DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A FIX AND OPTIMIZE ALGORITHM FOR THE ASSEMBLY LINE WORKER ASSIGNMENT AND BALANCING PROBLEM

by
Anıl AKPUNAR

July, 2022
İZMİR

A FIX AND OPTIMIZE ALGORITHM FOR THE ASSEMBLY LINE WORKER ASSIGNMENT AND BALANCING PROBLEM

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 Master of Science in Industrial Engineering, Industrial Engineering Program

by
Anıl AKPUNAR

July, 2022
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A FIX AND OPTIMIZE ALGORITHM FOR THE ASSEMBLY LINE WORKER ASSIGNMENT AND BALANCING PROBLEM"** completed by **ANIL AKPUNAR** under supervision of **ASSOC. PROF. DR. ŞENER AKPINAR** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

....

Assoc. Prof. Dr. Şener AKPINAR

Supervisor

....

Assoc. Prof. Dr. Fehmi Burçin ÖZSOYDAN

(Jury Member)

....

Assist. Prof. Dr. Mustafa AVCI

(Jury Member)

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Science

ACKNOWLEDGMENTS

Firstly, I would like to thank my supervisor, Assoc. Prof. Dr. Şener AKPINAR for his support; advice and encouragement that helped me overcome challenges throughout my M.Sc. thesis.

I would like to offer my eternal thanks to my valued partner Özge ŞATIR AKPUNAR for her endless support, understanding and patient.

Last, I also would like to thank my valued family, Mualla AKPUNAR, Murat AKPUNAR and my brother, Ahmet AKPUNAR for their love, encouragement and eternal support in my whole life.

Anıl AKPUNAR

A FIX AND OPTIMIZE ALGORITHM FOR THE ASSEMBLY LINE WORKER ASSINGMENT AND BALANCING PROBLEM

ABSTRACT

Mass production of standardised commodities are generally realized through assembly lines which are composed of a number of serial workstations and a material handling system that connect the workstations. The components are sequentially moves from one station to another until a finished product is produced. In today, to meet the customers' rising needs and expectations, it is crucial to response them in short time with reasonable production costs to be competitive. Therefore, companies need to optimize these aspects to increase the production rate or decreasing the production cost. The necessity of these configurations derived the assembly line balancing problem (ALBP). The ALBP seeks for a partition of tasks among the workstations to identify the optimum value of a performance measure under a set of precedence constraints. However, classical ALBP assumes that each task has a deterministic processing time which is an unsuitable assumption for reality, since, a task may have different operation times based upon the worker who executes it. Therefore, assembly line worker assignment and balancing problem (ALWABP), which considers also to optimally allocate operators among the workstations besides partitioning tasks among workstations, was derived to close this gap in the related literature. This study considers an ALWABP with different operation times of tasks because every worker has different ability, skill, and experience etc. On account of the NP-hard nature of the problem, a mat-heuristic algorithm is proposed for the first time in the related literature to get through the problem. The proposed algorithm mainly comprises the fix-and-optimize (FAO) approach and the simulated annealing (SA) algorithm. Performance evaluation tests of the proposed algorithm are done on a benchmark set of ALWABP instances available in the related literature. The computational results exhibit the satisfactory performance of the proposed algorithm and therefore indicate that the proposed algorithm is a competitive solution procedure in solving ALWABP instances.

Keywords: Assembly line worker assignment and balancing problem, fix-and-optimize heuristic, simulated annealing algorithm

MONTAJ HATTI İŞÇİ ATAMA VE DENGELEME PROBLEMİ İÇİN BİR SABİTLE VE OPTİMİZE ET ALGORİTMASI

ÖZ

Standartlaştırılmış ürünlerin seri üretimi genellikle birkaç seri iş istasyonundan ve iş istasyonlarını birbirine bağlayan bir malzeme taşıma sisteminden oluşan montaj hatları aracılığıyla gerçekleştirilir. Parçalar, bitmiş bir ürün üretilene kadar sırayla bir istasyondan diğerine taşınır. Günümüzde müşterilerin artan ihtiyaç ve beklentilerine cevap verebilmek, rekabetçi olabilmek için kabul edilebilir üretim maliyetleri ile kısa sürede cevap verebilmek büyük önem taşımaktadır. Bu nedenle, şirketlerin üretim hızını artırmak veya üretim maliyetini azaltmak için bu yönlerini en iyilemesi gerekmektedir. Bu düzenlemelerin gerekliliği, montaj hattı dengeleme problemini (MHDP) türetmiştir. MHDP, bir dizi öncelik kısıtlaması altında bir performans ölçüsünün eniyi değerini belirlemek için iş istasyonları arasında bir görev dağılımı arar. Bununla birlikte, klasik MHDP, her görevin belirli bir işlem süresine sahip olduğunu varsayar; bu, gerçekliğe uygun olmayan bir varsayımdır, çünkü bir görev, onu yerine getiren işçiye bağlı olarak farklı işlem sürelerine sahip olabilir. Bu nedenle, ilgili literatürdeki bu açığı kapatmak için, iş istasyonları arasında görevlerin dağıtılmasının yanı sıra operatörlerin iş istasyonları arasında en uygun şekilde tahsis edilmesini de dikkate alan montaj hattı işçi atama ve dengeleme problemi (MHİADP) türetilmiştir. Bu çalışma, her işçinin farklı yetenek, beceri ve deneyime sahip olması nedeniyle, görevlerin farklı işlem sürelerine sahip olduğu bir MHİADP 'yi ele almaktadır. Problemin NP-zor doğası nedeniyle, problemin üstesinden gelmek için ilgili literatürde ilk kez bir mat-sezgisel algoritma önerilmiştir. Önerilen algoritma temel olarak sabitle ve optimize et (SOE) yaklaşımını ve tavlama benzetimi (TB) algoritmasını içerir. Önerilen algoritmanın performans değerlendirme testleri, ilgili literatürde mevcut olan bir MHİADP örneklerinin karşılaştırmalı seti üzerinde yapılmıştır. Hesaplama sonuçları, önerilen algoritmanın tatmin edici performansını sergiler ve bu nedenle, önerilen algoritmanın ALWABP örneklerinin çözümünde rekabetçi bir çözüm yöntemi olduğunu gösterir.

Anahtar kelimeler: Montaj hattı işçi atama ve dengeleme problemi, sabitle ve optimize et sezgiseli, tavlama benzetimi algoritması

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 Objectives and Motivation

Mass production of standardised commodities are generally realized through assembly lines. The assembly line concept was firstly introduced by Henry Ford in Ford companies for improving efficiency while reducing production costs. Subsequently, assembly lines have widely been applied in the various type of industries such as automobiles, electronic goods, computers, etc. An assembly line is made up of a number of serial workstations and a material handling system that connect the workstations. Units of components are sequentially moves from one station to another until a finished product is assembled. In each station, the tasks must be completed within a definite time, namely cycle time. In today, to be competitive and to meet the customers' rising needs and expectations, it is crucial to response them in short time with reasonable production costs. Companies need to plan the configuration of the factory adequately to increase the production rate or decreasing the production cost (Cunningham, 1980). Therefore, the need to optimize these aspects derived the assembly line balancing problem (ALBP). The ALBP intend to partition tasks among the workstations to optimize an objective function and satisfy a set of precedence relations. Moreover, classical ALBP assumes that each task has a previously known deterministic processing time. However, this assumption does not correspond the reality, since a task may have different operation times based upon the worker who executes it. To fulfil this gap between the theory and reality, Miralles et al. (2007) acquainted the assembly line literature with the assembly line worker assignment and balancing problem (ALWABP) as a new type of balancing problem in which task times may vary due to worker skills. This problem may be typically faced in assembly lines with variety of task operation times because of the workers' variable abilities, skills and experiences. Moreover, this problem may also be arisen in assembly lines having disabled operators who may unable to execute some tasks. Therefore, ALWABP composed of two interrelated problems that must be solved simultaneously to obtain an efficient line balance; one seeks the optimum partitioning of tasks among workstations while the other seeks the optimum assignment of workers to the workstations. ALWABP can be mainly categorized into four types depending on the objective of the problem: ALWABP-1, ALWABP-2, ALWABP-F and ALWABP-E respectively. ALWABP-1 seeks the minimum number of workstations for a predetermined cycle time while ALWABP-2 seeks the minimum cycle time for a definite number of workstations. ALWABP-F tries to find a feasible solution for combination of values (c, m). ALWABP-E aims to maximize line productivity and efficiency by seeking a combination of values (c, m). This thesis concerns itself with the ALWABP-2 in which the number of available workers is equal to the number of workstations and intends to develop an effective solution procedure to tackle the problem.

1.2 Research Methodology

The classical ALBP considers worker-workstation assignment which makes the problem NP-hard. Because of different operation times depending on the workers, an additional assignment problem is occurred for the classical ALBP. Therefore, the ALWABP has double assignment problems which considers assignment of workers and tasks to workstations, simultaneously. Because of the NP-Hard nature of the ALWABP, there are plenty number of research aimed to develop effective solution procedures for the problem in the related literature. These solution methods could be classified as exact and heuristic solution procedures. Among these solution procedures, to the best of our knowledge and will be clarified in Chapter 3, there is not any algorithmic framework realizing a mathematical programming-based solution method for the ALWABP-2.

Within the context of this thesis, we propose an integer programming (IP)-based solution technique that integrates IP-based Fix-and-Optimize heuristic (FAO) with Simulated Annealing (SA). The FAO is an IP-based improvement heuristic, known as mat-heuristic, which has promising performance in finding near optimal solutions. The proposed FAO solution method is an effective algorithm which hybridizes an exact optimization technique with SA based on the neighbourhood structure. This method is mainly composed of two phases. First phase is about to fixing some of the decision variables by means of the SA algorithm while the second phase is about to optimizing remaining decision variables through an IP model. The proposed algorithm iteratively executes this two phases until a predefined stopping condition is satisfied.

