NOVEMBER 2017 M.Sc. in Industrial Engineering ÖMER NEDİM KENGER

UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES

ASSEMBLY LINE BALANCING PROBLEM WITH RESOURCE AND SEQUENCE DEPENDENT SETUP TIMES

M. Sc. THESIS
IN
INDUSTRIAL ENGINEERING

BY
ÖMER NEDİM KENGER

NOVEMBER 2017

Assembly Line Balancing Problem with Resource and Sequence Dependent Setup Times

M. Sc. Thesis
in
Industrial Engineering

University of Gaziantep

Supervisor
Assist. Prof. Dr. Eren ÖZCEYLAN

By
Ömer Nedim KENGER
November 2017

REPUBLIC OF TURKEY
UNIVERSITY OF GAZIANTEP
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
INDUSTRIAL ENGINEERING

Name of the thesis: Assembly Line Balancing Problem with Resource and Sequence Dependent Setup Times

Name of the student: Ömer Nedim KENGER

Exam date: 13/11/2017

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. A. Necmeddin YAZICI

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Serap ULUSAM SEÇKİNER

Head of Department

This is to certify that we have read this thesis and that in our consensus opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Eren ÖZCEYLAN

Supervisor

Examining Committee Members	Signature
Assist. Prof. Dr. Eren ÖZCEYLAN	....
Assist. Prof. Dr. Cihan ÇETİNKAYA	....
Assist. Prof. Dr. Süleyman METE	....

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original of this work.

Ömer Nedim KENGER

ABSTRACT

ASSEMBLY LINE BALANCING PROBLEM WITH RESOURCE AND SEQUENCE DEPENDENT SETUP TIMES

KENGER, Ömer Nedim
M. Sc. in Industrial Engineering
Supervisor: Assist. Prof. Dr. Eren ÖZCEYLAN

November 2017
39 pages

Traditional assembly line balancing research has focused on the simple assembly line balancing problem (SALBP), but in recent years studies have paid attention to real life problems. Real-world systems need additional characteristics like equipment selection, cost function, paralleling, mixed-model production and U-shaped line layout etc. One of these additional characteristics is setup times and researchers have focused on this term in order to approximate their problem to real-world problems. Processes such as walking distance, turning and lifting processes, cooling or curing processes, withdrawal times and changing tools which are used between two consecutive tasks exist in real life and cause setup times between tasks. Also, while a setup is performing between two consecutive tasks, a special resource (a special machine or qualified staff) may be needed. Therefore, in this study assembly line balancing problem (ALBP) with resource dependent setup times is analyzed. A mathematical model is presented to solve problem and run for several small sized test problems. For large sized test problems a heuristic approach based on COMSOAL is proposed.

Keywords: Assembly line balancing, Setup time, Resource dependent, Mathematical model, COMSOAL

ÖZET

KAYNAK VE SIRA BAĞIMLI HAZIRLIK ZAMANLI MONTAJ HATTI DENGELEME PROBLEMİ

KENGER, Ömer Nedim
Yüksek Lisans Tezi, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. Eren ÖZCEYLAN
Kasım 2017
39 sayfa

Geleneksel montaj hattı dengeleme ile ilgili yapılan çalışmalar genellikle düz montaj hattı dengeleme problemi üzerinde yoğunlaşmıştır. Fakat son zamanlarda yapılan çalışmalar gerçek sistemlere yakın problemlere önem vermeye başlamıştır. Gerçek sistemler ise ekipman seçimi, maliyet, paralel, karışık modelli üretim ve U-tipi hat yerleşimi gibi ek karakteristiklere ihtiyaç duymaktadır. Bu karakteristiklerden biri de hazırlık zamanlarıdır ve araştırmacılar problemi gerçek uygulamalara yaklaştırmak için bu konuya odaklanmışlardır. Yürüme mesafesi, dönme ve kaldırma işlemi, geri çekilme süresi ve iki ardışık görev arasındaki ekipman değişimi gibi işlemler gerçek uygulamalarda mevcuttur ve bu işlemler görevler arasında hazırlık zamanının oluşmasına neden olur. Aynı zamanda ardışık iki görev arasındaki hazırlık işlemi için özel bir kaynak (özel makine veya eğitilmiş personel) gerekebilir. Bu sebeple bu çalışmada kaynak bağımlı hazırlık zamanlı montaj hattı dengeleme problemi incelenmiştir. Problemin çözümüne yönelik bir matematiksel model önerilmiş ve küçük boyutlu test problemleri için çalıştırılmıştır. Büyük boyutlu test problemleri için ise COMSOAL tabanlı bir sezgisel yaklaşım önerilmiştir.

Anahtar Kelimeler: Montaj hattı dengeleme, Hazırlık zamanı, Kaynak bağımlı, Matematiksel model, COMSOAL

To My Parents and Wife

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Assist. Prof. Dr. Eren ÖZCEYLAN for the continuous support of my M. Sc. study and related research, for his patience, motivation, and immense knowledge.

I am also thankful to Prof. Dr. Türkay DERELİ for his encouragement, valuable guidance, support, and suggestions throughout this study.

Special thanks to Assoc. Prof. Dr. Kürşad AĞPAK for their helpful guidance and valuable suggestions.

My sincere thanks also goes to Dr. Süleyman METE, Dr. Zeynel Abidin ÇİL and Yavuz KAHRAMAN for their aspiring guidance, invaluably constructive criticism and friendly advice during the thesis.

I would like to thank all people in the Department of Industrial Engineering for their support and help during my study.

Finally, very special thanks to my family and my wife for their inseparable support and prayer.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF ABBREVIATIONS

AL	Assembly line
ALB	Assembly line balancing
ALBP	Assembly line balancing problem
ALBPRSS	Assembly line balancing with resource and sequence dependent setup time
SA	Simulated annealing
COMSOAL	Computer method of sequencing operations for assembly lines
PSO	Particle swarm optimization
BA	Bees algorithm
RCALBP	Resource constraint assembly line balancing problem
TSALBP	Two sided assembly line balancing problem
GRCTALB	Generalized resource constraint assembly line balancing problem
GA	Genetic algorithm
CPLEX	General Algebraic Modeling System

CHAPTER I

INTRODUCTION

An assembly line (AL), is a production line that constantly move between the consecutive work stations where the assembly process is made. Large proportion of process is performed by hand in this kind of non-automatic production lines. Therefore work can be divided into smaller pieces. AL is a special form of settlement based on the product type and is usually applied in mass production system. The aim of this operation is to eliminate the time-consuming production line or decrease to the lowest level. Assembly line balancing (ALB) is a process that assigns works to workstations without exceeding cycle time. Task, task time, workstation, cycle time and precedence relation diagram are common used basic terms related to ALB. Task is the smallest indivisible part of total work in assembly process. Task time is a period required for complete a task. Workstation is a place where a part of the total amount of work is made on the production line. Cycle time is a period needed in order to perform task or tasks in a workstation. Precedence relation diagram is a tool that shows the operation sequence of tasks. An assembly line example is shown in Figure 1.1.

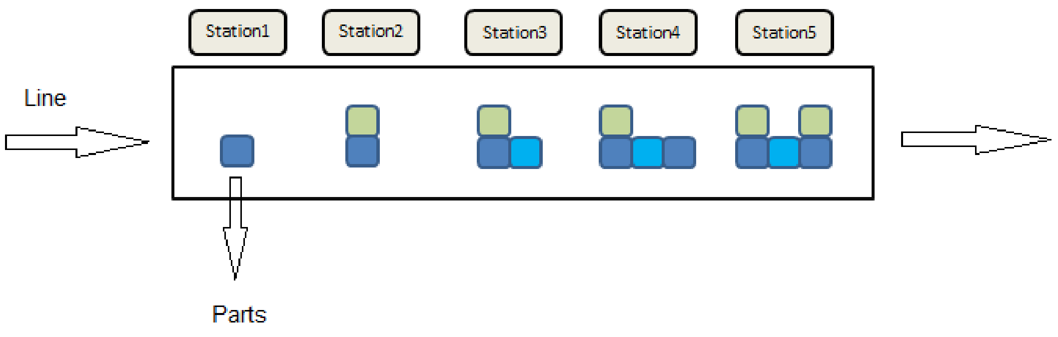

Figure 1.1. An assembly line example

AL story started with Henry Ford who is the founder of the Ford Motor Company. He changed the manufacturing system by applying mass production in order to produce automobile. Ford presented new mass-production methods, containing large manufacturing plants, the use of standardized, exchangeable parts and, in 1913, the world's first moving AL for cars in order to cope with excessive demand for the vehicles. Since then, a lot of studies have been performed on ALB.

