BI-OBJECTIVE ASSEMBLY LINE RE-BALANCING PROBLEM WITH EQUIPMENT ASSIGNMENT

ELİF ŞİMŞEK

JANUARY 2018

BI-OBJECTIVE ASSEMBLY LINE RE-BALANCING PROBLEM WITH EQUIPMENT ASSIGNMENT

A DISSERTATION SUBMITTED TO THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF IZMIR UNIVERSITY OF ECONOMICS

BY
ELIF SIMSEK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
IN THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

JANUARY 2018

M.S. DISSERTATION EXAMINATION RESULT FORM

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Abbas Kenan Çiftçi
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Selin Özpeynirci
Head of Department

We have read the dissertation entitled **"Bi-objective Assembly Line Re-balancing Problem with Equipment Assignment"** completed by **Elif Şimşek** under supervision of **Assoc. Prof. Dr. Selin Özpeynirci** and we certify that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Industrial Engineering.

Assoc. Prof. Dr. Özpeynirci
Supervisor

Examining Committee Members

Date: 18.01.2018

Assoc. Prof. Dr. Özpeynirci
Dept. of Industrial Engineering, IEU

Asst. Prof. Dr. Kamil Erkan Kabak
Dept. of Industrial Engineering, IEU

Asst. Prof. Muhittin Sağnak
Dept. of Philosophy, IKCU

ABSTRACT

BI-OBJECTIVE ASSEMBLY LINE RE-BALANCING PROBLEM WITH EQUIPMENT ASSIGNMENT

Şimşek, Elif
M.S., Department of Industrial Engineering
Graduate School of Natural and Applied Sciences
Supervisor: Assoc. Prof. Dr. Selin ÖZPEYNİRCİ
January 2018, 52 pages

In this thesis, a bi-objective assembly line re-balancing problem with equipment assignment is considered. We are given assignment of tasks to workstations with required tools and a balanced assembly line. Any disruptions in one or more workstations make the current solution infeasible. The tasks should be reassigned to the remaining workstations with required equipment. We may need to purchase new equipment if the required equipment of some tasks do not exist on the workstations they are assigned to. We consider two objectives while re-balancing the assembly line: minimizing cycle time and minimizing cost of purchasing new equipment. The aim of this study is to generate efficient solutions and determine the most preferred solution.

All non-dominated objective vectors are generated using two methods: Traditional ϵ-Constraint Method and Augmented ϵ-Constraint Method. An algorithm that works interactively with the decision maker is used to find the most preferred solution. Computational experiments are carried out in order to measure the performance of suggested methods and the solutions are reported.

Keywords: Assembly Line, Assembly Line Re-balancing, ϵ-Constrained Method, Interactive Method, Multi-objective Optimization Problem

ÖZ

EKİPMAN ATAMASI İLE İKİ AMAÇLI MONTAJ HATTI YENİDEN DENGELEME PROBLEMİ

Şimşek, Elif
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Fen Bilimleri Enstitüsü
Danışman: Doc. Dr. Selin ÖZPEYNİRCİ
Ocak 2018, 52 sayfa

Bu tezde, ekipman ataması ile iki amaçlı montaj hattı yeniden dengeleme problemi ele alınmıştır. Elimizde gerekli ekipmanla birlikte istasyonlara atanmış işler ve dengelenmiş bir montaj hattı bulunmaktadır. Bir veya daha fazla istasyonda meydana gelen bozulma elimizdeki sonucu kullanılamaz hale getirir. İşlerin gerekli ekipmanla birlikte kalan istasyonlara yeniden atanması gerekmektedir. Eğer işlerin atandığı istasyonlarda gerekli ekipman yer almıyorsa yeni ekipman satın almamız gerekmektedir. Montaj hattını yeniden dengelerken iki amacımız vardır: çevrim süresini en azlamak ve yeni alınan ekipmanın maliyetini en azlamak. Bu çalışmanın amacı bastırılmamış çözümleri yaratmak ve en çok tercih edilen çözümü bulmaktır.

İki yöntem kullanılarak tüm bastırılmamış çözümler yaratılmıştır: Geleneksel ϵ-Kısıt Yöntemi ve Artırılmış ϵ-Kısıt Yöntemi. Yaratılan çözümler arasından en çok tercih edilen çözümü bulmak için karar verici ile etkileşimli olarak çalışan bir algoritma kullanılmıştır. Önerilen yöntemlerin performanslarını ölçmek amacıyla deneyler yapılmış ve sonuçlar raporlanmıştır.