In an IP-based heuristics, a problem is decomposed into a set of subproblems and each subproblem is optimized iteratively. In our proposed solution methodology, we decompose the main problem into two subproblems: task and worker assignment problems. We then fix the decision variables of worker assignment subproblem using SA algorithm in *'fix'* phase. A neighbourhood structure is applied during the iteration to diversify and improve the current solution. With the accepted solution, worker assignment subproblem's variable is fixed to their existing values. In the second phase, *'optimize'* phase, the variables of task assignment subproblem are optimized via IP. This study contributes the related literature by proposing an IP-based solution procedure for the ALWABP for the first time, and then assessing the performance of the proposed algorithm on the standard benchmark problems used in the related literature.

1.3 Outline of the Thesis

The remainder of this thesis is organized as follows: the following chapter contains the problem definition, the developed IP model for ALWABP-2 and a literature survey on the ALWABPs. This chapter provides the detailed information on the model and proposed algorithm.

An explanation and a brief overview on the FAO heuristics literature given in Chapter 3.

Chapter 4 presents the proposed algorithm and evaluates the algorithm on a set of well-known ALWABP instances.

Finally, in addition to summary of the thesis, the contribution of this thesis and future research directions are discussed in the last chapter.

CHAPTER TWO

PROBLEM DEFINITION AND LITERATURE SURVEY

This chapter firstly describes the ALWABP through a formerly developed formulation (Miralles, Garcia-Sabater, Andres, & Cardos, 2007) and then reviews the literature on ALWABP.

2.1 Assembly Line Worker Assignment and Balancing

In real-life, manufacturing processes are generally composed of various tasks that require different levels of worker qualifications (Efe, Okudan K., & Kurt, 2018). The classical ALBP assumes that every task has a determined processing time. In cases where a task requires different operation times depending on the qualification and experience levels of the workers, it must be considered the assignment of worker to specific tasks in the balancing problems. This problem is known as the ALWABP which comprises two interrelated decisions which are required to be resolved as effectively as possible to design productive assembly lines (Mutlu, Polat, & Supciller, 2013). These decisions must be reached simultaneously and are related to partition both the tasks and workers among the workstations. The ALWABP was initially introduced by Miralles et al. (2007). ALWABP considers an available workforce which states workers who have diverse abilities and who execute the tasks allocated to each of the stations (Villa & Pereira, 2014). Furthermore, within the scope of ALWABP, task processing times may vary from one worker to another, and some workers would not be eligible to execute every task (Michels & Costa, 2020).

ALWABP is composed of a set of workstations S, a set of tasks N, and set of workers H, with $|H|=|S|$. Each task i has an operation time p_{hi} depending to the worker h performing it. It is important to partition the tasks among the workstations, where a worker is assigned to, with the target of designing an assembly line as productive as possible which equivalent of minimizing the cycle time. ALWABP differentiates from the simple ALBP because of the worker depended processing times of tasks. Like the categorization of simple ALBP, ALWABP can be classified into two types according to the objective of the problem. ALWABP-1 aims to minimize the number of required workstations and ALWABP-2 minimizes the cycle time with the allocation of the limited workers to the tasks.

2.1.1 An illustrative example

In the ALWABP-2, each worker takes a different time to execute a task, according to their abilities. There is no relation between the times needed to execute a task by any two workers. For example, for the first task, the first worker performs it in 20 seconds while the second worker needs 14 seconds. Similarity, for the second task, the first worker performs it in 12 seconds while the second worker needs 24 seconds. In the real-world, a worker may be unable to execute some tasks. Therefore, the related processing times are set as infinite. Table 2.1 shows task processing times according to the workers for an example ALWABP instance with 6 tasks and 3 workers.

Table 2.1 Task processing times for the example ALWABP problem

	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Task 6**
Worker 1	4	4	3	1	1	6
Worker 2	∞	5	6	5	2	4
Worker 3	3	4	2	∞	3	∞

In the given example, it is seen that some workers can do some work in a shorter time. The given processing times are determined according to the expertise of the workers in each task. Expertise and processing times of workers are important parameters for calculating the shortest cycle time in the used solution procedure. Additionally, some tasks may depend on some others to be executed. In Table 2.2, the interdependencies of the tasks are shown. According to the Table 2.2, the tasks that need to be done sequentially are shown with the value 1 and the related order is arranged from row to column in that Table. Depending on the task priority order in Table 2.2, four possible task assignments are shown as in Figure 2.1.

Table 2.2 Precedence relations between tasks for the example ALWABP problem

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Task 1	0	1	1	0	0	0
Task 2		0	0	0	1	0
Task 3			0	1	1	0
Task 4				0	0	0
Task 5					0	1
Task 6						0

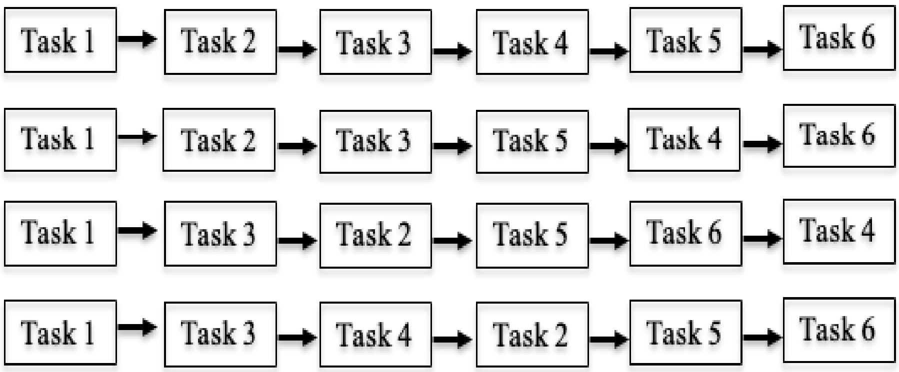

Figure 2.1 Four possible task sequences for the example ALWABP problem

The sequences given in Figure 2.1 are independent from the processing times and the worker abilities. Based on these possible sequences, it is aimed to identify a line balance so as to minimize the cycle time. Table 2.3 shows a possible line balance where each worker performs the same number of tasks for the illustrative example. According to the given line balance, worker 3 is assigned to station 1 while worker 1 is assigned to station 2 with the same workload (total workstation time) as 5, and worker 2 is assigned to station 3 with the workload as 6. As a result, the cycle time of the line is achieved as 6.

Table 2.3 An example line balance for the example ALWABP problem

Station 1		Station 2		Station 3	
Worker 3		Worker 1		Worker 2	
Task 1	3	Task 2	4	Task 5	2
Task 3	2	Task 4	1	Task 6	4
Total Time	5	Total Time	5	Total Time	6

2.1.2 Formal problem definition

The mathematical formulation (Miralles, Garcia-Sabater, Andres, & Cardos, 2008) of the ALWABP-2 is presented in this subsection. As stated previously, ALWABP-2 seeks the minimum cycle time for a definite number of workstations and comprises two interrelated decisions as partitioning both the tasks and workers among the workstations by considering worker eligibilities to perform tasks. The assumptions of the provided ALWABP-2 formulation are listed below.

- The serial paced assembly line is capable to produce a single product within m workstations.
- Worker dependent deterministic task processing times and the precedence relations between tasks are known.
- There are m workers (same as number of workstations) available and different task operation times based upon the worker performs the task.
- Each worker could be assigned to exactly one workstation.
- Each task could also be assigned to exactly one workstation.
- The eligibilities between the workers and tasks assigned to workstations must be satisfied.

The notations with their definitions for the ALWABP-2 formulation are presented in Table 2.4.

Table 2.4 Notations and definitions for ALWABP-2

Set and Indices	N	Total number of tasks to be partitioned among the available workstations
	H	Total number of available workers to do tasks on the assembly line
	S	Total number of available workstations on the assembly line
	i, j	Task
	h	Worker
	s	Workstation
Parameters	p_{hi}	Processing time of task i by worker h
	D_j	Set of immediate predecessors of task j with regard to precedence relations
Decision variables	C	Cycle time
	$x_{shi} \in \{0,1\}$	Equals to 1 if task i is performed by worker h in station s, and 0 otherwise
	$y_{sh} \in \{0,1\}$	Equals to 1 if worker h is assigned to station s, and 0 otherwise

With regard to given assumptions and notations, the ALWABP-2 has been formulized by Miralles, Garcia-Sabater, Andres, & Cardos, 2008. The objective function and the constraints of the constructed formulization are given below with their definitions.

$$Min\ z = C \tag{2.1}$$

$subject\ to$:

$$\sum_{h \in H} \sum_{s \in S} x_{shi} = 1 \qquad \forall_i \in N, \tag{2.2}$$

$$\sum_{s \in S} y_{sh} \leq 1 \qquad \forall_h \leq H, \tag{2.3}$$

$$\sum_{h \in H} y_{sh} \leq 1 \qquad \forall_s \in S, \tag{2.4}$$

$$\sum_{h \in H} \sum_{s \in S} s \cdot x_{shi} \leq \sum_{h \in H} \sum_{s \in S} s \cdot x_{shj} \qquad \forall_i;\ j/i \in D_j \tag{2.5}$$

$$\sum_{i \in N} p_{hi} \cdot x_{shi} \leq C \qquad \forall_h \in H;\ \forall_s \in S, \tag{2.6}$$

$$\sum_{i \in N} x_{shi} \leq M \cdot y_{sh} \qquad \forall_h \in H;\ \forall_s \in S, \tag{2.7}$$

$with$

$$y_{sh} \in \{0, 1\} \qquad \forall_s \in S;\ \forall_h \in H, \tag{2.8}$$

$$x_{shi} \in \{0, 1\} \qquad \forall_s \in S;\ \forall_h \in H;\ \forall_i \in N \tag{2.9}$$

$$M > \sum_{h \in H} \sum_{i \in N} p_{hi} \tag{2.10}$$

The cycle time is tried to be minimized trough the objective function (2.1). Constraint set (2.2) ensures to assign a task exactly one workstation to where only one worker assigned. Constraint sets (2.3) assigns each worker to strictly one workstation, while constraint set (2.4) satisfies that every workstation to have exactly one worker. Constraints set (2.5) guarantees not to violate the precedence relations between tasks. Constraint set (2.6) guarantees that the processing times of the tasks assigned to a workstation and worker pair not to exceed the given cycle time *C*. Constraint set (2.7) state that each worker assigned to a workstation may execute more than one task.