The ALBP can be classified in different ways. According to type of line ALBP can be separated four sections as one-sided, two-sided, U-line and parallel lines. Workstations are arranged one side of line in traditional one sided AL. It is easy to place and workers perform tasks in one side of line. In two sided AL workstations are laid out at both side of the line. Large sized products are produced in this type of line. The line is U-shaped in U-line type and firstly it is used in Toyota production system. There are several lines parallel to each other in parallel AL and it is preferred when demand is high.

In terms of number of model ALBP is classified into three parts: single model, multi model and mixed model. In single model ALBP one type of product is produced in large quantities (Figure 1.2). Production is made in batches in multi model ALBP (Figure 1.3). More than one similar models are produced simultaneously in mixed model ALBP (Figure 1.4).

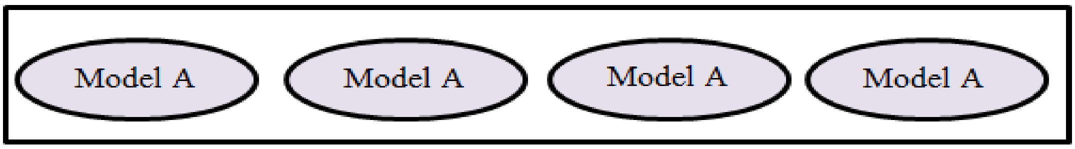

Figure 1.2. Single model assembly line

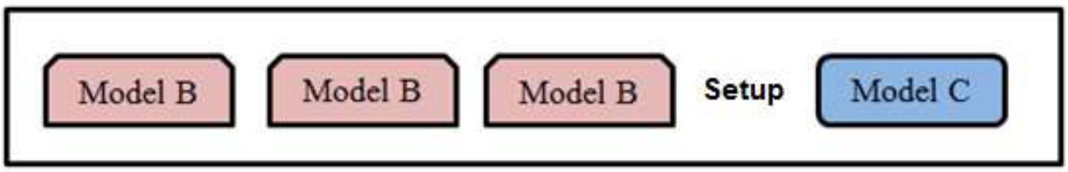

Figure 1.3. Multi model assembly line

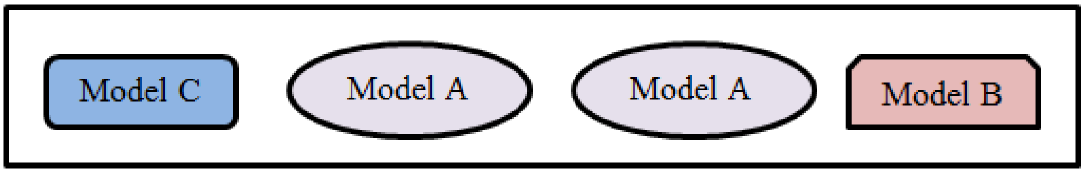

Figure 1.4. Mixed model assembly line

According to processing times ALBP is divided into three section: deterministic, stochastic and fuzzy processing time. In deterministic ALBP it is assumed that processing times are constant. In stochastic ALBP processing times are not constant. Processing times vary according to a probability distribution. Variation in processing times arise from human or operation. Processing times are uncertain in fuzzy ALBP.

Considering objective function ALBP is classified as type-1, type-2, type-E and type-F. In type-1 ALBP cycle time is constant and it is attempted to minimize number of workstations. Number of workstations are constant in type-2 ALBP and it is aimed to minimize cycle time. In type-E ALBP both cycle time and number of workstations are aimed to minimize simultaneously. And in type-F ALBP it is attempted to find feasible line balance for a given cycle time.

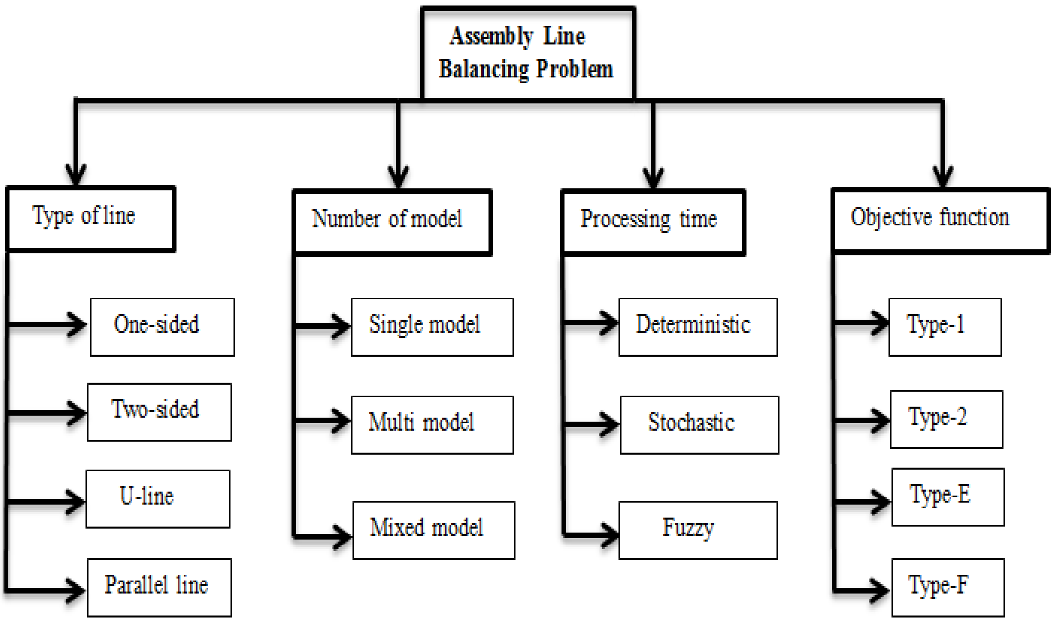

Figure 1.5. Classification of assembly line balancing problem

Setup times are another consideration in ALBP. However, most of studies in literature don’t take into account setup times. Because it is assumed that setup times have very low value compared with the processing time (Andres et al., 2008). But researches have shown that setup times have an important point at production system (Barnes 1959, Andres et al., 2008, Scholl et al., 2013). Also, the time for the tool settings can vary and depends on the sequence of jobs. Sometimes processes such as lifting, turning, cooling and curing cause a sequence-dependent setup times. Besides this, improving technology enforces facilities to use special equipment or workers (Ağpak and Gökçen, 2005). This problem arises resource constraint problem in literature. In this thesis resource and setup time are considered together. To our best knowledge there isn’t any study about this subject. Taking into consideration resource and setup times together approximate problem to real life applications. Because sometimes a special tool or worker may be needed to operate some tasks. It is meaningful to consider resource concept for task and setup times. Therefore, in this thesis ALBP with resource and sequence dependent setup times (ALBPRSS) is analyzed.

The rest of the paper is organized as follows:

In Chapter II literature review is analyzed. Studies related with ALBP with setup time and resource constraint ALBP are considered.

ALBPRSS is defined and discussed in Chapter III. ALBP and ALBPRSS are scrutinized and their assumptions are given.

Mathematical model is presented in Chapter IV. An illustrative example is solved to explain solution stages.

Heuristic solution is proposed for ALBPRSS in Chapter V and its effects are discussed.

In Chapter VI results of mathematical model and heuristic approach are presented and analyzed.

In last chapter Chapter VII conclusions of study are examined and future research directions are presented.