Anahtar Kelimeler: Montaj Hattı, Montaj Hattı Yeniden Dengeleme, ϵ-Kısıt Yöntemi, Interaktif Yöntem, Çok Amaçlı Optimizasyon Problemi

ACKNOWLEDGEMENT

This study focuses on assembly line re-balancing problem with tool assignment. Two different algorithms are applied to tested performance of the method, and the interactive method is applied for selecting the most preferred solution.

I would like to express our sincere gratitude to our supervisor, Assoc. Prof. Dr. Selin ÖZPENİRCİ for giving me the opportunity to study with themselves and for their encouragement, supervision, guidance and suggestions.

I would like to thank each of my college lecturers for all the information they have provided in my graduate education life.

I would also like to thank to my family who always encourage and support me to every difficulty in my life. Their love was the only source I needed and it was good to know.

I would like to thank to my dear friend Kasım Okan ÖZDEMİR for the support in the thesis completion and his help with coding.

Finally, I would like to thank members of the review committee for their valuable contributions to this work.

Elif ŞİMŞEK

January 2018, Balçova

Contents

Abstract iii

Öz iv

Acknowledgement v

Table of Contents vi

List of Figures viii

List of Tables ix

1 INTRODUCTION 1

2 LITERATURE REVIEW 3

2.1 LITERATURE ON ASSEMBLY LINE BALANCING 3

2.1.1 TYPE I ASSEMBLY LINE BALANCING PROBLEM 3

2.1.2 TYPE II ASSEMBLY LINE BALANCING PROBLEM 4

2.2 LITERATURE ON ASSEMBLY LINE RE-BALANCING 4

2.2.1 LITERATURE ON STOCHASTIC ASSEMBLY LINE RE-BALANCING 5

2.2.2 LITERATURE ON DETERMINISTIC ASSEMBLY LINE RE-BALANCING 6

2.3 LITERATURE ON MULTIPLE OBJECTIVE PROBLEM AND ϵ-CONSTRAINT METHOD 6

3 PROBLEM DEFINITION 9

3.1 MATHEMATICAL MODEL 10

4 SOLUTION APPROACH 12

4.1 TRADITIONAL ϵ-CONSTRAINT METHOD 12

4.2 THE AUGMENTED ϵ-CONSTRAINT METHOD (AUGMECON) 13

4.3 THE INTERACTIVE METHOD BASED ON AUGMECON 15

5 COMPUTATIONAL EXPERIMENTS 21

5.1 EXPERIMENT DESIGN 21

5.2 RESULTS OF THE EXPERIMENT 24

5.3 ILLUSTRATIVE EXAMPLE 28

6 CONCLUSION 41

7 APPENDIX 42

7.1 PRECEDENCE DIAGRAMS 42

7.2 CODES 47

7.2.1 TRADITIONAL ϵ - CONSTRAINT MODEL 47

7.2.2 AUGMENTED ϵ - CONSTRAINT MODEL 48

List of Figures

1 Flowchart of the interactive method based on AUGMECON [Mavrotas, 2009] . . 19

2 Graphical representation of the interactive use of AUGMECON [Mavrotas, 2009] 20

3 The precedence network of the sample problem 28

4 The initial configuration of the assembly line in the Gunther et al. (1983) 29

5 The calculated points in the Traditional ϵ-constraint Method 30

6 The efficient points in the AUGMECON Method 30

7 The precedence diagram of Mertens sample 42

8 The precedence diagram of Mansoor sample 42

9 The precedence diagram of Mitchell sample 43

10 The precedence diagram of Kilbridge-Wester sample 44

11 The precedence diagram of Hahn sample 45

12 The precedence diagram of Tonge sample 46

List of Tables

1 The parameters for seven experiment data 22

2 Generated equipment costs in the experiment 23

3 α_1 and α_2 values used in decision maker's utility function 23

4 The experiment results of the two approaches: Traditional ϵ-Constraint Method and The Augmented ϵ-Constraint Method 25

5 Savings on each experiment with AUGMECON compared to the Traditional ϵ-Constraint Method 26

6 The results of the interactive method for each experiment at $\alpha_1 = 0.4$, $\alpha_2 = 0.6$. 26

7 The results of the interactive method for each experiment at $\alpha_1 = 0.7$, $\alpha_2 = 0.3$. 27

8 The results of the interactive method for each experiment at $\alpha_1 = 0.2$, $\alpha_2 = 0.8$. 27

9 The task and equipment assignment of the initial configuration of the sample . . 29

10 The efficient points of the sample 31

11 L_p distances between the points at iteration 1 33

12 L_p distances between the points at iteration 2 35

13 L_p distances between the points at iteration 3 36

14 L_p distances between the points at iteration 4 38

15 L_p distances between the points at iteration 5 39

16 The results of the interactive method on Gunther sample 40

1 INTRODUCTION

Assembly line balancing problem is assigning individual processing and assembly tasks to the workstations. [Pape, 2015] Therefore, the total time required at each workstation is approximately the same and reaching perfect balance is nearly impossible. [Raj et al., 2016] There are some points to be considered in the assembly line balancing problem:

- There is a required order of tasks, called precedence constraints since sequence of tasks is restricted.
- There is a production rate needed for designing the line to meet demand.
- The task times may be deterministic or stochastic.
- The line may have serial layout or parallel layout with one-sided or two sided workstations.
- Single model assembly line or mixed model assembly line may be considered.

