & Demirkol Akyol, S. (2022a). Assembly line rebalancing and worker assignment considering ergonomic risks in an automotive parts manufacturing plant. *International Journal of Industrial Engineering Computations*, *13*(3), 363–384. https://doi.org/10.5267/j.ijiec.2022.2.001

Çimen, T., Baykasoğlu, A., & Demirkol Akyol, S. (2022b). A detailed review and analysis of assembly line rebalancing problems. *Assembly Automation*, *42*(6), 742–760. https://doi.org/10.1108/AA-02-2022-0031

Corominas, A., Pastor, R., & Plans, J. (2008). Balancing assembly line with skilled and unskilled workers. *Omega*, *36*(6), 1126–1132. https://doi.org/10.1016/j.omega.2006.03.003

Costa, A. M., & Miralles, C. (2009). Job rotation in assembly lines employing disabled

workers. *International Journal of Production Economics*, *120*(2), 625–632. https://doi.org/10.1016/j.ijpe.2009.04.013

Dar-El, E. M., & Rubinovitz, J. (1991). Using learning theory in assembly lines for new products. *International Journal of Production Economics*, *25*(1–3), 103–109. https://doi.org/10.1016/0925-5273(91)90136-H

Fichera, S., Costa, A., & Cappadonna, F. A. (2017). Heterogeneous workers with learning ability assignment in a cellular manufacturing system. *International Journal of Industrial Engineering Computations*, 427–440. https://doi.org/10.5267/j.ijiec.2017.3.005

Gamberini, R., Gebennini, E., Grassi, A., & Regattieri, A. (2009). A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem. *International Journal of Production Research*, *47*(8), 2141–2164. https://doi.org/10.1080/00207540802176046

Gamberini, R., Grassi, A., & Rimini, B. (2006). A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem. *International Journal of Production Economics*, *102*(2), 226–243. https://doi.org/10.1016/j.ijpe.2005.02.013

Gel, E. S., Hopp, W. J., & Van Oyen, M. P. (2002). Factors affecting opportunity of worksharing as a dynamic line balancing mechanism. *IIE Transactions (Institute of Industrial Engineers)*, *34*(10), 847–863. https://doi.org/10.1080/07408170208928917

Girit, U., & Azizoğlu, M. (2021). Rebalancing the assembly lines with total squared workload and total replacement distance objectives. *International Journal of Production Research*, *59*(22), 6702–6720. https://doi.org/10.1080/00207543.2020.1823027

Gökçen, H., & Erel, E. (1998). Binary Integer Formulation For Mixed-Model Assembly Line Balancing Problem. *Computers & Industrial Engineering*, *34*(2), 451–461. https://doi.org/10.1016/S0360-8352(97)00142-3

Grangeon, N., Leclaire, P., & Norre, S. (2011). Heuristics for the re-balancing of a vehicle assembly line. *International Journal of Production Research*, *49*(22), 6609–6628. https://doi.org/10.1080/00207543.2010.539025

Guo, Z. X., Wong, W. K., Leung, S. Y. S., Fan, J. T., & Chan, S. F. (2008). A Genetic-Algorithm-Based Optimization Model for Solving the Flexible Assembly Line Balancing Problem With Work Sharing and Workstation Revisiting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(2), 218–228. https://doi.org/10.1109/TSMCC.2007.913912

Hignett, S., & McAtamney, L. (2000). Rapid Entire Body Assessment (REBA). *Applied Ergonomics*, *31*(2), 201–205. https://doi.org/10.1016/S0003-6870(99)00039-3

Hu, X., Wu, C., & Zhang, Y. (2020). A heuristic method for two-sided assembly line reconfiguration in IoT environment. *Journal of Internet Technology*, *21*(5), 1491–1502. https://doi.org/10.3966/160792642020092105022

Ishigaki, A. (2018). Heuristic approach and application for rebalancing of a mixed-model assembly line. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, *12*(3), 1–10. https://doi.org/10.1299/jamdsm.2018jamdsm0078

Karas, A., & Ozcelik, F. (2021). Assembly line worker assignment and rebalancing problem: A mathematical model and an artificial bee colony algorithm. *Computers & Industrial Engineering*, *156*(1), 107195. https://doi.org/10.1016/j.cie.2021.107195