2.2 An Overview on ALWABP-2

In the literature, there are several studies related to ALWABP and its variations. The original version of the problem which is related to simultaneously assign tasks and workers to workstations considering cycle time minimization (ALWABP-2) was introduced by Miralles et al. (2007). They described the problem under the disabled worker assignment restriction as an optimization problem by means of the mathematical programming and tried to solve the problem to optimality through the developed model. Subsequent studies have been extended the problem by considering job rotation between operators (Costa & Miralles, 2009 and Cesar et al., 2013), by allowing parallel workstations and heterogeneous workers (Araujo et al., 2012, 2015) and by considering worker integration (Moreira et al., 2015). Another variant of the problem was studied by Ritt et al. (2016) under the indefinite worker availability restriction. The authors formulated this new variant of the problem as a two-level mixed integer programming (MIP) model and developed some local search approaches to tackle the problem. Pereira (2018) addressed the ALWABP with stochastic task times, and tried to solve the problem to optimality and approximately through the developed approaches. Liu et al. (2019) also extended the ALWABP under the stochastic worker availability restriction in such a way that to penalize the workers deficiency. The Authors formulated the problem as a stochastic programming model by relaxing some of the assumptions considered by Ritt et al. (2016) so as to minimize both the cycle time and penalty due to worker deficiency. Then, they adopted the sample average approximation (SAA) to solve the problem with different sized workers and workstations combinations.

Oksuz et al. (2017) studied the ALWABP problem for a different line configuration: ALWABP in the U-shaped assembly lines (ULWABP). The Authors formulated the problem under the U-shaped assembly line configuration as a nonlinear mathematical programming model. Afterwards, they designed two well-known metaheuristics – genetic and artificial bee colony algorithms – to identify optimal or near optimal solutions for the problem.

Akyol & Baykasoğlu (2019) presented a new type of ALWABP which considers ergonomic risks, called the ErgoALWABP problem. They designed a constructive heuristic which is composed of several rules to solve the problem. Yilmaz (2021) addressed the ALWABP that is extended by taking into consideration the sequence-dependent setup times between tasks. A MIP was developed, and a SA algorithm was proposed for the problem. In another study, Zacharia and Nearchou (2021) addressed another variant of the ALWABP-2 having fuzzy processing times and multiple objectives and they designed a multi-objective genetic algorithm to effectively tackle the problem.

Campana et al. (2021) studied the ALWABP with a multi-skilled workforce. This problem considers hierarchically ranked qualification levels among workers. They formulated a mathematical model with the aim to minimize the total cost, they proposed a variable neighbourhood descent heuristic (VNDH) for the large-size instances. Liu et al. (2021a) studied the ALWABP with the consideration of multi-skilled workers and energy consumption. They formulated a bi-objective mathematical model, and realized multi-objective variants of the genetic (NSGA-II) and simulated annealing (MOSA) algorithms to identify optimal or near-optimal solutions for the large-scale instances.

Lastly, Liu et al. (2021b) investigated a new risk averse ALWABP considering the uncertainty for available workforce. They aimed to minimize the summation of the cycle time and number of transient workers under risk averse with weight adjustment. They formulated a stochastic programming model having two phases and proposed a hybrid algorithm which is a combination of the genetic, K-means clustering and variable neighbourhood search algorithms (GAKV).

Within the context of this study, we focus on the ALWABP-2. In the literature, several solution techniques have been proposed for the considered problem. Chaves et al. (2007, 2009) generated ALWABP-2 instances based on the worker-dependent task times to be used as benchmark set. Miralles et al. (2007) provided the formal definition of the ALWABP through mathematical programming and designed a branch and bound (B&B) algorithm having several search strategies to explore the solution space of the problem. Vilà & Pereira (2014) proposed a station-oriented branch, bound, and remember (BB&R) algorithm to investigate the solution space of the problem for the optimal solutions. They obtained the optimal solutions for 319 out of 320 instances (Chaves et al. 2007). Yilmaz & Demir (2019) developed a new mathematical model for ALWABP-2. Moreover, several approximation algorithms (heuristics and metaheuristics) were designed to discover the solution space of the ALWABP-2. Chaves et al. (2009) presented a Clustering Search (CS) based hybrid solution approach including iterative local search (ILS). Blum & Miralles (2011) presented an iterated beam-search (IBS) algorithm to tackle the problem. Moreira et al. (2012) developed a constructive heuristic method depending on task and worker assignment considering priority rules for the problem. Moreira & Costa (2013) modelled a MIP and proposed hybrid heuristic algorithm. Mutlu et al. (2013) solved the problem via a modified genetic algorithm (MGA). Borba & Ritt (2014) presented an interval probabilistic beam search (PBS) algorithm to tackle with ALWABP-2 and, then hybridized it with the B&B algorithm. Polat et al. (2015) presented a two-phase VNS algorithm to tackle problem. In the first phase of the algorithm, they used the VNS to allocate tasks to workstations while in the second phase, they applied a VNDH for worker assignment to workstations. Ritt et al. (2015) modelled a two-stage MIP and proposed local search heuristics to solve the problem. Pereira (2018) proposed exact and heuristic methods with approximating regret evaluation.

Table 2.5 Solution techniques used in literature for Assembly Line Problems

Author(year)	**Characteristics**										**Solution method**
	Job rotation	Parallel workstations	Multi-skilled workers	U-shaped line	Ergonomic factors	Two-sided workstations	Setup time	Human factors	Uncertainty	Energy consumption	
Miralles et al. (2008)			√								B&B algorithm
Costa & Miralles (2009)	√		√								MILP/Heuristic
Chaves et al. (2009)			√								CS/ILS
Moreira & Costa (2009)			√								TS
Blum & Miralles (2011)			√								IBS
Moreira et al. (2012)			√								Constructive heuristic framework based on priority rules
Araujo et al. (2012, 2015)		√	√								GA/TS
Mutlu et al. (2013)			√								Iterative GA
Borba & Ritt (2014)			√								MIP/IPBS/BB
Villa & Pereira (2014)			√								BB&R
Polat et al. (2015)			√								Two-phase VNS
Ritt et al. (2015)			√						√		Two-stage MIP/Local search methods
Oksuz et al. (2017)			√	√							ABC/GA
Pereira (2018)			√						√		Exact/Heuristic, based on approximating regret evaluation
Akyol & Baykasoğlu (2019)			√		√						Constructive heuristic framework based on priority rules
Liu et al. (2019)			√						√		Stochastic programming/SAA
Campana et al. (2021)			√								VND
Liu et al. (2021a)			√							√	NSGA-II/MOSA
Liu et al. (2021b)									√		Stochastic programming/GAKV
Yilmaz et al. (2021)			√				√				MILP/SA
Zacharia & Nearchou (2021)			√						√		MOGA

According to Table 2.5, solution procedure developed to solve the ALWABP in the related literature generally focused on proposing heuristics and metaheuristics because of the NP-hard nature of the problem. On the other side, there is no study which has been proposed an IP-based heuristics for the problem. Therefore, this study proposes firstly an implementation of IP-based heuristics to the ALWABP-2. Our approach integrates IP-based FAO heuristics with SA.

CHAPTER THREE

FIX AND OPTIMIZE HEURISTIC

This chapter presents the FAO algorithm which is used as the solution procedure within the context of this thesis and provides a brief literature review on the FAO heuristic.

3.1 The Algorithm

The FAO is a MIP-based improvement heuristic that was developed independently and separately by Gintner et al. (2005) and Pochet & Wolsey (2006) named as "exchange heuristic". Subsequently, formal introduction of the FAO heuristic was done by Helber & Sahling (2010), who designed a FAO heuristic to solve multi-level capacitated lot sizing problem (CLSP). The FAO heuristic decomposes original problem into smaller sub-problems and improves the solutions by iteratively solving the sub-problems. A sub-problem is the reduced form of the original problem in which a set of decision variables are fixed in value while the remaining set are optimized (Gurkan & Tunc, 2021). The FAO heuristic requires an initial feasible solution to be generated to start its search on the solution space of a problem. This initial feasible solution is constructed by decomposing some of the decision variables and fixing them at their value. The remaining decision variables then optimized in accordance with the fixed variables through mixed-integer programming, integer programming etc. For the subsequent iterations, the FAO heuristic fixes different sets of decision variables in line with a decomposition scheme and optimizes the remaining decision variables. The FAO heuristic realizes the fix and optimize phases for each iteration sequentially until the stopping condition is satisfied. Within several iterations, the optimization procedure is executed with different sub-problems. Since the optimized variables are composed of a small group of variables of full model, each sub-problem could be solved quickly by a MIP solver. This process is repeated until all the sub-problems are solved. It is important to decide which and how many variables should be optimized.

For more information on the FAO algorithm, the reader can be referred to the study of Sahling et al. (2009).

Given a binary-integer linear program $Z: \min\{c^T y + d^T x: Ay + Bx \leq b, x \in [0,1], y \in [0,1]\}$ the FAO heuristic first fixes $\bar{y} \in [0,1]$, and then solves the subproblem $SP: \min\{d^T x: Bx \leq b - A\bar{y}, x \in [0,1]\}$. This is gradually and iteratively introduced into the IP.