There are many assembly line balancing studies in the literature from 1955 to present. Although there are many categorization methods for assembly line balancing, the fundamental classification is based on objective function of the problem. There are two types of assembly line balancing problems according to the objective function: Assembly Line Balancing Problem Type I aims to minimize the number of workstations for a given cycle time and is usually considered before the configuration of an assembly line. [Pape, 2015] However, Assembly Line Balancing Problem Type II aims to minimize the cycle time for a given number of workstations for configured assembly lines with a new problem. [Simaria and Vilarinho, 2004] Different approaches are suggested in the literature to solve the assembly line balancing problem.

There are a few assembly line re-balancing problem studies in the literature whereas there is practical implementation. Some changes may occur in input parameters such as change in demand pattern, change in task times, technological restrictions or workstation breakdowns. Thus, re-balancing of the line is more convenient than balancing problem each time when the input parameters changes. [Yang et al., 2013]

Multiobjective optimization involves minimizing or maximizing multiple objective functions subject to a set of constraints. There does not exist a single solution since each objective should be optimized. Therefore, there exists a set of Pareto optimal solutions. A solution is called nondominated or Pareto optimal if none of the objective functions can be improved in value without degrading one or more of the other objective values. [Saif et al., 2014] Example problems include analyzing design tradeoffs, selecting optimal product or process designs, or any other application with tradeoffs between two or more conflicting objectives.

For multi-objective linear programs, a solution is obtained which satisfies the decision maker's preferences, and optimization from the decision maker's point of view is considered. There are lots of studies in the literature since 1970 and different algorithms are proposed in order to solve this problem. According to our literature review, particle swarm optimization algorithm, colony optimization algorithm and genetic algorithm are frequently used to determine the optimal solutions. [Scholl et al., 1995]

In this thesis, we study a bi-objective assembly line re-balancing problem with equipment assignment. When one or more workstations are disrupted, the tasks should be reassigned to the nondisrupted workstations with respect to performance measures. If required, the required equipment of tasks are purchased to catch the re-balance. After one or more workstations are disrupted, the tasks on the disrupted workstations have to be moved to the nondisrupted workstations with their required equipment. Performance measures are cycle time and cost of purchasing new equipment in this study. Since the problem is a multi-objective problem, if the cycle time is decreased, the re-balancing cost is to increase. The aim of the problem is generating the efficient solutions in order to obtain the most preferred solution. We assume that the utility function of the decision maker is unknown. Therefore, decision maker selects the optimal solution in the generated Pareto frontier according to his/her preferences.

There are only a few studies about assembly line re-balancing in the literature. Also, after the literature review, we propose two algorithms to rebalance the assembly line with equipments on the assembly line. Traditional ϵ-Constrained Method and Augmented ϵ-Constrained Method are applied in this study, and their results and performances are compared to each other regarding the computational study. The cycle time and the cost of equipments purchased after disruption are minimized during the re-balancing. After the efficient points are obtained with these methods; the most preferred solution should be determined. The Interactive Method based on AUGMECON is applied in order to get optimal solution according to the decision maker's preferences.

This thesis includes the following sections to explain the assembly line re-balancing problem with equipment assignment: In Section 2, the related literature on the assembly line balancing problems, re-balancing problems and multi-objective optimization problems are reviewed. In Section 3, re-balancing problem is defined and the mathematical model is given. In Section 4, the Traditional ϵ-Constrained Method and the Augmented ϵ-Constrained Method are presented, respectively. Also, the Interactive method is explained to reach the most preferred solution. In Section 5, experiment design is given and the results of the computational experiments are discussed. One problem instance is presented as an illustrative example to show all calculation procedures. In Section 6, concluding remarks and possible future study are explained.

2 LITERATURE REVIEW

In this chapter, the studies on assembly line balancing problem and assembly line re-balancing problem are revival in the literature. Equipment assignment is reviewed for the assembly line problem. The details of multi-objective problems are searched, and suggested heuristic methods are analyzed in order to solve these problems.

2.1 LITERATURE ON ASSEMBLY LINE BALANCING

Assembly line balancing problems (ALBP) are called Type I and Type II ALBP. Type I ALBP is characterized by the minimization of the number of workstations at desired cycle time. However, Type II ALBP is characterized by the minimization of the cycle time at desired number of workstations.