Karp, R. M. (1972). Reducibility among Combinatorial Problems. In *Complexity of Computer Computations* (pp. 85–103). Springer US. https://doi.org/10.1007/978-1-4684-2001-2_9

Lai, R., Hou, L., & Wu, Y. (2015). Evolution Balancing of the Small-Sized Wheel Loader Assembly Line. *Shock and Vibration*, *2015*, 1–11. https://doi.org/10.1155/2015/213159

Li, G., & Buckle, P. (1999). Current techniques for assessing physical exposure to

work-related musculoskeletal risks, with emphasis on posture-based methods. *Ergonomics*, *42*(5), 674–695. https://doi.org/10.1080/001401399185388

Li, Y. (2017). The type-II assembly line rebalancing problem considering stochastic task learning. *International Journal of Production Research*, *55*(24), 7334–7355. https://doi.org/10.1080/00207543.2017.1346316

Li, Y., & Boucher, T. O. (2017). Assembly line balancing problem with task learning and dynamic task reassignment. *The International Journal of Advanced Manufacturing Technology*, *88*(9–12), 3089–3097. https://doi.org/10.1007/s00170-016-9014-5

Li, Y., Li, Z., & Saldanha-da-Gama, F. (2022). New approaches for rebalancing an assembly line with disruptions. *International Journal of Computer Integrated Manufacturing*, *35*(10–11), 1059–1076. https://doi.org/10.1080/0951192X.2021.1925967

Liu, C., Li, W., Lian, J., & Yin, Y. (2012). Reconfiguration of assembly systems: From conveyor assembly line to serus. *Journal of Manufacturing Systems*, *31*(3), 312–325. https://doi.org/10.1016/j.jmsy.2012.02.003

Makssoud, F., Battaïa, O., Dolgui, A., Mpofu, K., & Olabanji, O. (2015). Re-balancing problem for assembly lines: new mathematical model and exact solution method. *Assembly Automation*, *35*(1), 16–21. https://doi.org/10.1108/AA-07-2014-061

Makssoud, F., Battäıa, O., Dolgui, A., Battaïa, O., & Dolgui, A. (2013). An Exact Method for the Assembly Line Re-balancing Problem. *International Conference on Advances in Production Management Systems*, *414*, 159–166. https://doi.org/10.1007/978-3-642-41266-0_20

Mardani-Fard, H. A., Hadi-Vencheh, A., Mahmoodirad, A., & Niroomand, S. (2020). An effective hybrid goal programming approach for multi-objective straight assembly line balancing problem with stochastic parameters. *Operational Research*, *20*(4), 1939–1976. https://doi.org/10.1007/s12351-018-0428-8

McAtamney, L., & Nigel Corlett, E. (1993). RULA: a survey method for the

investigation of work-related upper limb disorders. *Applied Ergonomics*, *24*(2), 91–99. https://doi.org/10.1016/0003-6870(93)90080-S

Michels, A. S., & Costa, A. M. (2022). Conserving workforce while temporarily rebalancing assembly lines under demand disruption. *International Journal of Production Research*, *60*(21), 6616–6636. https://doi.org/10.1080/00207543.2021.1998694

Miralles, C., García-Sabater, J. P., Andrés, C., & Cardós, M. (2008). Branch and bound procedures for solving the Assembly Line Worker Assignment and Balancing Problem: Application to Sheltered Work centres for Disabled. *Discrete Applied Mathematics*, *156*(3), 352–367. https://doi.org/10.1016/j.dam.2005.12.012

Mokhtari, H., & Mozdgir, A. (2015). A Bi-Objective Approach for Design of an Assembly Line Re-Balancing System: Mathematical Model and Differential Evolution Algorithms. *International Journal of Industrial Engineering & Production Research*, *26*(2), 105–117. http://ijiepr.iust.ac.ir/article-1-613-en.html

Moreira, M. C. O., Miralles, C., & Costa, A. M. (2015). Model and heuristics for the Assembly Line Worker Integration and Balancing Problem. *Computers & Operations Research*, *54*, 64–73. https://doi.org/10.1016/j.cor.2014.08.021

Occhipinti, E. (1998). OCRA: A concise index for the assessment of exposure to repetitive movements of upper limbs. *Ergonomics*, *41*, 1290–1311.