Initial Solution	Worker assignment			Optimize Phase - Task assignment						
	Station 1	Station 2	Station 3	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Cycle Time
Worker 1	1	0	0	4		3				7
Worker 2	0	1	0		5			2		7
Worker 3	0	0	1				∞		∞	∞
Cycle Time	7	7	∞							

Iteration 1	Fix Phase - Worker assignment			Optimize Phase - Task assignment						
	Station 1	Station 2	Station 3	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Cycle Time
Worker 1	1	0	0	4		3				7
Worker 2	0	0	1					2	4	6
Worker 3	0	1	0		4		∞			∞
Cycle Time	7	∞	6							

Iteration 2	Fix Phase - Worker assignment			Optimize Phase - Task assignment						
	Station 1	Station 2	Station 3	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Cycle Time
Worker 1	0	1	0		4		1			5
Worker 2	0	0	1					2	4	6
Worker 3	1	0	0	3		2				5
Cycle Time	5	5	6							

Iteration 3	Fix Phase - Worker assignment			Optimize Phase - Task assignment						
	Station 1	Station 2	Station 3	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Cycle Time
Worker 1	0	0	1				1		6	7
Worker 2	0	1	0		5			2		7
Worker 3	1	0	0	3		2				5
Cycle Time	5	7	7							

Figure 3.1 Example of applying the FAO algorithm for subsection 2.1.1

We applied FAO heuristic to the illustrative example given in subsection 2.1.1. The algorithm is run for three iterations. The initial solution is constructed by assigning the workers to the stations sequentially, however it gives an infeasible solution with this assignment. In the first iteration, worker assignment is modified via swap operator and fixed its value. This value is given to sub-problem, which is task assignment problem, as an input and the sub-problem is optimized. In each iteration, this process is repeated. In the considered example, the best solution is found in the second iteration. The aim of this implementation is to reduce the solution times obtained by original mathematical model with FAO heuristic. The mathematical model requires to seek whole search space and try each possibility. However, FAO heuristic provides the optimum solution with the shorter time by seeking smaller search space reducing the number of decision variables.

3.2 An Overview on the Fix and Optimize Heuristic

The FAO heuristic has promising performance since it finds a near optimal solution to a MIP formulation (Gürkan & Tunç, 2021). In the last decade years, the heuristic has become popular in the literature due to it can be easily applicable and it has been applied several optimization problems such as production planning (Gürkan & Tunç, 2021), lot-sizing (Carvalho & Nascimento, 2022), and scheduling problems (Gintner et al., 2005). An overview of the related literature of FAO according to implementation area and search space diversification mechanism in “fix” phase is presented in Table 3.1.

Gintner et al. (2005) designed a FAO algorithm on the basis of the on the time-space network approach to discover the solution space of the multi-depot scheduling problem with multiple types of vehicles. The Authors firstly identified chains of trips which are served with the same vehicle for each solution. Fixing these chains provided the Authors to be able to solve smaller problems to optimality. Therefore, they developed a heuristic to determine stable chains for fixing procedure. They were able to find near-optimal solutions for the large-size problem instances.

Federgruen et al. (2007) developed two progressive interval heuristics based on FAO approach, called strict partitioning heuristics (SP) and expanding horizon heuristics (EH), for the dynamic CLSP. In another study, Sahling et al. (2009) presented a FAO heuristic for the dynamic multi-level CLSP with setup carry-overs. For each iteration, new solution is accepted according to the improvement of the fitness value and finding a feasible solution considering the limited period time. The proposed algorithm provided high-quality results.

Helber & Sahling (2010) presented an optimization-based methodology to solve the dynamic multi-level CLSP with lead times. They solved a series of MIP in an iterative FAO algorithm. An approach as mentioned above has been applied in the algorithm.

James & Almada-Lobo (2011) presented iterative neighbourhood search heuristics based on a MIP formulation for the single and parallel machine CLSPs having sequence-dependent setup times under the machine eligibility constraints. They proposed a time-based FAO heuristic, and to obtain improved feasible solutions local and neighbourhood search heuristics were developed (Xiao, Zhang, & Gupta, 2013).

Goren et al. (2012) developed a hybrid method combines GA with FAO heuristic to tackle the single-level CLSP with setup carryover.

Guimaraes et al. (2013) presented a MIP-based approach to solve the single machine CLSP with sequence-dependent setup times and costs. FAO heuristic is used to as an improvement heuristic to escape from local minima by using re-optimizing the solution iteratively.

Seeanner et al. (2013) presented a heuristic approach which combines the rules of the variable neighbourhood decomposition search (VNDS) and FAO for solving general lot-sizing and scheduling problem for multiple production stages. They used neighbourhood structures for *'fix'* part of the algorithm.

In another study, Xiao et al. (2013) handled with the integrated parallel machine lot sizing and scheduling problem considering sequence-dependent setup times, time windows, machine eligibility and preference constraints. They presented two MIP-based FAO algorithms for solving the problem in which the machine assignment variables are fixed by randomised least flexible machine (RLFM) rule.

For another application field, Ghaderi & Jabalameli (2013) developed a FAO algorithm based on SA and exact methods for solving the budget-constrained dynamic (multi-period) uncapacitated facility location–network design problem. They fixed the variables using SA. On the other hand, Rahmaniani & Ghaderi (2013) proposed a FAO algorithm based on the evolutionary firefly algorithm (FA) to tackle the facility location/network design problem.

Toledo et al. (2013) studied a multi-level CLSP with backlogging. They proposed a hybrid approach which combines multi-population-based metaheuristic using FAO algorithm and mathematical programming techniques. FAO fixes the value for all binary variables according to two types of windows; certain number of period and family products. A sub-problem with binary variables related to window only is solved and all other binary variables act as fixed parameters.

Dorneles et al. (2014) presented a MIP-based FAO heuristic hybridised with a VND approach to solve the high school timetabling problem. They decomposed the main problem into three type single problem including class, teacher, and day. They showed that the proposed FAO heuristic outperformed state-of-the-art solution methodologies for the solution of the problem.

Sel & Bilgen (2014) addressed a production and distribution planning problem in the soft drink industry. They proposed a hybrid method combining simulation model and MIP-based FAO heuristic.

Chen (2015) proposed a FAO approach for two dynamic multi-level CLSP; one with setup carryover and the other one without setup carryover. They also developed FAO based VNS algorithm to solve the multi-level CLSP without setup carryover. They applied the algorithm integrated in a local search.

Helber et al. (2016) developed a hierarchical modelling system and a decomposition algorithm based on FAO heuristic to solve facility layout planning problem.

Moreno et al. (2016) tackled the facility location and transportation problem considering fleet decisions and uncertainty. They presented two MIP-based heuristics, relax-and-fix and FAO, to provide good-quality solutions.

Furlan & Santos (2017) presented a hybrid approach based on the bees algorithm combined with the FAO heuristic to tackle the multi-level CSLP. They used the bees algorithm to realize random search in FAO and thus avoid from the local optima.

Wei et al. (2017) studied integrated production and distribution planning problem. The FAO improvement approach embedding a VNS was presented. In this study, the VNS acts as a diversification process that systematically explores the neighbourhood of current solution and searches the different region of the solution space.

Guo et al. (2018) proposed a FAO heuristic to solve crane scheduling problem. They used several decomposition operators to find the set of free variables. The remaining variables was set to be fixed. This idea is assumed from adaptive large neighbourhood search algorithm (Ropke & Pisinger, 2006).

Dastjerd & Ertogral, (2019) tackled an integrated fleet sizing and replenishment planning problem. They proposed two versions of a FAO heuristic; with and without moving window. The results showed the effectiveness of these heuristics on a set of problems.

Tanksale & Jha (2020) addressed a capacitated multi-item inventory-transportation problem. They hybridized the FAO heuristic with two metaheuristics; GA and iterated local search (ILS) individually, as a diversification mechanism.

Turhan & Bilgen (2020) hybridized FAO and SA approaches for nurse rostering problem. They used the FAO heuristic to diversify the search space and obtain good solutions.

Gurkan & Tunc (2021) addressed the stochastic multi-item CLSP solving by a FAO heuristic based on an item-period oriented decomposition scheme.

Carvalho & Nascimento (2022) studied the integrated lot sizing and scheduling problem considering capacitated non-identical parallel machines, sequence-dependent setup costs, setup carryover and non-triangular times. They solved the problem by proposing a combination of three different heuristics that are relax-and-fix and FAO heuristics with the path-relinking (PR) and kernel search (KS) methods.

Friske et al. (2022) proposed a hybridization of relax-and-fix and FAO mat-heuristics for solving a maritime inventory routing problem. They used the FAO approach as an improvement heuristic.

Pourrezaie-Khaligh et al. (2022) addressed a healthcare facility location/network design problem with the aim of minimizing the inequality and maximizing accessibility among all demand nodes. They proposed a FAO approach based on the enhanced tabu search algorithm (ETSA) for handling the large-scale problems. The ETSA is iteratively used to fix the value of binary variables, and the subproblem was solved by CPLEX solver to find the optimal solution at each iteration.