2.1.1 TYPE I ASSEMBLY LINE BALANCING PROBLEM

In type I problems, the cycle time and the production rate have to be predetermined. Therefore, they are more frequently used in the design of a new assembly line since the demand can be easily forecasted.

[Sarin et al., 1999] consider a single model stochastic assembly line balancing problem for minimizing the total labor cost and the expected incompletion cost arising from tasks in the prescribed cycle time. Solution procedure starts with dynamic programming method and continues with branch and bound method to generate improved solution. After the improvements, the tasks associated with less workstation have smaller cost values. [Sarin et al., 1999]

[Pape, 2015] considers heuristic methods and lower bounds for solving simple assembly line balancing problem of Type I. In this paper, 12 heuristics and 9 lower bounds are provided from the literature is published before 2011. Genetic algorithm, a differential evolutionary algorithm (DEA), ant colony optimisation (ACO), tabu search method, bounded dynamic programming, station-oriented, depth-first, bidirectional branch-and-bound algorithm (SALOME) and random search are explained with details. Heuristics and lower bounds are compared with computational results. They improve dynamic programming and a tabu search approach and conclude that these are also identified as the most effective heuristics. [Pape, 2015]

[Grzechca, 2014] proposes a heuristic in order to reduce the number of workstations and to improve productivity for different structural problems such as serial line, U-line and parallel line. Heuristics are based on genetic algorithms, tabu search and simulated annealing techniques. Line efficieny, smoothness index and line time are used and compared for performance measures. Paralellizing of serial lines reduces the number of workstations according to the calculation results. [Grzechca, 2014]

2.1.2 TYPE II ASSEMBLY LINE BALANCING PROBLEM

Type II problems deal with the maximisation of the production rate of an existing assembly line.

[Simaria and Vilarinho, 2004] consider a mixed model assembly line problem with parallel workstations in order to maximize the production rate of the assembly line for a prespecified number of operators and minimizing cycle time. This study presents a mathematical programming model, and an iterative genetic algorithm based procedure is developed to solve the problem with reflecting zoning constraints, workload balancing and controlling the generation of parallel workstations by the decision maker. The model also balances the workloads within the workstations for the different models to be assembled. After the computational calculations, the heuristic performance is good and there is a slight increase in the workload balance. [Simaria and Vilarinho, 2004]

[Raj et al., 2016] consider an approach to solve mixed model assembly line balancing problem with parallel workstations. In the study, genetic algorithm based procedure is proposed in order to optimize cycle time. The genetic algorithm procedure is run for 100 iterations in the numerical example. After 16 iterations, there is no change in the cycle time. [Raj et al., 2016]

[Lyu, 1997] proposes the single-run optimization approach that is based on a simulation model for stochastic assembly line balancing problem. The study considers Perturbation-Analysis- Robbins-Monro-Single-Run Algorithm (PARMSR) since this algorithm is claimed to be effective and efficient for these type of test problems. The proposed algorithm has a much faster execution time than two others. [Lyu, 1997]

2.2 LITERATURE ON ASSEMBLY LINE RE-BALANCING

An existing assembly line has to be rebalanced if there are breakdowns of a workstation on the assembly line and some changes in the input parameters. The occurrences of variety of modifications in the input parameters are given below for re-balancing of the existing assembly line problems:

- Changes in properties of the product, addition or removal of tasks and precedence relationships.
- Increasing or decreasing of tasks time for adopting new equipments
- Changes in cycle time due to variations in market demand

There are a few studies in the literature about re-balancing of assembly lines and these studies are classified as stochastic and deterministic regarding to the nature of the task times.

2.2.1 LITERATURE ON STOCHASTIC ASSEMBLY LINE RE-BALANCING

[Gamberini et al., 2006], [Gamberini et al., 2009] and [Celik et al., 2014] are considered assembly line re-balancing studies with stochastic task times in the literature. These three studies review different heuristic methods with parameters and performance measures.