Oliveira, F. S., Vittori, K., Russel, R. M. O., & Travassos, X. L. (2012). Mixed assembly line rebalancing: A binary integer approach applied to real world problems in the automotive industry. *International Journal of Automotive Technology*, *13*(6), 933–940. https://doi.org/10.1007/s12239-012-0094-4

Otto, A., & Scholl, A. (2011). Incorporating ergonomic risks into assembly line balancing. *European Journal of Operational Research*, *212*(2), 277–286. https://doi.org/10.1016/j.ejor.2011.01.056

Rahman, C. L. (2010). Re-balancing of Generalized Assembly Lines – Searching

Optimal Solutions for SALBP. *International Conference on Industrial Engineering and Operations Management*, *January*, 3–8. https://doi.org/10.13140/2.1.1620.7047

Ramezanian, R., & Khalesi, S. (2021). Integration of multi-product supply chain network design and assembly line balancing. *Operational Research*, *21*(1), 453–483. https://doi.org/10.1007/s12351-019-00453-9

Salveson, M. E. (1955). The assembly line balancing problem. *The Journal of Industrial Engineering*, *6*, 18–25. http://ci.nii.ac.jp/naid/10003095017/en/

Samadhi, T., & Sumihartati, A. (2016). Model for Assembly Line Re-Balancing Considering Additional Capacity and Outsourcing to Face Demand Fluctuations. *IOP Conference Series: Materials Science and Engineering*, *114*, 012092. https://doi.org/10.1088/1757-899X/114/1/012092

Sancı, E., & Azizoğlu, M. (2017). Rebalancing the assembly lines: exact solution approaches. *International Journal of Production Research*, *55*(20), 5991–6010. https://doi.org/10.1080/00207543.2017.1319583

Schaub, K., Caragnano, G., Britzke, B., & Bruder, R. (2013). The European Assembly Worksheet. *Theoretical Issues in Ergonomics Science*, *14*(6), 616–639. https://doi.org/10.1080/1463922X.2012.678283

Scholl, A. (1993). Data of assembly line balancing problems. *Schriften Zur Quantitativen Betriebswirtschaftslehre*, *16/93*.

Scholl, A. (1999). *Balancing and Sequencing of Assembly Lines* (2nd ed., p. 318). Physica. https://ideas.repec.org/p/dar/wpaper/10881.html

Scholl, A., & Klein, R. (1997). SALOME: A Bidirectional Branch-and-Bound Procedure for Assembly Line Balancing. *INFORMS Journal on Computing*, *9*(4), 319–334. https://doi.org/10.1287/ijoc.9.4.319

Scholl, A., & Klein, R. (1999). ULINO: Optimally balancing U-shaped JIT assembly lines. *International Journal of Production Research*, *37*(4), 721–736.

https://doi.org/10.1080/002075499191481

Sculli, D. (1979). Dynamic aspects of line balancing. *Omega*, *7*(6), 557–561. https://doi.org/10.1016/0305-0483(79)90075-6

Sculli, D. (1984). Short term adjustments to production lines. *Computers and Industrial Engineering*, *8*(1), 53–63. https://doi.org/10.1016/0360-8352(84)90021-4

Serin, F., Mete, S., & Çelik, E. (2019). An efficient algorithm for U-type assembly line re-balancing problem with stochastic task times. *Assembly Automation*, *39*(4), 581–595. https://doi.org/10.1108/AA-07-2018-106

Sikora, C. G. S., Lopes, T. C., & Magatão, L. (2017). Traveling worker assembly line (re)balancing problem: Model, reduction techniques, and real case studies. *European Journal of Operational Research*, *259*(3), 949–971. https://doi.org/10.1016/j.ejor.2016.11.027

Slack, N., Chambers, S., & Johnston, R. (1995). *Operations Management* (1st ed.). Prentice-Hall.