CHAPTER II

LITERATURE REVIEW

The first mathematical model for ALBP was proposed by Salveson (1955), and since then several studies were published in order to balance of AL. In this section resource constraint ALBP and ALBP with setup times are examined in detail. Scheduling problems with resource dependent setup time were analyzed, too (Cheng et al., 2001, Janiak et al., 2005, Harikrishnan and Ishii, 2005). But they are not took part in this thesis.

2.1 ALBP with Sequence-dependent Setup Times

Andres et al. (2008) define the sequence-dependent setup time concept to ALBP literature. The first study considers sequence-dependent setup times is published by Andres et al. in ALBP literature. They mentioned about the existence of setup times in real life and necessity of considering it in literature. For problem a mathematical model was presented. Besides this, eight different heuristic rules were proposed and compared with GRASP algorithm's results.

Özcan and Toklu (2010) consider two-sided ALB with sequence-dependent setup times. It is the first study considers sequence-dependent setup times for two-sided ALBP. The problem was considered for two objective: I. minimizing the number of the mated-stations, II. minimizing the number of the workstations for a given cycle time. A mathematical model was proposed to solve problem. Also, COMSOAL was presented for large size problems.

Seyed-Alagheband et al. (2010) deal with ALBP with sequence-dependent setup time in terms of minimizing cycle time. Simulated Annealing (SA) was proposed for solving problem. In order to increase effectiveness of SA, Taguchi method was applied.

Hamta et al. (2011) consider ALBP with sequence-dependent setup times with three objective. A mathematical model is presented for problem. They assumed that only known information for operation times is the upper and lower bound. In addition to this, operation times are dependent on worker(s) (or machine(s)) learning. For large sized test problems hybrid particle swarm optimization (PSO) is proposed by Hamta et al. (2013) for the same subject.

Yolmeh and Kianfar (2012) consider line balancing problem with sequence-dependent setup times and for solving problem a hybrid genetic algorithm is proposed by using dynamic programming. Also, sequencing of tasks was considered in addition to assignment of them to stations.

Akpınar et al. (2013) consider mixed-model ALBP with zoning constraint, parallel workstations and sequence-dependent setup times. For solving problem a hybrid algorithm which combines genetic algorithm and ant colony optimization is proposed. In addition to this, setup times are considered as low, medium, and high variability.

Scholl et al. (2013) deal with ALB and scheduling problem with sequence-dependent setup times. For solving problem a mathematical model and heuristic methods are proposed. Scholl et al. (2013)’s study is extended by adding forward and backward setup concept to Andres et al. (2008)'s study. It is emphasized that considering setup times is necessary and considering setup times approximates problem to real world problems.

Akpinar and Baykasoğlu (2014a, 2014b) publish two papers connected with each other considering mixed-model ALBP with setups in 2014. In first paper Akpinar and Baykasoğlu (2014a) proposed a new mixed-integer linear mathematical programming model for problem. The objective of mathematical model is minimization the number of the stations. In addition to setups, zoning constraints and parallel workstations are considered in order to approximate problem to real world problems.

Akpinar and Baykasoğlu (2014b) is the second part of Akpinar and Baykasoğlu (2014a)’s study. They proposed multiple colony bees algorithm (BA) for solving Akpinar and Baykasoğlu (2014a) Part-I. The multiple colony BA was tested and compared with single colony BA. Results have shown that multiple colony BA is better than single colony BA with regards to solution quality.

Diri et al. (2015) consider stochastic sequence-dependent setup times for ALBP for the first time. Considering stochastic setup times approximates problem to real world problems. In real life, operation times are not constant because of environmental factors, machine breakdowns, human factors and lack of equipment etc. In this study both operation and setup times are considered stochastic and a mathematical model is developed for problem.

Şahin and Kellegöz (2017) deal with the mixed-model U-type AL with sequence-dependent setup times and increasing production rates in that line. A mathematical model and two heuristic algorithms based on SA and GA are proposed for the problem. Results have shown that GA outperforms SA and increasing the setup times makes the performance of the algorithms worse.

Most of the previous studies claim that setup times have low proportion compared with task times and therefore setup times are not important. However, recent studies take attention that these times make the problem more applicable and realistic. Also, while a setup is being performed, a special tool or worker may be needed. Small numbers of studies consider setup times in ALBP literature but resource dependent setup times have not been considered, yet. Therefore, in this thesis resource dependent concept for setup times is considered and analyzed.

2.2 Resource Constraint Assembly Line Balancing Problem

In ALBP literature there are just a few studies related to the resource constraint. In other studies, same resources are used for all operations. Studies related with resource constraint are given below.

Ağpak and Gökçen (2005) describe ALBP with resource constraint. Objective of study is balancing AL with minimum number of resource and workstations. Problem is defined in two ways: I. Type1 resource constraint ALBP (RCALBP), II. Type2 RCALBP. In type1 RCALBP each operation is carried out by only one resource and in type2 RCALBP certain operation can be performed by two resource alternatively. For solving problem, 0-1 integer programming model is proposed and conducted on a numerical example.

Corominas et al. (2011) deal with general RCALBP in their study. A mathematical model presented to solve problem. Corominas et al. (2011) consider problem in terms of more general point of view. In this case, a task can be performed by one or more than one resource simultaneously. In their study, two different types of model are defined in terms of resource sequence. Also, an upper bound is defined for number of resources. Presented mathematical model is performed on benchmark test problems and results have shown that problems are solved efficiently.

Purnomo et al. (2013) consider two sided ALBP (TSALBP) with assignment constraint. A mathematical model is proposed for problem. Objective of model is to minimize cycle time. First-fit-rule and Genetic Algorithm (GA) are proposed for solving problem. Also they considered resource constraint for TSALBP.

Mete and Ağpak (2013) consider multi objective generalized resource constraint TSALBP (GRCTALB). A mathematical model is presented for problem. Three objective function are considered for problem as minimization number of station, number of position and resource cost. Proposed model is run on test problems and results have shown that objective sequence is effective on solution.

Triki et al. (2017) deal with AL resource assignment and balancing problem of type2. Differently from other studies, objective of this study is minimizing the cycle time for fixed number stations. A mathematical model is presented but it is insufficient for problem solving. Therefore, hybrid multi objective genetic algorithm, a new kind of multi-objective genetic algorithm, is proposed for large scale problems.

Quyen et al. (2017a), consider RCALBP and applied to sewing line of a footwear company. A mathematical model is proposed based on Ağpak and Gökçen (2005). Also, hybrid GA is proposed for problem and results have shown that hybrid GA provides improved results for the sewing line of company.

Quyen et al. (2017b), deal with RCALBP. Differently from their previous study, dynamic programming is proposed for the sewing line of a footwear company. The proposed algorithm is run for 18 test problems and it is seen that dynamic programming achieve a good performance for footwear company.

Existing studies in the literature, consider resource constraints, neglect the setup times. However, setup times have an important role in real manufacturing system (Andres et al., 2008, Scholl et al., 2013) and should be taken into consideration while balancing AL. In addition, there isn’t any study considers both setup times and resource concept together. Therefore, in this thesis ALBPRSS is analyzed.

CHAPTER III

PROBLEM DEFINITION

In this chapter ALBPRSS is defined and analyzed. The objective function is to minimize number of resource under given cycle time. Definition of ALBP and ALBPRSS and their assumptions are given below.

3.1 Assembly Line Balancing Problem

During the industrialization process, by separating the total work items, It has been suggested that faster and mass production and cheaper production can be done by the workers. As a result, production takes place through the passing of parts over a specific line on which different workstations are located. The system consisting of the stations that they create by combining the materials on the basis of the priorities of the operations on the part and the constraints such as cycle time is called the "assembly line". Workers located at the stations on the line carry out one or more operations related to their work items. As a result of this process, even the incoming parts and semi-finished products come out from the end of the line as a product.

During production, assigning tasks to workstations in such a way to provide some conditions (cycle time, precedence relationship and assignment) is called ALB. The AL are a special type of arrangement according to the product and are usually applied in systems with mass production type. The purpose of this arrangement is to eliminate or at least reduce the time loss in the work being done on the production line.