[Gamberini et al., 2006] consider single model assembly line re-balancing problem and develop a new heuristic method that is based on "Technique for Order Preference by Similarity to Ideal Solution" and the heuristic approach developed by [Kottas and Lau, 1973] in order to solve assembly line re-balancing problem. The study deal with the minimization of the unit labour and expected unit incompletion costs and the minimization of tasks re-assignment. The study assumes that total expected completion cost is the sum of the total labor cost and total expected incompletion cost uses the idea of [Kottas and Lau, 1973]. A multi-objective heuristic algorithm is suggested which is incorparated in [Kottas and Lau, 1973] heuristic algorithm developed for solving stochastic assembly line balancing problems and [Hwang and Yoon, 1981] algorithm for order preference by similarity to ideal solution. The similarity index is used and the computational calculations are made due to changes of precedence relationships and the performance is tested according to 2160 test results. The suggested algorithm provides cost reduction and increases similarity index. [Gamberini et al., 2006]

[Gamberini et al., 2009] propose a multiple single-pass heuristic algorithm to find Pareto frontier of the problem. There are two objectives. One objective is optimizing the total expected cost for a new assembly line, the other is optimizing the similarity between the new line and the old line. Multiple single-pass heuristic algorithm includes different heuristic procedures to solve the re-balancing problem. Multi-objective genetic algorithm is developed to compare with multiple single-pass heuristic algorithm. The experiment results showed that multiple single-pass heuristic algorithm is more effective than genetic algorithm for obtaining consistent results in terms of similarity optimisation. [Gamberini et al., 2009]

[Celik et al., 2014] consider a u-line re-balancing problem due to demand variations, changes in product design or changes in task times. This study proposes an ant-colony optimization algorithm to solve U-line re-balancing problem with stochastic task times. The objective is the minimization of total cost which includes task transposition costs, workstation opening or closing costs and workstation operating cost over a definite planning horizon. After computational experiments, 6600 re-balancing solutions are obtained, and they fiund that average of CPU times is equal to different α levels, number of periods and demand varying situations.

[Celik et al., 2014]

2.2.2 LITERATURE ON DETERMINISTIC ASSEMBLY LINE RE-BALANCING

[Grangeon et al., 2011], [Yang et al., 2013] and [Zha and Yu, 2014] propose heuristic methods with deterministic task times using different performance measures.

[Grangeon et al., 2011] propose a mixed model assembly line re-balancing problem in an automative firm. The assembly line manager assigns tasks to workstations using cycle time, operator time, section length, working height and precedence constraints among the operators. The study explaines a heuristic solution procedure which has three steps in order to obtain feasible solution, minimize the number of workstations and improves a feasible solution by smoothing the workload of the workstations. [Grangeon et al., 2011]

[Yang et al., 2013] address a mixed-model assembly line re-balancing problem with seasonal demand. This paper concerns reassigning assembly tasks and operators to the workstations of given cycle time. The objectives are minimizing the number of stations, workload variation at each station for different models and cost. A multi-objective genetic algorithm is proposed to solve this problem. The performance of proposed algorithm is tested on 23 problems, and obtained results show better solutions than the others in the literaure. [Yang et al., 2013]

[Zha and Yu, 2014] study U-line re-balancing problem with respect to minimizing moving cost of machines and labor cost. A new hybrid algorithm is developed by combining ant colony optimization and filtered by beam search to solve the problem. In the procedure, each ant searches several nodes for one step and chooses the best one at a given probability. The proposed algorithm shows effective performance than the others in the literature for solving U-line re-balancing problems according to computational calculations. [Zha and Yu, 2014]

2.3 LITERATURE ON MULTIPLE OBJECTIVE PROBLEM AND ϵ-CONSTRAINT METHOD

There are several studies on multiobjective assembly line balancing problem in the literature. These types of problems are reviewed in this study and they are briefly summarized with proposed algorithm.

[Yuguang et al., 2016] consider a mixed-model assembly line on which different hull blocks can be assembled at the same time. The objectives of the problem are minimizing the cycle time, the static workload variations, the dynamic workload variations and the multi-station associated complexity. The discrete particle swarm optimization algorithm is generated based on the stratified optimization idea. The performance of the proposed algorithm is examined over several test problems in terms of solution quality and running time. [Yuguang et al., 2016]

[Wang et al., 2011] proposed genetic algorithms to help designers make decisions on multi-objective optimization problems to minimize complexity and maximize market share. [Wang et al., 2011]

[Yoosefelahi et al., 2012] study Type II robotic assembly line balancing problem with the aim of minimizing cycle time, robot setup costs and robot costs. A new mixed-integer linear programming model is developed to solve the problem. Multi-objective evolution strategies are run since the problem is NP-hard. The proposed multi-objective evolution strategies are more efficient according to the computational results. [Yoosefelahi et al., 2012]

[Chutima and Chimklai, 2012] consider a particle swarm optimization algorithm with negative knowledge in order to solve multi-objective two-sided mixed-model assembly line balancing problems. The aims are optimizing the number of mated-stations, the number of workstations and work relatedness, and workload smoothness. After applying algorithm, improved Pareto frontiers are obtained, but longer computation times are required. [Chutima and Chimklai, 2012]