Table 3.1 Summary of works using FAO mat-heuristics as solution method

Author(year)	**Diversification Mechanism**	**Application Area**
James & Almada-Lobo (2011)	Iterative neighborhood search procedure	Lot-sizing and scheduling
Seeanner et al. (2013)	VNDS	
Xiao et al. (2013)	RLFM rule	
Guimaraes et al. (2013)	Iterative heuristics	
Carvalho & Nascimento (2022)	PR-KS heuristics	
Federgruen et al. (2007)	Progressive interval heuristics	Production planning and dynamic lot sizing
Sahling et al. (2009)	Improvement procedure	
Helber & Sahling (2010)	Improvement procedure	
Goren et al. (2012)	GA procedure	
Toledo et al. (2013)	Multi-population GA	
Chen (2015)	VNS	
Furlan & Santos (2017)	Bees Algorithm	
Gürkan & Tunç (2021)	Decomposition schemes	
Ghaderi & Jabalameli (2013)	SA	Facility location problem
Rahmaniani & Ghaderi (2013)	FA	
Moreno et al. (2016)	Improvement procedure	
Pourrezaie-Khaligh et sl. (2022)	ETSA	
Gintner et al. (2005)	Stable chain algorithm	Multiple-depot-vehicle-type scheduling problem
Dorneles et al. (2014)	VNDS	High school time-tabling problem
Sel & Bilgen (2014)	Simulation model	Production and distribution planning
Helber et al. (2016)	Improvement procedure	Facility layout planning
Wei et al. (2017)	VNS	Production and distribution planning
Guo et al. (2018)	Decomposition operators	Gantry crane scheduling
Dastjerd & Ertogral (2019)	Improvement procedure	Integrated replenishment planning and fleet sizing
Tanksale & Jha (2020)	GA/ILS	Integrated inventory-transportation problem
Turhan & Bilgen (2020)	SA	Nurse rostering problem
Friske et al. (2022)	Improvement procedure	Maritime inventory routing problem

According to the Table 3.1, we can conclude that the FAO heuristics has been widely applied to the lot sizing and scheduling problems. Moreover, there are few studies which consider other type of optimization problems such as facility location, maritime inventory routing, production-distribution, and timetabling problems. These studies have improved the performance of the basic FAO heuristic in a manner of quality. Unfortunately, the basic FAO search procedure tends to get trapped into a local optimum due to the lack of having an efficient search space diversification mechanism. Therefore, in many studies, FAO heuristic is usually hybridized with other metaheuristics. In our study, we present a hybrid approach combining FAO and SA methodologies. As far as we know, this current thesis is the first research attempt to solve ALWABP-2 via a FOA heuristic in the literature. As mentioned previous sections, ALWABP-2 considers two different assignment decisions must be reached simultaneously. The difficulty in solving the handled problem mainly springs from the binary decision variables in use. Therefore, the FAO approach can be a promising methodology as it aims to find near optimal solution to an IP formulation by decomposing and solving sub-problems iteratively. ALWABP-2 is suitable to be divided into two sub-problems; task assignment and worker assignment. This property indicates that the FAO can be successful in solving ALWABP-2, since fixing binary variables to their values would reduce complexity and directly impact on the performance of the algorithm.

CHAPTER FOUR

A FIX AND OPTIMIZE ALGORITHM FOR THE ALWABP-2

In this chapter, the proposed solution methodology is presented in detail. This section explains the structure of the algorithm, the performance evaluation of the proposed solution methodology on benchmark sets, and the computational results. The proposed fix and optimize algorithm is explained in the next section detailed. Afterwards, results of the performance evaluation tests of the developed solution procedure on a standard benchmark set are given.

4.1 Proposed Fix and Optimize Algorithm

This study mainly is about to develop a hybrid solution algorithm for the ALWABP-2. The developed algorithm is a combination of an IP-based FAO heuristics and SA. Figure 4.1 illustrates the main steps of the proposed hybrid FAO algorithm. The proposed algorithm has mainly two steps as fix (SA) and optimize (MIP solver), which are executed hierarchically in each iteration. In the first step, which will be detailed in the following, the proposed algorithm determines which binary variables will be fixed by means of the SA. In each iteration, the proposed algorithm executes the MIP solver to find an optimal solution for the remaining decision variables in accordance with the fixed ones as the second step. At the beginning of the algorithm, the worker-workstation assignment decision is constructed randomly. Then, by solving the mathematical model, it is obtained the worker-work assignment as a current solution. In the second step of the algorithm, the FAO algorithm is applied to obtain new worker assignment by performing swap operator. Then the new solution is put it to an acceptance criterion approval. The solution that has a better objective value is accepted as a current solution, otherwise SA algorithm’s acceptance rule is applied to decide whether worse solution will be accepted or not. In continuation, the partial solution of the fixed variables is set as a parameter ($y_{sh} = \overline{y_{sh}}$) for the execution of the IP which optimizes remaining binary variables (x_{shi}). The pseudocode of the proposed algorithm is presented in Figure 4.2.

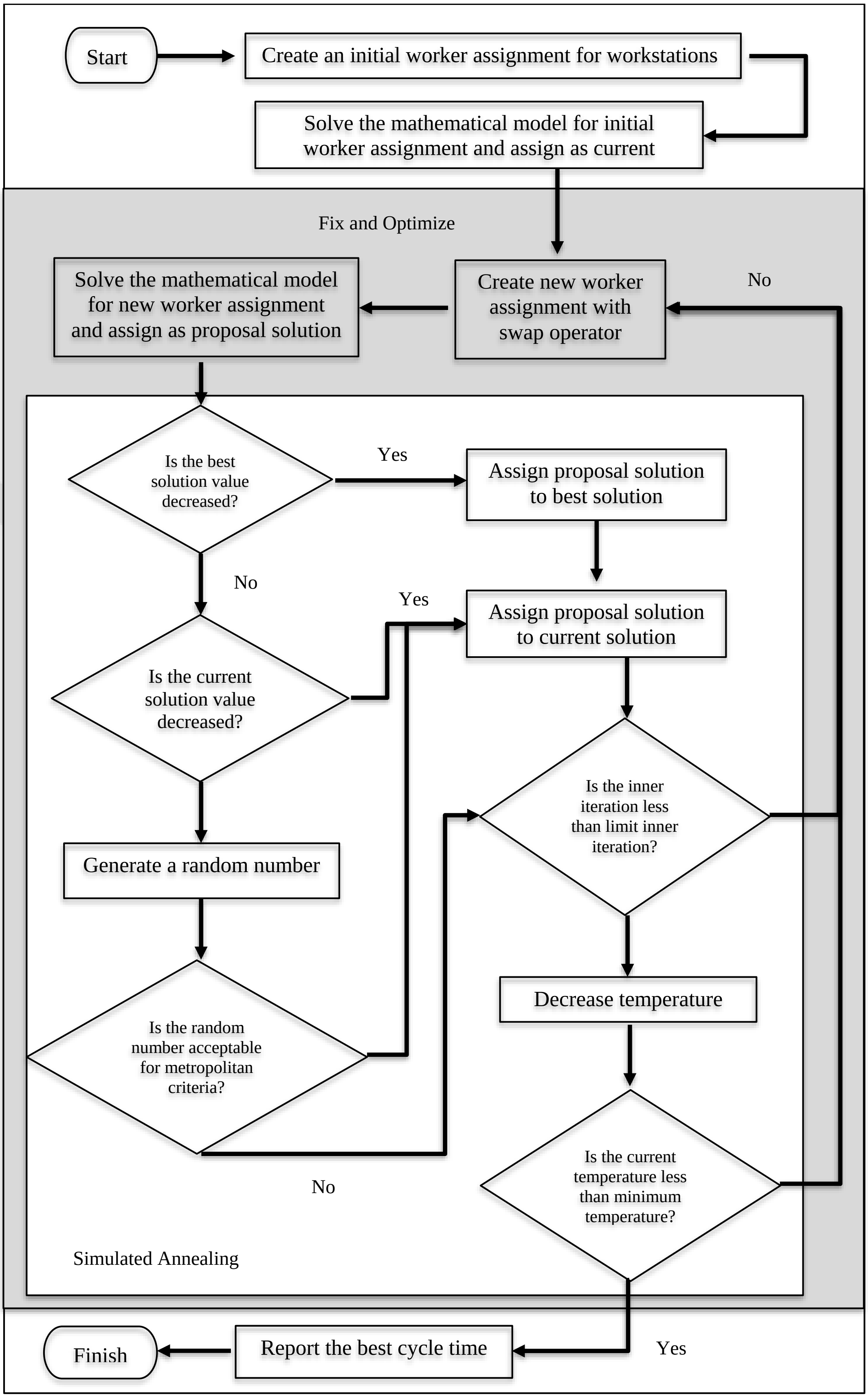

Figure 4.1 Main steps of the proposed hybrid FAO algorithm

```
initialize T_0, T_min, β, y_sh^init;
T = T_0;
y_sh = y_sh^init;
C ← solveIP(y_sh);
y*_sh = y_sh;
C* = C;
while T_min < T
        while innerIter < innerMaxIter
                overline(y_sh) = swap(y_sh);
                C̄ = solveIP(overline(y_sh));
                if status == infeasbible then
                        C̄ = infinite;
                end if
                if C̄ < C*
                        C* = C̄;
                        y*_sh = overline(y_sh);
                end if
                if C̄ < C
                        C = C̄;
                        y_sh = overline(y_sh);
                else if ϵ^((C−C̄)/T) > rand(0,1)
                        C = C̄;
                        y_sh = overline(y_sh);
                end if
                innerIter = innerIter + 1;
        end while
        T = T · β;
end while
```

Figure 4.2 Pseudocode of the proposed hybrid FAO algorithm

4.1.1 The initialization phase

Firstly, the parameters defined on the problem set assign with the proposed solution algorithm. These parameters include number of workers, number of tasks, number of stations and processing times. All the parameters are determined by the benchmark sets. The SA algorithm's parameters are initial temperature and cooling rate. The initial solution is generated randomly. Before running the mathematical model, we create an initial worker assignment for workstations as an input parameter. Then the mathematical model gives task-workstation assignment as a current solution.

4.1.2 Fix phase

The part of the hybrid algorithm is the FAO heuristics which improves the initial solution. First, the value of y_{sh} variables is fixed to the obtained value in the initial solution. The SA algorithm searches the feasible space. The process starts from the neighbourhood of the current solution and lasts when a satisfactory solution is obtained or the termination criteria is achieved. A neighbourhood structure i.e., swap operator, is applied for worker-workstation assignment and fixed to their current values. *2-exchange* swap is used to shift two randomly selected workers.

Hence, the SA is used to discover on the fixed variables which are y_{sh}. This procedure includes two parameters: the current temperature (T) and the cooling rate (β). (T_0) is the initial temperature and decreases with formula ($T * \beta$) using the cooling rate (β) at each iteration. (y_{sh}^{init}) is set according to the initial solution given the current solution ($\overline{y_{sh}}$) having an objective value ($\bar{C}$) and a new neighbour solution (y_{sh}) is accepted if ($\varepsilon^{\frac{(C-\bar{C})}{T}}$) is greater than or equal to (*rand(0,1)*). This feature of the SA allows the search to expand towards to the global optima. The termination criteria is controlled with the final temperature (T_{min}) is reached.