Advantages of AL are given below:

- Capacity utilization is generally high and unit production times are small.
- Material flow is systematic and easy to control.
- There is no need for qualified staff depending on the work sharing and staff training period is short.

- Workload is evenly distributed to stations.
- The total production area is small, since fewer transport and storage areas are needed.
- In-process inventory requirement is very low.

Objectives of ALBP are given below:

- Providing regular material flow.
- Using the minimum amount of time for processing.
- Using the minimum amount of material for processing.
- Minimizing the number of workstations.
- Minimizing idle times.
- Minimizing line balancing cost.

There are three main constraints in simple ALBP. These are: I. Assignment constraint, II. Precedence constraint, III. Cycle time constraint. Assignment constraint provides that each task is assigned to only one station. Precedence constraint provides precedence relations between tasks. Cycle time constraint provides total station time cannot bigger than cycle time. One the other hand, these constraints are not enough to meet customer demands in current market conditions. Therefore, much type of lines is designed and new constraints are added in addition to these three basic constraints.

3.2 Problem definition

The resource constraint ALBP emerged in a factory that has some specific tools and a few workers who can use these tools (Ağpak and Gökçen, 2005). So as to operate some tasks a special tool or worker may be needed and each worker may operate tasks in different periods. In ALBPRSS, each task and setup can be performed by special tools or workers and objective of problem is to minimize resource usage and workstation simultaneously. In this study it is assumed that each task and setup may be performed by three different resources. Also, operation and setup time of a task can be different for each resource.

Setup times are considered same for both cycles in Andres et al. (2008)'s study. However, Scholl et al. (2013) define the setup times in two different ways (forward and backward setup). If task i is performed before task p in the same station and there is a setup between them in the same cycle, this setup is defined as forward setup. If task i is the first task and task p is the last task in the same station and there is a setup between task p and task I in the next cycle, this setup is defined as backward setup. In this thesis, both forward-backward setup times and resource concept are considered together. Setup times vary according to cycle type (same or next cycle).

Table 3.1 Forward setup times depend on resources

Tasks	Resource		
	A	B	C
1,2	3	1	2
1,3	2	1	1
2,4	4	2	3
2,5	1	3	1
3,6	1	2	3
6,7	2	1	4

Table 3.2 Backward setup times depend on resources

Tasks	Resource		
	A	B	C
2,1	1	2	4
3,1	4	1	3
4,2	2	3	1
5,4	3	4	2
6,5	3	1	4
7,4	4	1	2
7,6	2	2	3

As seen in the tables 3.1 and 3.2, there are three resources that can perform each setup and objective is to determine which setup is performed by which resource in order to minimize total number of resource. In this study this concept is considered for both task and setup times. Resource-dependent setup time is firstly considered for ALBP in this thesis.

Problem assumptions are presented below.

- Precedence relations between tasks are known.
- A task cannot be assigned to more than one station.
- Setups occur between tasks and task and setup times are resource-dependent.
- Task and setup times vary according to resource.
- Tasks cannot be taken to pieces and they must be done.
- One side of the line is used.

CHAPTER IV

MATHEMATICAL MODELS

In this study ALBP with resource-dependent setup times is considered that each task and setup can be performed by special tool or worker. In this chapter mathematical model which developed by Diri, Z. (2015) is modified according to resource-dependent setup times. Modified constraints by adding resource constraints are explained and analyzed.

4.1 Mathematical model

The mathematical model which is developed for ALBPRSS is given below and constraints are explained.

Table 4.1 Notations

i,k	Task
j	Station
s	Position within the sequence of a station
r	Resource
N	Task set (i=1, ... , N)
m_{min}	Lower bound for station number
m_{max}	Upper bound for station number
t_{ir}	Operation time of task i with resource r
TC	Cycle time
T_j	Set of tasks that can be assigned to station j

A_{rj}	Resource used in workstation j
P	Set of pairs of tasks (i,k) in which i is immediate predecessor of k
PT_i	Set of tasks all predecessors of task i, consisting non immediate predecessors.
ST_i	Set of tasks all successors of task i
Nm_j	Maximum task number that can be assigned to station j
NT_m	Maximum task number that can be assigned to any station $NT_m = max_j(Nm_j)$
tsu_{ikr}	Forward setup time with resource r when task k is performed just after task i inside the same station
$tsub_{ik}$	Backward setup time with resource r when task k is performed just after task i inside the same station
W_{max}	Big number

Variables

$x_{ijs} \epsilon \{0,1\}$ 1 if task i is assigned to station j in position s of its schedule

$y_j \epsilon \{0,1\}$ 1 if any task is assigned to station j

$z_{ikj} \epsilon \{0,1\}$ 1 if task i immediate predecessor of task k in the station j in the same cycle

$zb_{ikj} \epsilon \{0,1\}$ 1 if task i immediate predecessor of task k in the station j in the next cycle

$w_{ij} \epsilon \{0,1\}$ 1 if task i is the last one in the sequence of tasks assigned to station j

Objective function

$$Min\ Z = \sum_{j=1}^{m_{max}} (j.A_{rj}) \quad (4.1)$$

Subject to:

$$\sum_{j=1}^{m_{max}} \sum_{s=1}^{Nm_j} x_{ijs} = 1 \quad \forall(i), \quad (4.2)$$

$$\sum_{\forall_i \epsilon T_j} x_{ijs} \leq 1 \quad (\forall j;\ s = 1, \dots, Nm_j), \quad (4.3)$$

$$\sum_{\forall_i \epsilon T_j} x_{ij,s+1} \leq \sum_{\forall_i \epsilon T_j} x_{ijs} \quad (\forall j;\ s = 1, \dots, Nm_j - 1), \quad (4.4)$$

$$\sum_{j=1}^{m_{max}} \sum_{s=1}^{Nm_j} (NT_m.(j-1)+s).x_{ijs} \leq \sum_{j=1}^{m_{max}} \sum_{s=1}^{Nm_j} (NT_m.(j-1)+s).x_{kjs} \quad (\forall(i,k)\epsilon P), \quad (4.5)$$

$$\sum_{\forall_i \epsilon T_j} \sum_{s=1}^{Nm_j} t_{ir}.x_{ijs} + \sum_{\forall(i,k)(i \neq k)\wedge(i,k\epsilon T_j)} tsu_{ikr}.z_{ikj} + \sum_{\forall(i,k)(i \neq k)\wedge(i,k\epsilon T_j)} tsub_{ikr}.zb_{ikj} \leq TC * y_j + M(1 - A_{rj})$$

$$(j = 1, \dots, m_{max}), \quad (4.6)$$

$$x_{ijs} + x_{kj,s+1} \leq 1 + z_{ikj} \quad (\forall j;\ s = 1, \dots, Nm_j - 1; \forall(i,k)/(i \neq k)\wedge(i,k\epsilon T_j)\wedge(k \notin PT_i)), \quad (4.7)$$

$$x_{ijs} - \sum_{\forall_k \epsilon T_j (i \neq k)\wedge(k \notin PT_i)} x_{ij,s+1} \leq w_{ij} \quad (\forall j; s = 1, .., Nm_j - 1;\ \forall(i) \in T_j), \quad (4.8)$$

$$w_{ij} + x_{kj1} \leq 1 + zb_{ikj} \quad (\forall j;\ \forall(i,k)|(i \neq k)\wedge(i,k\epsilon T_j)\wedge(i \notin PT_k)), \quad (4.9)$$

$$\sum_{s \in Nm_j} \sum_{\forall_i \epsilon = T_j} X_{ijs} . \quad - W_{max} . y_j \leq 0 \qquad \forall \, (j) \in J \tag{4.10}$$

$$y_{j+1} \leq y_j \qquad (j = 1, .., m_{max} - 1), \tag{4.11}$$

$$\sum_{j=1}^{m_{max}} \sum_{\forall k | tsu_{ik} > 0 \wedge k \notin \mathrm{PT_i}}^{N} z_{ikj} \leq 1 \qquad \forall i \tag{4.12}$$

$$\sum_{j=1}^{m_{max}} \sum_{\forall k | tsub_{ik} > 0 \wedge k \notin \mathrm{ST_i}}^{N} zb_{ikj} \leq 1 \qquad \forall i \tag{4.13}$$