[Rada-Vilela et al., 2013] presents the configuration of an assembly line problem which is commonly found in the automotive industry. The objectives of these type of problems are minimizing the number of workstations and the required physical area. Multi-Objective Ant Colony Optimization algorithm is used to solve the assembly line problem. [Rada-Vilela et al., 2013]

[Saif et al., 2014] consider a single model assembly line balancing problem with uncertain task times and multiple objectives. The Pareto based artificial bee colony algorithm is proposed to get Pareto solution for minimizing cycle time, smoothness index and maximizing the probability that completion time of tasks will not exceed the cycle time. Computational result shows that the proposed algorithm outperforms NSGA II in terms of the quality of Pareto solutions and computational time. [Saif et al., 2014]

[Mavrotas, 2009] proposes the Augmented ϵ-Constraint Method to omit weakly Pareto optimal solutions and accelerates the whole process by avoiding redundant iterations. Also, an interactive approach that is based on AUGMECON is proposed in the paper and the most preferred Pareto optimal solution is generated. [Mavrotas, 2009]

[Bérubé et al., 2009] deal with bi-objective combinatorial optimization problems with integer objective values by solving single objective subproblems. Pareto frontier of the Travelling Salesman Person Problem with profits is obtained to minimize the traveled costs and maximize the collected prize. [Bérubé et al., 2009]

3 PROBLEM DEFINITION

In this chapter, the problem description, notations and mathematical model are presented.

Consider a single model assembly line. The assembly line is serial with one-sided workstations. There are n tasks assigned to $w\prime$ workstations. After disruption of some workstations with their equipments, the tasks are assigned to the non-disrupted workstations. Disruption is defined as machine breakdowns in the workstations. [Celik et al., 2014] Let w be the number of non-disrupted workstations in the new configuration. There are q equipments required for assembly lines. Each task requires a subset of equipments to be processed. After disruption, equipments are assigned to the non-disrupted workstations in the new configuration. If required, new equipments purchased.

The assembly line is assumed to be deterministic and static. Also, initial configuration of the assembly line is known. Processing times of the tasks, precedence relations of the tasks, cycle time and cost of the equipment define the problem data. After disruption, all tasks are reassigned to remaining workstations with required equipment.

The indices and parameters are defined below:

Indices:

i,h: task index $\{1, 2, ..., n\}$

j: equipment index $\{1, 2, ..., q\}$

k: workstation index after disruption $\{1, 2, ..., w\}$

Parameters:

$$l_{ij} = \begin{cases} 1, & \text{if task } i \text{ requires equipment } j \\ 0, & \text{otherwise} \end{cases}$$

$$P_{ih} = \begin{cases} 1, & \text{if task } i \text{ is immediately performed before task } h \\ 0, & \text{otherwise} \end{cases}$$

$$u_{jk} = \begin{cases} 1, & \text{if equipment } j \text{ is already exists on workstation } k \\ 0, & \text{otherwise} \end{cases}$$

t_i= processing time of task i

EC_j= cost of equipment j

We consider a bi-objective assembly line re-balancing problem with the aim of minimizing cycle time and total cost of purchasing new equipments.

3.1 MATHEMATICAL MODEL

In this section, the bi-objective mathematical model used to solve assembly line re-balancing problem with equipment assignment is presented.

The following binary decision variables are used in the mathematical model:

$$x_{ik} = \begin{cases} 1, & \text{if task } i \text{ is assigned to workstation } k \text{ after the disruption} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{jk} = \begin{cases} 1, & \text{if equipment } j \text{ is purchased to be used on workstation } k \text{ after the disruption} \\ 0, & \text{otherwise} \end{cases}$$

for $i = \{1, 2, ..., n\}$, $j = \{1, 2, ..., q\}$ and $k = \{1, 2, ..., w\}$

In the assembly line re-balancing problem with equipment assignment, we consider the following two objectives:

$$Minimize \quad CT \tag{1}$$

$$Minimize \quad \sum_{j=1}^{q} \sum_{k=1}^{w} EC_j y_{jk} \tag{2}$$

The first objective represents minimizing cycle time and the second objective is minimizing the cost of equipments purchased after disruption. Constraint sets of assembly line re-balancing problem with equipment assignment are defined below:

$$\sum_{k=1}^{w} x_{ik} = 1, \forall i \tag{3}$$

$$\sum_{i=1}^{n} t_i x_{ik} \leq CT, \forall k \tag{4}$$

$$\sum_{k=1}^{w} k x_{ik} \leq \sum_{k=1}^{w} k x_{hk}, \forall (i,h) \| P_{ih} = 1 \tag{5}$$

$$x_{ik} \leq u_{jk} + y_{jk}, \forall (i,j,k) \| l_{ij} = 1 \tag{6}$$

$$x_{ik} \in \{0,1\}, \forall (i,k) \tag{7}$$

$$y_{jk} \in \{0,1\}, \forall (j,k) \tag{8}$$

Constraint set 3 ensures that each task is assigned to exactly one workstation. Constraint set 4 defines the cycle time (CT) as the maximum workload among all workstations. Constraint set 5 defines the precedence relations between tasks i and h. In accordance with constraint 5, if task i immediately precedes task h, then task i cannot be assigned to a later workstation than station of task h. Constraint set 6 ensures that a task can be assigned to a workstation if its required equipment is loaded on that station. Constraint sets 7 and 8 define x_{ik} and y_{jk} as binary variables.