4.1.3 Optimize phase

The mathematical formulation of the ALWABP-2 contains a set of binary decision variables which are named as binary assignment variables. In the optimize phase of the proposed algorithm, IP model optimizes a subset of these binary assignment variables, namely *'free'* variables. The remaining set of binary assignment variables are temporarily fixed to their values assigned in previous iterations. In line with the free and fixed sets of binary variables, we reformulate the original model by projecting variable y_{sh} out of formulation (1)-(7) yielding the following sub-problem (SP) that only models the assignment problem of the works.

SP: Objective function (1)

subject to

Constraints (2), (5)-(7) with $y_{sh} = \overline{y_{sh}}$

The original model is formulated by using two sets of assignment (x_{shi}, y_{sh}) variables. The model, the workers assigned to the workstations solved jointly with determining the task performing within these workers. In the proposed algorithm, the workers are assigned to the workstations, which leaves the SP to determine the tasks within each worker are executed. Therefore, the constraint sets (3)-(4) are not necessary. In other words, the algorithm first matches workers with workstations representing *fix* phase and then determines the task assignments for each worker representing *optimize* phase. Then mathematical model optimizes the assigned works to the workers. In each iteration, most of the binary assignment variables y_{sh} are set to a fixed value $\overline{y_{sh}}$. This reduces the number of binary variables to be decided in the problem of the current iteration. It provides the shorter solution time as the number of binary decision variables of the sub-problem is smaller than in the original problem.

4.2 Computational Study

The proposed solution methodology was implemented in C++ and compiled with the GUROBI compiler and run on an Intel core i7, 2.8 GHz processor and 16 GB RAM. To assess the algorithm performance, it is used to ALWABP benchmark data set proposed by Chaves et al. (2007).

4.2.1 Parameter tuning

This section describes the parameters used for obtaining the result. All values are based on preliminary tests. As mentioned in previous sections, SA algorithm is proposed as an acceptance criterion. For the parameters, the starting temperature, and the cooling rate β is set to 100 and 0.99975, respectively. Finally, the number of inner iterations is set to 5.

4.2.2 Benchmark instances

The benchmark set consists of 320 instances from the benchmark data set (Chaves et al., 2007). These instances are comprised of four families: Roszieg, Heskia, Tonge and Wee-Mag. Each family has 80 instances in total including 32 task groups with 10 instances. Each task group is defined by a number between 1 and 8. The characteristics for each family group are composed of the number of tasks (|N|) and the number of workers (|H|).

In this study, we solve all instances for the Roszieg and Heskia families (each 80 instances) and first four groups for Tonge and Wee-Mag families (each 40 instances). In summary, we run 240 instances in total. The details of data set characteristics used our study are given in Table 4.1.

Table 4.1 Data set (Chaves et al., 2009)

Family	|N|	|H|
Roszieg	25	4 (group 1-4) or 6 (group 5-8)
Heskia	28	4 (group 1-4) or 7 (group 5-8)
Tonge	70	10 (group 1-4)
Wee-Mag	75	11 (group 1-4)

The proposed algorithm is compared against the best solutions of seven methods given as ILS/CS (Chaves et al., 2009) TS (Moreira & Costa, 2009), IBS (Blum & Miralles, 2011), HGA (Moreira et al., 2012), IGA (Mutluet al., 2013), IPBS/B&B (Borba & Ritt, 2014), BB&R (Villa & Pereira, 2014) and MRBCRS (Akyol & Baykasoğlu, 2019) for each benchmark instance. The computational experiments consider the best results in the manner of quality for comparison.

The individual results given in appendices consider the average GAP for comparison. The average GAP between the best solution obtained by the proposed method and the best-known solution in the relevant literature calculated via Equation (4.1) as below

$$GAP = \frac{(best - BKS)}{BKS} \times 100\% \tag{4.1}$$

where the *best* is best solution obtained by the proposed method, and the BKS is the best-known solution (BKS) reported in the literature.

4.2.3 Computational results

The proposed algorithm is compared against previously developed methods in the related literature. Tables 4.2 and 4.3 reports the results of the comparison, in terms of quality, between the proposed algorithm and the seven methods from the literature for the instances generated by Chaves et al. (2009). Additionally, we report the detailed comparison results with Pereira (2014) and Blum (2011) in appendices as other authors are not reported in their studies.

Table 4.2 presents the results for 160 problem instances of Roszieg and Heskia families and Table 4.3 presents the results for 80 problem instances of Tonge and Wee-Mag families in total. The first column shows the benchmark data family, while the second column gives the group number of the related family. Since each family group includes 10 instances, therefore, the average results of 10 instances for each group of each family were given. The third column gives the optimum results obtained in the related literature. The rest of the columns give the solution methodologies and results. We report the best solution obtained by each of the algorithms.

Table 4.2 Results for the Roszieg and Heskia families

Family	Gr.	FAO	CS	ILS	TS	IBS	IGA	HGA	IPBS	BB	BB&R 60s	BB&R 600s	BB&R no time limit
Roszieg	1	20.10	20.10	20.10	20.10	20.10	20.10	20.10	20.10	20.10	20.10	20.10	20.10
	2	31.50	31.50	32.40	31.50	31.50	31.50	31.50	31.50	31.50	31.50	31.50	31.50
	3	28.10	28.10	28.20	28.10	28.10	28.10	28.10	28.10	28.10	28.10	28.10	28.10
	4	28.00	28.00	28.00	28.00	28.00	28.00	28.00	28.00	28.00	28.00	28.00	28.00
	5	9.70	9.70	10.30	9.70	9.70	9.70	9.70	9.70	9.70	9.70	9.70	9.70
	6	11.00	11.00	11.50	11.00	11.00	11.00	11.10	11.00	11.00	11.00	11.00	11.00
	7	16.00	16.00	16.50	16.00	16.00	16.00	16.00	16.00	16.00	16.00	16.00	16.00
	8	15.10	15.10	15.30	15.10	15.10	15.10	15.10	15.10	15.10	15.10	15.10	15.10
	Avg.	19.90	19.94	20.29	19.90	19.90	19.90	20.00	19.90	19.90	19.90	19.90	19.90
Heskia	1	102.30	102.30	102.30	102.3	102.30	102.30	102.30	102.30	102.30	102.30	102.30	102.30
	2	122.60	122.60	122.70	122.6	122.60	122.60	122.70	122.60	122.60	122.60	12..60	122.60
	3	172.50	172.50	172.60	172.5	172.50	172.50	172.50	172.50	172.50	172.50	172.50	172.50
	4	171.20	171.20	171.30	171.2	171.20	171.20	171.20	171.20	171.20	171.20	171.20	171.20
	5	34.90	34.90	35.30	35.0	34.90	34.90	34.90	34.90	34.90	34.90	34.90	34.90
	6	42.60	42.60	43.60	43.1	42.60	42.60	42.60	42.70	42.70	42.70	42.70	42.70
	7	75.20	75.20	76.70	75.2	75.20	75.20	75.20	75.20	75.20	75.20	75.20	75.20
	8	67.20	67.20	68.10	67.3	67.20	67.20	67.20	67.20	67.20	67.20	67.20	67.20
	Avg.	98.60	98.56	99.08	98.7	98.60	98.60	98.60	98.60	98.60	98.60	98.60	98.60

According to the Table 4.2, we obtain the best results for both Roszieg and Heskia families as the state-of-the-art algorithms in the related literature. For the Roszieg family, the proposed FAO algorithm is able to find the optimum results for each instance. Our algorithm has the best average result, which is 19.90, same as the result found by other algorithms (TS, IBS, IGA, IPBS, BB and BB&R), followed by CS with average result of 19.94, ILS with 20.29 and HGA with 20.00. In similar, for the Heskia family, the proposed FAO algorithm has the optimum result which is 98.60 (same as CS, IBS, IGA, HGA, IPBS, BB and BB&R algorithms), followed by TS with the result of 98.70 while the result obtained by ILS is 99.08 on average. The average results indicate that our algorithm is an efficient solution method for solving small-size problem instances.

Table 4.3 Results for the Tonge and Wee-Mag families

Family	Gr.	FAO	CS	ILS	TS	IBS	HGA	IGA	IPBS	BB	BB&R 60s	BB&R 600s	BB&R no time limit
Tonge	1	93.50	96.70	120.00	100.10	94.90	92.80	93.00	91.30	90.60	90.60	90.60	90.60
	2	116.40	116.00	151.80	117.70	110.20	109.30	109.30	107.80	106.70	106.70	106.70	106.70
	3	167.80	167.70	214.60	171.50	165.00	162.20	162.40	160.80	159.30	160.30	159.30	159.30
	4	174.20	174.00	220.70	178.30	170.00	168.40	168.40	165.90	163.90	165.90	163.90	163.90
	Avg.	137.97	138.60	176.78	141.90	135.03	133.18	133.28	131.45	130.13	130.88	130.13	130.13
Wee-Mag	1	27.50	29.00	31.20	35.30	28.70	26.70	26.70	27.10	26.80	26.60	26.10	26.10
	2	33.00	34.60	37.40	41.10	33.60	32.20	32.30	32.10	32.20	31.70	31.20	31.20
	3	48.30	50.80	54.30	58.40	50.10	47.60	47.60	47.50	47.50	46.60	46.10	45.80
	4	46.80	49.60	52.70	56.10	48.60	45.60	45.80	45.40	45.20	44.80	44.30	44.30
	Avg.	38.90	41.00	43.90	47.73	40.25	38.03	38.10	38.03	37.93	39.33	36.85	36.85

Table 4.3 gives the results for both Tonge and Wee-Mag families. We only run the algorithm for first 4 groups of each. And the algorithm is able to find near optimal solution effectively for these instances. For the Tonge family, BB and BB&R algorithms find the best-known result which is 130.13, followed by IPBS with the result of 131.45, HGA with 133.18, IGA with 133.28, IBS with 135.03, behind proposed FAO with 137.97, CS with 138.60, TS with 141.90 while the result obtained by ILS is 176.78 on average.