$$y_j - \sum_{1}^{R} (A_{rj}) \leq 0 \qquad \forall \mathrm{j} \tag{4.14}$$

$$x_{ijs}, y_j, z_{ikj}, zb_{ikj,} w_{ij} \; \epsilon \; \{0,1\} \tag{4.15}$$

Both minimization of the number of workstations and resource usage are provided by objective function (4.1). Constraint (4.2) is assignment constraint and ensures that each task must be assigned to only one position inside only one station. Constraint (4.3) ensures that no more than one task assigned to every position inside every station. Constraint (4.4) ensures that the tasks must be assigned in increasing positions in the schedule of each station. Constraint (4.5) is the precedence constraint and provides that next task cannot be assigned without performing previous task. Constraint (4.6) is cycle time constraint and ensures that the total station time cannot exceed the cycle time. Constraint (4.7) ensures that z_{ikj} is equal to 1 if tasks i and k are assigned to positions s and s+1 respectively. Constraint (4.8) ensures that w_{ij} is equal to 1 if task i is assigned to the last position in station j. Constraint (4.9) ensures that z_{ikj} is equal to 1 if task i is assigned to the last position and task k to the first position in station j. Constraint (4.10) controls if station open or not; constraint (4.11) ensures opening workstations respectively. (4.12) and (4.13) prevent z to take unneeded value. Constraint (4.14) ensures that if station is open, at same time resource must assign to this station. Constraint (4.15) defines the binary variables.

4.2 An Illustrative Example

An example is solved in GAMS for ALBPRSS to in order to test validity of mathematical model. An 11 tasks test problem from SBF-2 data sets is used for solving mathematical model. The A resource values are accepted same with SBF-2 data sets and the other resource values (B,C,D,E) are produced randomly. Cycle time is taken as 7 units. Task times, forward and backward setup times are given in Table 4.2, Table 4.3 and Table 4.4.

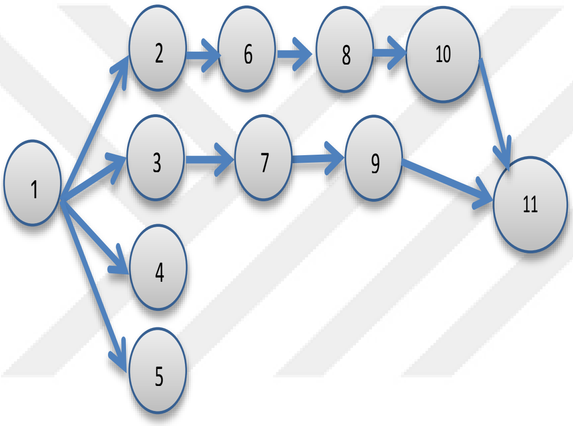

Figure 4.1. An example of precedence diagram with 11 tasks

Table 4.2 Operation time of tasks

Task	Resource				
	A	B	C	D	E
1	6	5	4	7	2
2	2	1	3	1	2
3	5	5	6	4	3
4	7	3	8	4	1
5	1	2	5	4	1
6	2	3	3	4	2
7	3	6	3	5	7
8	6	8	4	1	5
9	5	5	3	6	7
10	5	3	4	6	2
11	4	4	5	6	3

Table 4.3 Forward setup times between tasks

Task	Resource				
	A	B	C	D	E
1.2	1	0	2	1	3
1.3	1	2	1	0	2
1.4	1	0	2	1	2
2.4	2	1	1	0	2
2.5	1	1	1	1	2
2.6	1	0	2	1	3
2.7	1	3	1	0	1
2.9	1	2	0	0	1
3.4	2	1	2	3	1
3.5	1	2	1	2	3
3.6	1	1	2	2	2
3.7	1	3	0	1	2
3.8	1	0	2	2	1
4.2	2	1	0	0	0
4.3	2	1	2	2	2
4.5	1	0	1	1	3
4.6	1	3	1	1	1
4.7	1	1	1	2	1
4.8	1	2	0	0	0
4.10	2	1	0	2	1
5.2	1	2	1	1	1
5.3	1	0	2	1	2
5.4	1	1	0	2	0
5.10	1	2	2	1	1
6.3	1	1	0	0	0
6.4	1	1	2	2	1
7.2	1	0	1	1	0
7.10	1	2	1	1	3
8.3	1	1	2	1	0
8.4	1	3	1	0	1
8.10	1	2	2	0	2
9.2	1	1	1	1	1
9.10	1	0	0	0	2
9.11	2	1	0	1	0
10.4	2	1	0	0	1
10.5	1	0	1	1	0
10.7	1	1	0	2	1
10.9	1	2	0	1	1
10.11	1	1	1	2	0

Table 4.4 Backward setup times between tasks

Task	Resource				
	A	B	C	D	E
4.1	1	0	2	1	0
4.2	2	1	1	1	1
4.3	2	2	1	0	0
4.5	1	1	0	0	2
4.6	1	0	0	1	2
4.8	1	1	2	2	1
4.10	2	0	0	0	3
5.2	1	0	3	2	1
5.3	1	3	1	2	2
5.10	1	2	1	0	0
6.2	1	1	2	1	2
6.3	1	3	0	1	2
7.2	1	0	1	2	1
7.3	1	2	2	1	1
7.10	1	1	2	3	1
8.2	1	0	3	2	3
8.3	1	0	2	2	3
9.2	1	2	2	0	1
9.3	1	2	0	0	0
9.10	1	0	1	0	1

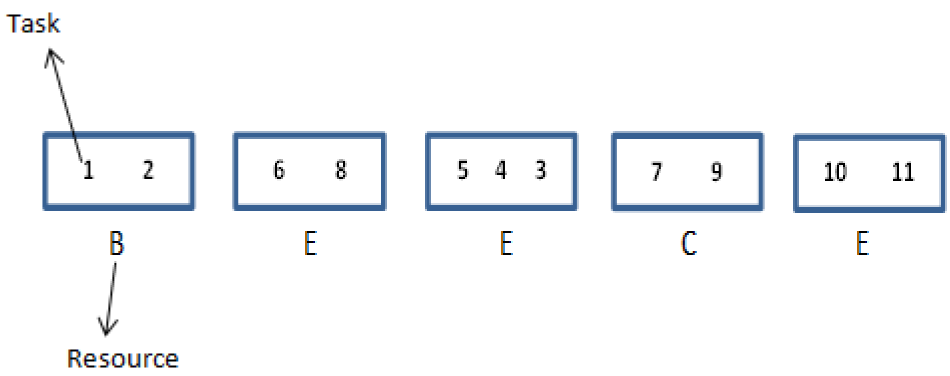

Figure 4.2. Result of test problem with 11 tasks

The proposed mathematical model is formulated with GAMS/CPLEX (General Algebraic Modeling System) and run on a computer with Core(TM) i7-2630QM 2.0 GHz dual processor and 6.00 GB memories for certain test problems. The result of test problem is given above and as it is understood from Figure 4.2, 5 workstations are open and in total 3 resources (B, C, E) are used in these workstations. In normal situation when we consider resources as workers, one worker is needed to assign each station, so 5 workers are assigned instead of 3 workers. However, when resource concept is taken into consideration, number of worker decrease, therefore cost of workers decrease.

CHAPTER V

HEURISTIC PROCEDURE

Heuristic is a method designed for solving a problem. Heuristic methods solve problems more quickly than classical methods or find an approximate solution when classic methods are unsuccessful to find exact solution. The purpose of the heuristic method is to produce a solution in a reasonable time that is good enough for solving the problem at hand. This solution may not be the best of all solutions but it may be helpful to approximate the exact solution. Heuristics may produce results by themselves, or they may be used for algorithms to improve their efficiency. In this study Computer Method of Sequencing Operations for Assembly Lines (COMSOAL), which is an efficient procedure for ALBP, is proposed for solving the ALBPRSS.

5.1 COMSOAL

COMSOAL method is developed by Arcus in 1966. COMSOAL is able to produce several balances for ALBP by considering some constraints. The procedure aims to iterate through a sequence of alternative solutions and keep the best one.