Tóth, N., & Kulcsár, G. (2021). New models and algorithms to solve integrated problems of production planning and control taking into account worker skills in flexible manufacturing systems. *International Journal of Industrial Engineering Computations*, *12*(4), 381–400. https://doi.org/10.5267/j.ijiec.2021.5.004

Van Oyen, M. P., Gel, E. G., & Hopp, W. J. (2001). Performance opportunity for workforce agility in collaborative and noncollaborative work systems. *IIE Transactions (Institute of Industrial Engineers)*, *33*(9), 761–777. https://doi.org/10.1080/07408170108936871

Waters, T. R. (1994). *Applications Manual for the Revised NIOSH Lifting Equation*. U.S. Department of Health and Human Services, Public Health Service, Centers for Disease Control and Prevention, National Institute for Occupational Safety and Health, Division of Biomedical and Behavioral Science.

Yang, C., & Gao, J. (2016). Balancing mixed-model assembly lines using adjacent cross-training in a demand variation environment. *Computers & Operations Research*, *65*, 139–148. https://doi.org/10.1016/j.cor.2015.07.007

Yang, C., Gao, J., & Sun, L. (2013). A multi-objective genetic algorithm for mixed-model assembly line rebalancing. *Computers & Industrial Engineering*, *65*(1), 109–116. https://doi.org/10.1016/j.cie.2011.11.033

Yılmaz, E., & Erol, R. (2005). Montaj hatlarının değişen koşullar altında yeniden dengelenmesi. *Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, *20*(1), 213–227.

Zacharia, P. T., & Nearchou, A. C. (2016). A population-based algorithm for the bi-objective assembly line worker assignment and balancing problem. *Engineering Applications of Artificial Intelligence*, *49*, 1–9. https://doi.org/10.1016/j.engappai.2015.11.007

Zha, J., & Yu, J. J. (2014). A hybrid ant colony algorithm for U-line balancing and rebalancing in just-in-time production environment. *Journal of Manufacturing Systems*, *33*(1), 93–102. https://doi.org/10.1016/j.jmsy.2013.08.002

Zhang, D. Z., Kucukkoc, I., & Karaoglan, A. D. (2016). Rebalancing of mixed-model two-sided assembly lines with incompatible task groups: An industrial case study. *46th International Conference on Computers & Industrial Engineering (CIE46)*.

Zhang, Y., Hu, X., & Wu, C. (2018a). A modified multi-objective genetic algorithm for two-sided assembly line re-balancing problem of a shovel loader. *International Journal of Production Research*, *56*(9), 3043–3063.

Zhang, Y., Hu, X., & Wu, C. (2018b). Heuristic Algorithm for Type II Two-sided Assembly Line Rebalancing Problem with Multi-objective. *MATEC Web of Conferences*, *175*.

Zhang, Y., Hu, X., & Wu, C. (2018c). Push Rule-Based Heuristic Method for Multi-objective Two-Sided Assembly Line Rebalancing Problem. *DEStech Transactions on Engineering and Technology Research*, *pmsms*.

Zhang, Y., Hu, X., & Wu, C. (2020). Improved imperialist competitive algorithms for rebalancing multi-objective two-sided assembly lines with space and resource constraints. *International Journal of Production Research*, *58*(12), 3589–3617. https://doi.org/10.1080/00207543.2019.1633023

APPENDICES

APPENDIX A: Calculation of OCRA Index:

$$OCRA = Actual\ frequency\ /\ Recommended\ frequency \quad \text{(A.1)}$$

$$Actual\ frequency\ = \frac{Number\ of\ technical\ actions}{station\ time} * 60 \quad \text{(A.2)}$$

$$Recommended\ frequency\ = CF * PM * FM * RM * ARF * (RcM * DuM) \quad \text{(A.3)}$$

$$avgOCRA = \frac{sum\ of\ OCRA\ indexes\ of\ all\ stations}{number\ of\ stations} \quad \text{(A.4)}$$

$$madOCRA = \sqrt{\frac{(avgOCRA - OCRA_a)^2}{m-1}} \quad \text{(A.5)}$$

OCRA index is calculated for each station after the assignment of tasks and workers in that particular station has been completed. Details of the OCRA index parameters can be examined from Akyol & Baykasoğlu (2019a) and Otto & Scholl (2011).