In similar, for the Wee-Mag family, BB&R algorithm has the best results on average with 36.85, followed by BB with 37.93, HGA and IBS algorithms with 38.03, IGA with 38.10, while the proposed FAO has an average result with 38.90, IBS with 40.25, CS with 41.00, ILS with 43.90 and TS with 47.73, respectively.

In summary, the results show that our algorithm is a promising solution methodology. Therefore, we can conclude that the results show the efficiency of decomposition-based methodology for the ALWABP and indicate that the algorithm can also be more capable for larger instances if more effective heuristics were to be developed.

CHAPTER FIVE

CONCLUSION

In this study, we developed a mat-heuristic algorithm for the ALWABP-2 problem which combines IP-based FAO heuristic and SA algorithm. The ALWABP-2 problem contains two assignment problems which are worker and task assignments to the workstations. By utilizing this structure, we developed a FAO algorithm, which solves the task assignment as a main problem and the worker assignment as a slave problem. We hybridized the FAO with the SA algorithm to increase the diversification intensification ability of the algorithm.

In the related literature, there is lack of efficient exact solution methods for the considered problem since it is a NP-hard which could not be solved by mathematical programming models efficiently. The proposed algorithm in this paper, which was the first implementation of FAO algorithm for the ALWABP literature. We implemented the proposed algorithm on the ALWABP benchmark set. Our numerical results demonstrated that the proposed algorithm is capable to solve small instances to optimality and has promising solutions for instances with more than 70 tasks. Additionally, the results confirm the comparable performance of the proposed algorithm with the state-of-the-art solution techniques in terms of solution quality.

The proposed algorithm can be improved by developing new neighbourhood structures in the fix phase to increase the competence of the algorithm.

REFERENCES

Akyol, Ş. D., & Baykasoğlu, A. (2019). ErgoALWABP: a multiple-rule based constructive randomized search algorithm for solving assembly line worker assignment and balancing problem under ergonomic risk factors. *Journal of Intelligence Manufacturing, 30*, 291-302.

Araujo, F. B., Costa, A. M., & Miralles, C. (2012). Two extensions for the ALWABP: Parallel stations and colloborative approach. *Int. J. Production Economics, 140*, 483-495.

Araujo, F. B., Costa, A. M., & Miralles, C. (2015). Balancing parallel assembly lines with disabled workers. *European J. Industrial Engineering, 9*(3), 344-365.

Blum, C., & Miralles, C. (2011). On solving the assembly line worker assignment and balancing problem via beam search. *Computers & Operations Research, 38*, 328-339.

Borba, L., & Ritt, M. (2014). A heuristic and a branch-and-bound algorithm for the assembly line worker assignment and balancing problem. *Computers and Operations Research, 45*, 87-96.

Campana, N. P., Lori, M., & Moreira, M. C. (2021). Mathematical models and heuristic methods for the assembly line balancing problem with hierarchical worker assignment. *International Journal of Production Research*, 2193-2211.

Carvalho, D. M., & Nascimento, M. C. (2022). Hybrid matheuristics to solve the integrated lot sizing and scheduling problem on parallel machines with sequence-dependent and non-triangular setup. *European Joural of Operational Research*, *296*, 158-173.

Cesar, M., Moreira, M. C., & Costa, A. M. (2013). Hybrid heuristics for planning job rotation schedules in assembly lines with heterogeneous workers. *International Journal of Production Economics*, *141*(2), 552-560.

Chaves, A. A., Lorena, L. A., & Miralles, C. (2007). Clustering search approach for the assembly line worker assignment and balancing problem. *In: Proceedings of the 37th international conference on computers and industrial engineering.*, 1469-1478.

Chaves, A. A., Lorena, L. A., & Miralles, C. (2009). Hybrid Metaheuristic for the Assembly Line Worker Assignment and Balancing Problem. *International Workshop on Hybrid Metahuristics. 5818.* Lecture Notes in Computer Science.

Chen, H. (2015). Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, *56*, 25-36.

Costa, A. M., & Miralles, C. (2009). Job rotation in assembly lines employing disabled workers. *International Journal of Production Economics, 120*(2), 625-632.

Cunningham, J. A. (1980). A. Management: Using the learning curve as a management tool: The learning curve can help in preparing cost reduction programs, pricing forecasts, and product development goals. *Spectrum IEEE, 17*(6), 45-48.

Dastjerd, N. K., & Ertogral, K. (2019). A fix-and-optimize heuristic for the integrated fleet sizing and replenishment planning problem with predetermined delivery frequencies. *Computers & Industrial Engineering*, *127*, 778-787.

Dorneles, A. P., De Araujo, O. C., & Buriol, L. S. (2014). A fix-and-optimize heuristic for the high school timetabling problem. *Computers & Operations Research, 52*, 29-38.

Efe, B., Okudan K., G. E., & Kurt, M. (2018). Age and gender-based workload constraint for assembly line worker assignment band balancing problem in a textile firm. *International Journal of Industrial Engineering, 25(1),* 1-17.

Federgruen, A., Meissner, J., & Tzur, M. (2007). Progressive Interval Heuristics for Multi-Item Capacitated Lot-Sizing Problem. *Operations Research, 55*(3), 490-502.

Friske, M. W., Buriol, L. S., & Camponogara, E. (2022). A relax-and-fix and fix-and-optimize algorithm for a Maritime Inventory Routing Problem. *Computers and Operations Research*, 137-153.

Furlan, M. M., & Santos, M. O. (2017). BFO: a hybrid bees algorithm for the multi-level capacitated lot-sizing problem. *Journal of Intelligence Manufacture*, 929-944.

Ghaderi, A., & Jabalameli, M. S. (2013). Modeling the budget-constrained dynamic uncapacitated facility location–network design problem and solving it via two efficient heuristics: A case study of health care. *Mathematical and Computer Modelling, 57*, 382-400.

Gintner, V., Kliewer, N., & Suhl, L. (2005). Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice. *OR Spectrum, 27*, 507-523.

Goren, H. G., Tunali, S., & Jans, R. (2011). A hybrid approach for the capacitated lot sizing problem with setup carryover. *International Journal of Production Research*, 1582-1597.

Guimaraes, L., Klabjan, D., & Almada-Lobo, B. (2013). Pricing, relaxing and fixing under lot sizing and scheduling. *European Journal of Operations Research*, 399-411.

Guo, P., Cheng, W., Wang, Y., & Boysen, N. (2018). Gantry crane scheduling in intermodal rail-road container terminals. *International Journal of Production Research*, 5419-5436.

Gürkan, M. E., & Tunç, H. (2021). A fix-and-optimize heuristic for the capacited multi-item stochastic lot-sizing problem. *An International Journal of Optimization and Control: Theories & Applications*, *11*(1), 41-51.

Helber, S., & Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *Int. J. Production Economics*, *123*, 247-256.

Helber, S., Böhme, D., Oucherif, F., Lagershausen, S., & Kasper, S. (2015). A hierarchical facility layout planning approach for large and complex hospitals. *Flex Service Manufacture Journal*, *28*, 5-29.

Liu, M., Liu, R., & Chu, F. (2019). An improved model for assembly line worker assignment and balancing problem considering stochastic worker availability. *International Conference on Industrial Engineering and Systems Management, 731-736*.

Liu, R., Liu, M., Chu, F., Zheng, F., & Chu, C. (2021a). Eco-friendly multi-skilled worker assignment and assembly line balancing problem. *Computers & Industrial Engineering, 151*.

Liu, M., Liu, Z., Chu, F., Liu, R., Zheng, F., & Chu, C. (2021b). Risk-averse assembly line worker assignment and balancing problem with limited temporary workers and moving workers. *International Journal of Production Research*.

Michels, A. S., & Costa, A. M. (2020). A note to: A multiple-rule based constructiv randomized search algortihm for solving assembly line worker assingment and balancing problem. *Journal of Intelligent Manufacturing, 32*, 2121-2124.

Miralles, C., & et al. (2007). Advantages of assembly lines in Sheltered Work Centres for Disabled. A case study. *International Journal of production Economics, 110*(1-2), 187-197.

Miralles, C., Garcia-Sabater, J. P., Andres, C., & Cardos, 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*, 352-367.

Moreira, M. C., & Costa, A. M. (2013). Hybrid heuristics for planning job rotation schedules in assembly lines with heterogeneous workers. *International Journal of Production Economics, 141*(2), 552-560.

Moreira, M. C., Miralles, C., & Costa, A. M. (2015). Model and heuristics for the assembly line worker integration and balancing problem. *Computers and Operations Research, 54*, 64-73.

Moreira, M. C., Ritt, M., Costa, A. M., & Chaves, A. A. (2012). Simple heuristics for the assembly line worker assignment and balancing problem. *J Heuristics, 18*, 505-524.

Moreira, M. C., & Costa, A. M. (2009). A minimalist yet efficient tabu search algorithm for balancing assembly lines with disabled workers. *Pesquisa Operacional na Gestão do Conhecimento*, 660-671.

Moreria, M. C., Cordeau, J.-F., & Laporte, G. (2015a). Robust assembly line baancing with heterogeneous workers. *Computers Industrial Engineering*, 254-263.

Moreno, A., Alem, D., & Ferreira, D. (2015). Heuristic approaches for the multiperiod location-transportation problem with reuse of vehicles in emergency logistics. *Computers & Operations Research, 69,* 79-96.

Mutlu, O., Polat, O., & Supciller, A. (2013). An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II. *Computers & Operations Research, 40*(1), 418-426.

Öksüz, M. K., Büyüközkan, K., & Şatoğlu, Ş. I. (2017). U-shaped Assembly Line Worker Assignment and Balancing Problem: a Math- ematical Model and Two Meta-heuristics. *Computers & Industrial Engineering, 112,* 246-263.

Pereira, J. (2018). The Robust (minmax regret) assembly line worker assignment and balancing problem. *Computers and Operations Research,, 93,* 27-40.

Pochet, Y., & Wolsey, L. A. (2006). Production plannning by mixed integer programming. *Springer Science & Business Media*.