The COMSOAL program proceeds in 6 steps as follows:

STEP 1: For each task, identify those tasks which immediately follow it in precedence order.

STEP 2: Place in list A for each task in the assembly, the total number of tasks which immediately precede it in the precedence diagram.

STEP 3: From list A, create list B composed of the tasks which have zero predecessors. If no task remains predecessors. If no task remains unassigned to stations, then stop.

STEP 4: From list B, create list C composed of the tasks whose performance times are no greater than the available time times are no greater than the available time at the station. If list C is empty, open a new station with the full cycle time available and go through Step 4 again.

STEP 5: Randomly select a task from list C for assignment to the station.

STEP 6: Update the time available at the station and list B to reflect the time station and list B to reflect the time consumed and the completed predecessors at this stage. If list B is empty update list A and return to Step 3 otherwise return to step 4.

The advantages of COMSOAL algorithm are listed below:

- Simplifies complex ALBP ,
- Gives faster, easier, and more accurate result than calculating by hand ,
- Saves time and money.

The general procedure steps of the proposed approach which is based on COMSOAL algorithm for deterministic model are summarized below:

STEP 1: Precedence diagram and setup time between tasks are occurred. Then, Create a list (list A), in one column, for each task in the assembly, the total number of tasks which immediately precede it in the precedence diagram.

STEP 2: From list A, create list B composed of the tasks which have zero predecessors. If no task remain predecessors. If the task is first assigned task in the station, return Step 3. Otherwise, return Step 4.

STEP 3: From list B, create list C composed of the tasks whose performance times are no greater than the available time times are no greater than the available time at the station. Continue Step 5.

STEP 4: Add to C list the task in the B list if the sum of task time, forward setup times and backward setup between tasks is less than or equal idle time of station.

STEP 5: If list C is empty, open a new station with the full cycle time available and go through Step 3 again. Otherwise continue.

STEP 6: Randomly select a task from list C for assignment to the station. (This random selection causes the formation of different balances.)

STEP 7: Selected task is removed from the precedence diagram and update idle time in the station. If all tasks assigned to a station, balancing has been completed. If there are tasks which are not assigned, return to Step 2.

CHAPTER VI

COMPUTATIONAL RESULTS

6.1 Mathematical model’s results

In this section, test problems are taken from Scholl et al. (2013)'s study (SBF2-data sets) are solved and results are given in Table 6.1. There are four different levels for setup time (0.25, 0.5, 0.75, and 1.0) in SBF-2 data sets and “1.0” level is determined in this thesis in order to run models on test problems. Models are tested according to minimization the number of resources and workstations.

The solution time is limited within 3600 seconds. The proposed mathematical model is solved with GAMS by a computer Core(TM) i7-2630QM 2.0 GHz dual processor and 6.00 GB memories for certain test problems. The results of mathematical model are shown in table 6.1. Proposed model solves the test problems include 21 tasks and founds optimal solution except T11/C48, T21/C14, T21/C15, T21/C21. However, it can't be reached to optimal solution for large-sized problems within an acceptable time.

Table 6.1 Test results of mathematical model

Problem	Cycle time	Station numbers	Resource	CPU
T7	6	4	A,C	0,968
	7	4	A,C	0,265
	8	3	A,C	0,421
	10	3	B,C	0,25
	15	2	A,C	0,68
	18	2	A,C	0,413
T8	20	5	A	0,562
T9	6	5	B,C	0,125
	7	4	C	0,235
	8	3	B,C	0,078
	10	3	C	0,062
	18	3	A,C	0,056
T11	7	6	A,B,C	11,75
	9	4	A,B,C	0,266
	10	4	A,B,C	0,285
	13	3	A,B,C	0,282
	14	3	A,B,C	0,198
	21	3	A,C	0,194
T11	48	6**	A,B,C	100,62
	62	3	A,B	16,785
	94	3	A,B	2,293
T21	14	#	#	#
	15	#	#	#
	21	#	#	#
	26	4	A,B	15,65
	35	4	A,B	16,47
	39	4	A,B	14,812

** Integer Solution, # error No Solution

6.2 Heuristic procedure's results

In this section, the large size test problems with different cycle time are solved. The "1.0" level is determined from SBF-2 data sets in this thesis in order to test COMSOAL for ALBPRSS. The heuristic algorithm are solved by a computer with MATLAB Intel(R) Core(TM) i7-2630QM 2.0 GHz dual processor and 6.00 GB memories for certain test problems. 269 test problems are solved in total, but these problems' results are not compared with any results as this study is the first study in literature. A small number of heuristic results are given in below and the all results are presented in Appendix.

Table 6.2 Test results of heuristic algorithm

Problem	Cycle time	Station number	CPU	Resource
T9	6	5	12,660	B,C
	7	4	13,480	B,C
	8	4	13,140	B,C
	10	3	13,540	A,C
	18	3	13,530	C
T11	7	6	16,270	A,B,C
	9	5	16,880	A,B,C
	10	4	17,150	A,B,C
	13	3	17,210	A,B,C
	14	3	17,990	A,B,C
	21	3	17,890	B,C
T28	138	7	63,180	A,B,C
	205	5	68,190	A,B,C
	216	5	68,200	A,B,C
	256	4	68,440	A,B,C
	324	4	73,250	A,C
	342	3	74,140	A,B
T53	2004	8	94,010	A,B,C
	2338	7	94,990	A,B,C
	2806	6	96,670	A,B,C
	3507	5	97,880	A,B
	4676	4	97,290	A,C

6.3 Comparison of mathematical model and heuristic's results

Mathematical model and heuristic solution results cannot be compared with any results as this study is the first study in literature. However, when mathematical model and heuristic solution results (small sized test problems) are compared with each other, it is seen that the mathematical model achieved better results in terms of CPU and resource number. In addition station numbers are found to be equal for the both solution method. Mathematical model and heuristic solution results are given in Table 6.3.

Table 6.3 Comparison of mathematical model and heuristic's results

Problem	Cycle time	Station numbers	Resource	CPU	Station numbers*	Resource *	CPU*
P7	6	4	A,C	0,968	4	A,C	9,650
	7	4	A,C	0,265	4	A,C	9,850
	8	3	A,C	0,421	3	B,C	10,030
	10	3	B,C	0,25	3	A,B,C	10,380
	15	2	A,C	0,68	2	B	10,630
	18	2	A,C	0,413	2	B,C	10,790
P8	20	5	A	0,562	5	A,B,C	11,070
P9	6	5	B,C	0,125	5	B,C	12,660
	7	4	C	0,235	4	B,C	13,480
	8	3	B,C	0,078	4	B,C	13,140
	10	3	C	0,062	3	A,C	13,540
	18	3	A,C	0,056	3	C	13,530
P11	7	6	A,B,C	11,75	6	A,B,C	16,270
	9	4	A,B,C	0,266	5	A,B,C	16,880
	10	4	A,B,C	0,285	4	A,B,C	17,150
	13	3	A,B,C	0,282	3	A,B,C	17,210
	14	3	A,B,C	0,198	3	A,B,C	17,990
	21	3	A,C	0,194	3	B,C	17,890
P11	48	6**	A,B,C	100,62	6	A,B	16,480
	62	3	A,B	16,785	3	A,B	16,630
	94	3	A,B	2,293	3	A,B	16,850
P21	14	#	#	#	7	A,B,C	32,490
	15	#	#	#	7	A,B,C	32,940
	21	#	#	#	5	A,B,C	33,100
	26	4	A,B	15,65	4	A,B	33,520
	35	4	A,B	16,47	4	A,B	33,540
	39	4	A,B	14,812	4	A,B	33,600

*Heuristic solution results

CHAPTER VII

CONCLUSIONS

In this thesis, a mathematical model and a heuristic algorithm is presented for ALBPRSS. Proposed model and algorithm is conducted on test problems and results are analyzed. In ALBP literature additional characteristics like equipment selection, cost function, paralleling, mixed-model production and U-shaped line layout have been paid attention in order to approximate problem to real-life problems. One of these characteristics is setup times and this thesis has focused on setup times because of existence of them in real-world systems. On the other way, while a setup is performing between two consecutive tasks, a special resource (a special machine or qualified staff) may be needed. So, in this thesis these two concept are considered together and applied to ALBP for the first time.