4 SOLUTION APPROACH

In this chapter, two methods that are used to solve bi-objective assembly line re-balancing problem with equipment assignment are explained in detail. These methods are Traditional ϵ-Constraint Method and The Augmented ϵ-Constraint Method (AUGMECON) developed by Mavrotas (2009).

4.1 TRADITIONAL ϵ-CONSTRAINT METHOD

Traditional ϵ-Constraint Method is developed for multi-objective problems that transforms one of the objectives into a constraint. This incorporates other constraints in the mathematical model in order to optimize the problem [Steuer, 1986].

In the ϵ-Constraint Method, efficient solutions of the problem are obtained by varying right hand side of constraints. Obtained points include all solution points that are defined as efficient and non-efficient solutions. At each iteration ϵ value is incremented by 1.

The stepwise procedure of the traditional ϵ-constraint method used in this study is defined below:

Step 0: Set $\epsilon = 0$ and let CT be the cycle time of the solution before disruption.

Step 1: Construct and solve the mathematical model 3.1 (ϵ) below:

$$Minimize \quad \sum_{j=1}^{l} \quad \sum_{k=1}^{w} EC_j y_{jk} \tag{9}$$

subject to:

$$\sum_{k=1}^{w} x_{ik} = 1, \forall i \tag{10}$$

$$\sum_{i=1}^{n} t_i x_{ik} \leq CT + \epsilon, \forall (k, \epsilon) \tag{11}$$

$$\sum_{k=1}^{w} k x_{ak} \leq \sum_{k=1}^{w} k x_{bk}, \forall (a, b) \| P_{ab} = 1 \tag{12}$$

$$x_{ik} < u_{jk} + y_{jk}, \forall (i, j, k) \| l_{ij} = 1 \tag{13}$$

$$x_{ik} \in \{0, 1\}, \forall (i, k) \tag{14}$$

$$y_{jk} \in \{0, 1\}, \forall (j, k) \tag{15}$$

Step 2: Set $\epsilon = \epsilon + 1$ and go to Step 1.

Traditional ϵ- Constrained Method generates nondominated objective vectors with its efficient solutions. [Mavrotas, 2009] The most preferred solution can be found by using a method developed to solve multi-criteria choice problem.

4.2 THE AUGMENTED ϵ-CONSTRAINT METHOD (AUGMECON)

The Augmented ϵ-Constraint Method avoids the generation of nonefficient solution and solution procedures proceeds by avoiding unnecessary iterations. As a result of the AUGMECON procedure, efficient frontier is obtained and an interactive method is suggested to determine optimal solution by the decision maker.[Mavrotas, 2009]

The payoff table is determined by simply calculating the individual optima of the objective functions. The payoff table represents all solution points in a tabular form during the calculation. The obtained optimal solution in the payoff table is not efficient solution, but it is weakly efficient solution. In order to omit these weakly efficient solutions, the objective function constraints are transformed to equalities by incorporating the appropriate slack or surplus variables as equation with ϵ coefficient in equation 17. Also, these slack or surplus variables and ϵ are added as the second term to objective function 16. According to these corrections, the program has to generate only efficient solutions and the problem model becomes as given below:

$$Minimize \quad \sum_{j=1}^{q} \quad \sum_{k=1}^{w} EC_j y_{jk} + \epsilon s \tag{16}$$

$$CT + s = e_2 \tag{17}$$

where ϵ is a small number and e_2 is the constrained objective value that is used in the last iteration.