Polat, O., Kalaycı, C. B., Mutlu, O., & Gupta, S. M. (2015). A two-phase variable neighbourhood search algorithm for assembly line worker assignment and balancing problem type-II: an industrial case study. *International Journal of Production Research*, 722-741.

Pourrezaie-Khaligh, P., Bozorgi-Amiri, A., & Yousefi-Babadi, A. (2022). A relax-and-fix and fix-and-optimize algorithm for a Maritime Inventory Routing Proble. *Applied Mathematical Modelling, 102*, 243-267.

Rahmaniani, R., & Ghaderi, A. (2013). A combined facility location and network design problem with multi-type of capacitated links. *Applied Mathematical Modelling, 37*, 6400-6414.

Ritt, M., Costa, A. M., & Miralles, C. (2015). The assembly line worker assignment and balancing problem with stochastic worker availability. *International Journal of Production Research*, 907-922.

Ritt, M., Costa, A. M., & Miralles, C. (2016). The assembly line worker assignment and balancing problem with stochastic worker availability. *International Journal of Production Research, 54*(3), 907-922.

Sahling, F., Buschkühl, L., Tempelmeier, H., & Helber, S. (2009). Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research, 36*, 2546-2553.

Seeanner, F., Almada-Lobo, B., & Meyr, H. (2013). Combining the principles of variable neighborhood decomposition search and the fix&optimize heuristic to solve multi-level lot-sizing and scheduling problems. *Computers & Operations Research*, *40*, 303-317.

Sel, Ç., & Bilgen, B. (2014). Hybrid simulation and MIP based heuristic algorithm for the production and distribution planning in the soft drink industry. *Journal of Manufacturing Systems*, *33(3)*, 385-399.

Tanksale, A., & JHA, J. K. (2020). A hybrid fix-and-optimize heuristic for integrated inventory-transportation problem in a multi-region multi-facility supply chain. *RAIRO Operations Research*, 749-782.

Toledo, C. F., de Oliveira, R. R., & França, P. M. (2013). A hybrid multi-population genetic algorithm applied to solve the multi-level capacitated lot sizing problem with backlogging. *Computers & Operations Research*, 910-919.

Villa, M., & Pereira, J. (2014). A branch-and-bound algorithm for assembly line worker assignment and balancing problems. *Computers & Operations Research*, *44*, 105-114.

Wei, W., Guimaraes, L., Amorim, P., & Almada-Lobo, B. (2016). Tactical production and distribution planning with dependency on the production process. *OMEGA*, *67*, 99-114.

Xiao, J., Zhang, C., & Gupta, J. N. (2013). MIP-based fix-and-optimise algorithms for the parallel machine capacitated lot-sizing and scheduling problem. *Internationarl Journal of Production Research, 51*(16), 5011-5028.

Yılmaz, H. (2021). Modeling and solving assembly line worker assignment and balancing problem with sequence-dependent setup times. *Soft Computing, 25,* 12899-12914.

Yılmaz, H., & Demir, Y. (2019). A New Mathematical Model for Assembly Line Worker Assignment and Balancing. *Journal of the Institute of Science and Technology, 9*(4), 2002-2008.

Zacharia, P. T., & Nearchou, A. C. (2021). Balancing assembly lines operating with heterogeneous workers under assembly lines uncertainty in task processing times. *Engineering Computations, 38*(10), 3853-3878.

APPENDICES

AP 1: Detailed results as obtained by Akpunar for 80 instances from the Roszieg and Heskia families

	Roszieg		Heskia			Roszieg		Heskia	
(G, I)	**BKS**	**Akpunar (2022)**	**BKS**	**Akpunar (2022)**	**(G, I)**	**BKS**	**Akpunar (2022)**	**BKS**	**Akpunar (2022)**
G1 I1	20	20	94	94	**G5 I1**	10	10	35	35
G1 I2	22	22	95	95	**G5 I2**	10	10	40	40
G1 I3	18	18	102	102	**G5 I3**	10	10	35	35
G1 I4	18	18	103	103	**G5 I4**	9	9	30	30
G1 I5	17	17	92	92	**G5 I5**	12	12	40	40
G1 I6	24	24	98	98	**G5 I6**	9	9	29	29
G1 I7	21	21	116	116	**G5 I7**	10	10	25	25
G1 I8	20	20	86	86	**G5 I8**	8	8	43	43
G1 I9	22	22	95	95	**G5 I9**	10	10	38	38
G1 I10	19	19	142	142	**G5 I10**	9	9	34	34
G2 I1	30	30	169	169	**G6 I1**	11	11	51	51
G2 I2	27	27	107	107	**G6 I2**	10	10	50	50
G2 I3	76	76	108	108	**G6 I3**	10	10	52	52
G2 I4	25	25	96	96	**G6 I4**	10	10	33	33
G2 I5	26	26	130	130	**G6 I5**	11	11	38	38
G2 I6	22	22	117	117	**G6 I6**	13	13	34	34
G2 I7	22	22	146	146	**G6 I7**	13	13	42	42
G2 I8	20	20	132	132	**G6 I8**	11	11	39	39
G2 I9	27	27	101	101	**G6 I9**	12	12	59	59
G2 I10	40	40	120	120	**G6 I10**	9	9	28	28
G3 I1	28	28	200	200	**G7 I1**	16	16	66	66
G3 I2	30	30	147	147	**G7 I2**	13	13	56	56
G3 I3	26	26	186	186	**G7 I3**	19	19	69	69
G3 I4	22	22	181	181	**G7 I4**	16	16	126	126
G3 I5	28	28	143	143	**G7 I5**	14	14	107	107
G3 I6	27	27	194	194	**G7 I6**	17	17	39	39
G3 I7	21	21	149	149	**G7 I7**	17	17	87	87
G3 I8	28	28	191	191	**G7 I8**	16	16	62	62
G3 I9	27	27	170	170	**G7 I9**	15	15	61	61
G3 I10	33	33	165	165	**G7 I10**	17	17	79	79
G4 I1	31	31	204	204	**G8 I1**	15	15	91	91
G4 I2	29	29	147	147	**G8 I2**	16	16	65	65
G4 I3	32	32	211	211	**G8 I3**	16	16	73	73
G4 I4	27	27	127	127	**G8 I4**	16	16	57	57
G4 I5	27	27	181	181	**G8 I5**	16	16	65	65

G4 I6	29	29	179	179	**G8 I6**	17	17	70	70
G4 I7	27	27	191	191	**G8 I7**	13	13	71	71
G4 I8	28	28	152	152	**G8 I8**	14	14	58	58
G4 I9	21	21	167	167	**G8 I9**	14	14	46	46
G4 I10	29	29	153	153	**G8 I10**	14	14	76	76

AP 2: Detailed results as obtained by Akpunar for 40 instances from Tonge and Wee-Mag families

	Tonge				**Wee-Mag**			
(Grp, Inst)	**Periera (2014)**	**Blum (2011)**	**Akpunar (2022)**	**According to the Best (%)**	**Periera (2014)**	**Blum (2011)**	**Akpunar (2022)**	**According to the Best (%)**
G1 I1	87	93	87	0.00 %	25	28	27	8.00 %
G1 I2	87	94	88	1.15 %	26	28	27	3.85 %
G1 I3	100	106	105	5.00 %	25	27	26	4.00 %
G1 I4	105	112	111	5.71 %	29	33	31	6.90 %
G1 I5	89	92	96	7.87 %	29	32	31	6.90 %
G1 I6	88	90	88	0.00 %	25	26	25	0.00 %
G1 I7	94	95	95	1.06 %	24	27	26	8.33 %
G1 I8	104	108	111	6.73 %	27	31	29	7.41 %
G1 I9	69	73	71	2.90 %	25	27	26	4.00 %
G1 I10	83	86	83	0.00 %	26	28	27	3.85 %
G2 I1	110	112	113	2.73 %	29	32	31	6.90 %
G2 I2	108	116	118	9.26 %	30	33	33	10.00 %
G2 I3	103	105	118	14.56 %	31	32	32	3.23 %
G2 I4	93	93	103	10.75 %	30	33	33	10.00 %
G2 I5	89	95	102	14.61 %	32	34	35	9.38 %
G2 I6	100	101	100	0.00 %	29	32	31	6.90 %
G2 I7	120	123	133	10.83 %	31	34	32	3.23 %
G2 I8	112	113	128	14.29 %	34	37	35	2.94 %
G2 I9	119	128	132	10.92 %	31	33	32	3.23 %
G2 I10	113	116	117	3.54 %	35	36	36	2.86 %
G3 I1	158	171	174	10.13 %	46	50	51	10.87 %
G3 I2	160	161	169	5.63 %	44	48	48	9.09 %
G3 I3	144	148	148	2.78 %	46	48	47	2.17 %
G3 I4	147	150	164	11.56 %	47	52	49	4.26 %
G3 I5	166	174	184	10.84 %	46	49	49	6.52 %
G3 I6	161	173	170	5.59 %	45	47	46	2.22 %
G3 I7	146	149	146	0.00 %	48	56	53	10.42 %
G3 I8	176	182	182	3.41 %	45	51	46	2.22 %
G3 I9	182	189	188	3.30 %	43	45	44	2.33 %
G3 I10	153	153	153	0.00 %	48	55	50	4.17 %
G4 I1	167	175	170	1.80 %	47	51	50	6.38 %
G4 I2	177	180	187	5.65 %	44	49	46	4.55 %
G4 I3	157	157	165	5.10 %	47	52	53	12.77 %
G4 I4	172	181	184	6.98 %	51	58	54	5.88 %
G4 I5	146	152	147	0.68 %	42	47	43	2.38 %

G4 I6	163	169	174	6.75 %	46	49	47	2.17 %
G4 I7	166	167	190	14.46 %	38	42	41	7.89 %
G4 I8	152	153	162	6.58 %	44	47	47	6.82 %
G4 I9	180	183	181	0.56 %	45	50	47	4.44 %
G4 I10	159	183	182	14.47 %	39	41	40	2.56 %