Proposed mathematical models are solved in GAMS/CPLEX, firstly. Results have shown that deterministic model reached to optimal solution including 21 tasks test problem except a few test problems. For large size test problems, an effective heuristic approach for ALBP, COMSOAL is presented and solved in MATLAB. Results have shown that algorithm is effective for the problem.

To the best of our knowledge, ALBPRSS is firstly studied in this thesis. Therefore, several new ideas can be improved about this subject. Future researches can focus on applying more effective algorithm like beam search, genetic algorithm etc. for the problem.

REFERENCES

Ağpak, K. and Gökçen, H. (2005). Assembly line balancing: Two resource constraint cases. *International.J ournal. Production Economics*, **96**, 129-140.

Akpınar, S., Bayhan, G. M., and Baykasoğlu, A. (2013). Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks. *Applied Soft Computing* , **13,** 574–589.

Akpinar ,S., and Baykasoglu, A. (2014a). Modeling and solving mixed-model assembly line balancing problem with setups. Part I: A mixed integer linear programming model. *Journal of Manufacturing Systems* , **33**:1, 117-187.

Akpinar ,S., and Baykasoglu, A. (2014b). Modeling and solving mixed-model assembly line balancing problem with setups. Part II: A multiple colony hybrid bees algorithm. *Journal of Manufacturing Systems*, **33:**4, 445-461.

Andres, C., Miralles, C., Pastor, R. (2008). Balancing and scheduling tasks in assembly lines with sequence-dependent setup times. *European Journal of Operational Research* ,**187,** 1212–1223.

Arcus, A. (1966). A computer method of sequencing operations for assembly lines. International Journal of Production Research , **4**, 259-277.

Barnes, R. (1959) Motion and time study. Wiley, New York

Cheng, T.C.E., Janiak, A., and Kovalyov, M.Y. (2001). Single machine batch scheduling with resource dependent setup and processing types. *European Journal of Operation Research* ,**135**, 177-183.

Corominas, A., Ferrer, L., and Pastor, R. (2011). Assembly line balancing: general resource-constrained case. *International Journal of Production Research,* **49**:12**,** 3527-3542.

Diri, Z., Mete, S., Çil, Z.A., and Ağpak, K. (2015). Assembly line balancing problem with stochastic sequence-dependent setup times. *Pamukkale University Journal of Engineering Sciences*, **21**:4**,** 152-157.

Hamta, N., Ghomi, S.M.T. F., Hakimi-Asiabar, M. and Tabrizi, P. H. (2011). Multi-objective assembly line balancing problem with bounded processing times, learning effect, and sequence-dependent setup times. *Proceeding of the 2011 IEEE IEEM*, 978-1-4577-0739-1/11.

Hamta, N., Ghomi, S.M.T.F., Jolai, F., and Shirazi, M. A. (2013). A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect. *International Journal of Production Economics,* **141**, 99–111.

Harikrishnan, K.K., and Ishii, H. (2005). Single machine batch scheduling problem with resource dependent setup and processing time in the presence of fuzzy due date. *Fuzzy Optimization and Decision Making,* **4**, 141-147.

Janiak, A., Kovalyov M.Y., and Portmann, M.C. (2005). Single machine group scheduling with resource dependent setup and processing times. *European Journal of Operation Research*, **162**, 112-121.

Mete, S., and Ağpak, K. (2013). Multi objective generalized resource constrained two sided assembly line balancing problem and computational analysis. *Gazi Üniversitesi Mühendislik-Mimarlık. Fakültesi Dergisi*, **Cilt 28**, No 3, 567-576.

Özcan, U., and Toklu, B. (2010). Balancing two-sided assembly lines with sequence dependent setup times. *International Journal of Production Research* , **48:18**, 5363-5383.

Purnomo, H., Wee,H., and Rau, H. (2013). Two-sided assembly line balancing with assignment restrictions. *Mathematical and Computer Modelling,* **57**:189-199.

Quyen, N.T.P., Chen, J.C., and Yang, C.L. (2017a). Hybrid genetic algorithm to solve resource constraint assembly line balancing problem in footwear manufacturing. *Soft Computing,* **21**:6279-6295.

Quyen, N.T.P., Kuo, R.J., Chen, J.C., and Yang, C.L. (2017b). Dynamic programming to solve resource constraint assembly line balancing problem in footwear manufacturing. *International Conference on Industrial Engineering and Applications.*

Salveson, M. (1955). The Assembly line balancing problem. *The Journal of Industrial Engineering* , **6**:3, 18-25.

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

Seyed-Alagheband, S.A., Ghomi, S.M.T.F., and Zandieh, M. (2010). A simulated annealing algorithm for balancing the assembly line type II problem with sequence-dependent setup times between tasks. *International Journal of Production Research*, **49:3**, 805-825.

Şahin, M., and Kellegöz, T. (2017) Increasing production rate in U-type assembly lines with sequence-dependent setup times. *Engineering Optimization*, **49:8**, 1401-1419.

Triki, H., Mellouli, A., and Masmoudi, F. (2017). A multi-objective genetic algorithm for assembly line resource assignment and balancing problem of type 2, *Journal of Intelligent Manufacturing,* **28**:371-385.

Yolmeh, A., and Kianfar, F. (2012). An efficient hybrid genetic algorithm to solve assembly line balancing problem with sequence-dependent setup times. *Computers & Industrial Engineering* , **62**, 936–945.

APPENDIX

RESULTS OF HEURISTIC APPROACH

Problem	Cycle time	Number of Workstation	CPU	Resource
T83	3786	23	174,060	A,B,C
	3985	21	176,210	A,B,C
	4206	21	184,910	A,B,C
	4454	20	178,810	A,B
	4732	19	179,890	A,B,C
	5048	18	179,104	A,B,C
	5408	17	181,510	A,B
	5824	16	300,370	A,B
	5853	16	257,690	A,B
	6309	14	184,940	A,B,C
	6842	13	189,160	A,B
	6883	13	184,290	A,B
	7571	13	188,840	A,B,C
	8412	11	155,650	A,B
	8898	11	184,180	A,B
	10816	9	157,240	A
T111	5755	19	157,400	A,B,C
	5785	18	156,200	A,B,C
	6016	18	157,400	A,B,C
	6267	18	157,000	A,B,C
	6540	17	156,600	A,B,C
	6837	16	157,400	A,B,C
	7162	16	161,600	A,B,C
	7520	15	162,000	A,B,C
	7916	14	159,400	A,B,C
	8356	14	161,200	A,B,C
	8847	13	162,000	A,B,C
	9400	13	163,600	A,B,C
	10027	12	165,000	A,B,C
	10743	11	164,600	A,B,C
	11378	11	162,200	A,B,C
	11570	11	163,800	A,B,C
	17067	8	169,400	A,B,C
T148	84	50	344,600	A,B,C
	85	50	343,800	A,B,C
	87	49	341,800	A,B,C

Problem	Cycle time	Number of workstation	CPU	Resource
T148	89	49	348,800	A,B,C
	91	46	354,000	A,B,C
	93	45	348,400	A,B,C
	95	46	350,800	A,B,C
	97	44	353,000	A,B,C
	99	44	354,800	A,B,C
	101	43	352,600	A,B,C
	104	42	353,600	A,B,C
	106	41	355,300	A,B,C
	109	40	355,700	A,B,C
	112	39	356,200	A,B,C
	115	39	360,400	A,B,C
	118	37	362,000	A,B,C
	121	37	365,100	A,B,C
	125	36	363,800	A,B,C
	129	35	364,600	A,B,C
	133	34	365,700	A,B,C
	137	33	366,500	A,B,C
	142	32	368,400	A,B,C
	146	31	367,200	A,B,C
	152	30	369,500	A,B,C
	157	29	370,450	A,B,C
	163	28	372,800	A,B,C
	170	27	373,600	A,B,C
	403	13	368,400	A,B,C
	434	12	370,700	A,B,C
	470	11	369,200	A,B,C
	513	10	370,400	A,B,C
	564	9	372,600	A,C
	626	9	373,700	A
	705	8	375,800	A,C
	805	7	374,300	A
T8	20	5	11,070	A,B,C
T29	27	12	48,710	A,B,C
	30	11	48,910	A,B,C
	33	11	49,520	A,B,C
	36	10	49,990	A,B,C
	41	9	50,640	A,B,C
	47	7	51,560	A,B,C
	54	7	55,980	A,B,C
T35	41	11	59,830	A,B,C

Problem	Cycle time	Number of workstation	CPU	Resource
T35	44	11	60,120	A,B,C
	49	9	60,680	A,B,C
	54	8	60,750	A,B,C
	61	8	61,350	A,B,C
	69	7	61,910	A,B,C
	81	6	62,240	A,B
	2004	8	94,010	A,B,C
T53	2338	7	94,990	A,B,C
	2806	6	96,670	A,B,C
	3507	5	97,880	A,B
	4676	4	97,290	A,C
	138	7	63,180	A,B,C
T28	205	5	68,190	A,B,C
	216	5	68,200	A,B,C
	256	4	68,440	A,B,C
	324	4	73,250	A,C
	342	3	74,140	A,B
	7	6	16,270	A,B,C
T11	9	5	16,880	A,B,C
	10	4	17,150	A,B,C
	13	3	17,210	A,B,C
	14	3	17,990	A,B,C
	21	3	17,890	B,C
	6	5	12,660	B,C
T9	7	4	13,480	B,C
	8	4	13,140	B,C
	10	3	13,540	A,C
	18	3	13,530	C
	56	11	92,400	A,C
T45	57	11	93,800	A,C
	62	10	94,460	A,C
	69	9	94,720	A,C
	79	8	99,130	A,C
	92	7	98,270	A,C
	110	6	99,580	A
	111	6	99,890	A
	138	5	100,840	A
	184	4	102,840	A
	1414	11	47,950	A,B,C
T32	1572	10	48,280	A,B,C
	1768	9	48,460	A,B,C

Problem	Cycle time	Number of workstation	CPU	Resource
T32	2020	7	49,700	A,B,C
	2357	7	49,640	A,B,C
	2828	6	54,530	A,B,C
T89	11	39	183,560	A,B,C
	12	35	199,620	A,B,C
	13	34	240,540	A,B,C
	14	31	192.70	A,B,C
	15	29	227,520	A,B,C
	16	28	197,040	A,B,C
	17	26	207,140	A,B,C
	18	25	266,000	A,B,C
	19	24	194,530	A,B,C
	20	22	225,700	A,B,C
	21	21	270,460	A,B,C
	75	7	333,540	A,B,C
	79	7	385,260	B,C
	83	6	304,380	A,B,C
	87	6	286,060	A,B,C
	92	6	202,940	A,B,C
	97	6	198,730	A,B,C
	103	6	255,900	A,B,C
	110	5	320,790	A,B,C
	118	5	313,000	A,B,C
	127	5	287,730	A,B
	137	4	235,650	A,B
	150	4	233,670	A,B
T11	48	6	16,480	A,B
	62	3	16,630	A,B
	94	3	16,850	A,B
T7	6	4	9,650	A,C
	7	4	9,850	A,C
	8	3	10,030	B,C
	10	3	10,380	A,B,C
	15	2	10,630	B
	18	2	10,790	B,C
T21	14	7	32,490	A,B,C
	15	7	32,940	A,B,C
	21	5	33,100	A,B,C
	26	4	33,520	A,B
	35	4	33,540	A,B
	39	4	33,600	A,B

Problem	Cycle time	Number of workstation	CPU	Resource
T94	176	25	146,560	A,B,C
	183	25	148,320	A,B,C
	192	23	149,460	A,B,C
	201	23	150,120	A,B,C
	211	22	152,420	A,B,C
	222	21	152,600	A,B,C
	234	20	153,120	A,B,C
	248	19	152,460	A,B,C
	263	18	152,320	A,B,C
	281	17	153,150	A,B,C
	301	15	154,560	A,B,C
	324	15	156,450	A,B,C
	351	14	155,230	A,B,C
T25	14	9	38,860	A,B,C
	16	8	39,390	A,B,C
	18	8	39,170	A,B,C
	21	7	39,890	A,C
	25	5	40,340	A,B,C
	32	5	39,810	A,C
T30	25	11	50,860	A,B,C
	27	11	51,810	A,B,C
	30	10	52,430	A,B,C
	33	9	52,840	A,B,C
	36	9	53,470	A,B,C
	41	8	54,440	A,B,C
	47	7	58,570	A,B,C
	54	6	59,170	A,B,C
	75	5	60,360	A,B
T297	1394	61	574,300	A,B,C
	1422	60	526,800	A,B,C
	1452	58	511,600	A,B,C
	1483	58	512,600	A,B,C
	1515	56	507,200	A,C
	1548	56	558,700	A,B,C
	1584	54	525,600	A,B,C
	1620	53	510,200	A,B,C
	1659	51	512,600	A,B,C
	1699	50	533,300	A,B,C
	1742	49	532,750	A,B,C
	1787	47	511,900	A,B,C
	1834	47	539,400	A,C

Problem	Cycle time	Number of workstation	CPU	Resource
T297	1883	45	520,350	A,B,C
	1935	44	513,500	A,B,C
	1991	43	597,340	A,B,C
	2049	42	746,100	A,B,C
	2111	41	598,400	A,C
	2177	40	661,000	A,C
	2247	38	535,600	A,B,C
	2322	37	525,500	A
	2402	35	566,000	A
	2488	35	559,800	A,C
	2580	33	541,600	A
	2680	32	543,100	A
	2787	31	520,800	A,C
T70	160	20	142,540	A,B,C
	168	20	142,950	A,B,C
	176	19	142,330	A,B,C
	185	18	143,250	A,B,C
	195	17	144,500	A,B,C
	207	16	145,200	A,B,C
	220	16	146,460	A,B,C
	234	15	147,450	A,B,C
	251	14	147,720	A,B,C
	270	13	148,740	A,B,C
	293	13	149,200	A,C
	320	11	149,750	A,C
	364	10	150,100	A,B,C
	410	9	150,680	A,C
	468	8	151,350	A,B,C
	527	8	151,540	A,C
T58	54	29	97,500	A,B,C
	56	28	98,200	A,B,C
	58	28	99,300	A,B,C
	60	27	100,500	A,B,C
	62	26	110,400	A,B,C
	65	25	111,300	A,B,C
	68	24	112,100	A,B,C
	71	23	113,220	A,B,C
	74	22	114,870	A,B,C
	78	22	115,230	A,B,C
	82	19	116,140	A,B,C
	86	19	117,200	A,B,C

Problem	Cycle time	Number of workstation	CPU	Resource
T58	92	18	118,770	A,B,C
	97	17	121,100	A,B,C
	104	16	121,150	A,B,C
	111	15	122,200	A,B,C
T75	28	37	174,000	A,B,C
	29	36	178,560	A,B,C
	30	36	177,290	A,B,C
	31	35	182,840	A,B,C
	32	34	181,960	A,B,C
	33	33	180,280	A,B,C
	34	31	184,620	A,B,C
	35	30	183,010	A,B,C
	36	30	183,470	A,B,C
	37	29	184,200	A,B,C
	38	29	184,020	A,B,C
	39	28	182,360	A,B,C
	40	27	184,700	A,B,C
	41	27	186,800	A,B,C
	42	26	187,300	A,B,C
	43	26	188,400	A,B,C
	45	26	190,320	A,B,C
	46	24	178,600	A,B,C
	47	24	177,800	A,B,C
	49	24	178,650	A,B,C
	50	24	180,740	A,B,C
	52	22	182,620	A,B,C
	54	21	184,560	A,B,C
	56	21	185,470	A,B,C