By modifying only the specific objective functions and constraints in the code, Pareto optimal solutions are obtained. After these changes, obtained mathematical model is presented below.

$$Minimize \quad \sum_{j=1}^{q} \quad \sum_{k=1}^{w} EC_j y_{jk} + \epsilon s \tag{18}$$

subject to:

$$\sum_{k=1}^{w} x_{ik} = 1, \forall i \tag{19}$$

$$\sum_{i=1}^{n} t_i x_{ik} \leq CT, \forall k \tag{20}$$

$$CT + s = e_2 \tag{21}$$

$$\sum_{k=1}^{w} k x_{ik} \leq \sum_{k=1}^{w} k x_{hk}, \forall (i,h) \| P_{ih} = 1 \tag{22}$$

$$x_{ik} < u_{jk} + y_{jk}, \forall (i,j,k) \| l_{ij} = 1 \tag{23}$$

$$x_{ik} \in \{0,1\}, \forall (i,k) \tag{24}$$

$$y_{jk} \in \{0,1\}, \forall (j,k) \tag{25}$$

Advantages of the Augmented ϵ-Constraint Method:

- AUGMECON method is beneficial for solving large scale problems when there are several objective functions in the problem.
- This technique saves a lot of time in multi-objective problems by reducing the number of models solved, due to the computational effort.
- AUGMECON method omits nonefficient solutions and obtains only Pareto optimal solutions.
- The number of iterations are decreased for each experiment by the Augmented ϵ-Constraint Method.
- Different Pareto optimal solutions can be found by changing ϵ values.
- This technique can be used for problems having convex and nonconvex spaces alike.

Disadvantage of the Augmented ϵ-Constraint Method:

- In the AUGMECON method, as the number of objectives increases, there are more ϵ values which require more information from the user. [Mavrotas, 2009]

4.3 THE INTERACTIVE METHOD BASED ON AUGMECON

There are two aspects in multi-objective optimization problem: optimization and decision support. Generally, a decision maker should provide additional preference information in order to identify the "most preferred" solution. Also, the decision maker selects his/her most preferred solution among generated Pareto optimal solutions.

In this thesis, we modify an interactive approach based on AUGMECON to identfy decision maker's most preferred Pareto optimal solution developed by [Mavrotas, 2009]. The interactive approach works as follows:

- AUGMECON is applied in the multi-objective problem, and Pareto optimal solutions are obtained.

- The derived Pareto optimal solutions are filtered down, using the Forward and Reverse Filtering Process [Steuer, 1986] that is described below:

π_i is the range equalization weight associated with the i^{th} component of the vectors being filtered that is calculated by equation 26. The goal of the π_i weights is to balance the ranges of the components of the vectors in V. Let R_i be the *range* of the i^{th} component of the vectors in V. It is used to calculate the range of equalization and calculated by equation 27. [Mavrotas, 2009]

$$R_i = max_{v \in V}\{v_i\} - min_{v \in V}\{v_i\} \tag{26}$$

$$\pi_i = \frac{1}{R_i}\left[\sum_{j=1}^{q} \frac{1}{R_j}\right]^{-1} \tag{27}$$

Filtering relationship is used to compare the weighted L_p distances between points and given below:

$$\left[\sum_{i=1}^{q} \left(\pi_i \mid v_i^t - v_i^h \mid\right)^p\right]^{1/p} < d \tag{28}$$

where

t: the identification superscript of a vector not retained by the filter

h: the identification superscript of a vector retained by the filter

d: the test distance parameter

q: the length of the vectors being filtered

p: the metric parameter

After the calculation of the weighted distance measures (D), Pareto optimal solutions are filtered down with the following rule:

$$\begin{cases} \text{Total number of points} \geq 10 & \text{5 points are selected} \\ 5 \leq \text{Total number of points} < 10 & \text{3 points are selected} \\ \text{Total number of points} < 5 & \text{2 points are selected} \end{cases}$$

For selecting points, the test distance parameter (d) is calculated by using weighted distance measures (D) in equation 29.

$$d = \left\lceil \frac{D}{(\text{Number of Selected Points} - 1)} \right\rceil \tag{29}$$

- The selected points are shown to the decision maker and the decision maker selects his/her most preferred one.

  Underlying value function of decision maker (α_1 and α_2)is assumed as Equation 30, and the values of the selected points are calculated by using this underlying value function 30 to simulate decision maker's preferences. In this study, α_1 indicates the weight of cycle time and α_2 indicates the weight of equipment cost.

$$\left[\alpha_1 \frac{CT - CT_{min}}{CT_{max} - CT_{min}} - \alpha_2 \frac{Cost - Cost_{min}}{Cost_{max} - Cost_{min}}\right] \tag{30}$$

- After the selected preferred solution, interactive method uses this data in order to narrow down the search field around the selected solution. It is associated with objective functions to calculate lower bounds for new Pareto solution generation.

  Suppose, decision maker selects one point using underlying value function 30 and lower bounds are calculated for new iteration using.

$$LB_j^{(i+1)} = z_{*j}^{(i)} - a_i z_{*j}^{(i)} - z_j^{min} \tag{31}$$

Figure 1 represents steps of the interactive process for selecting optimal solution by the decision maker. a is the contraction parameter that takes values in [0,1] and controls the rate of search space. Figure 2 shows how interactive approach of AUGMECON works with iteration steps.

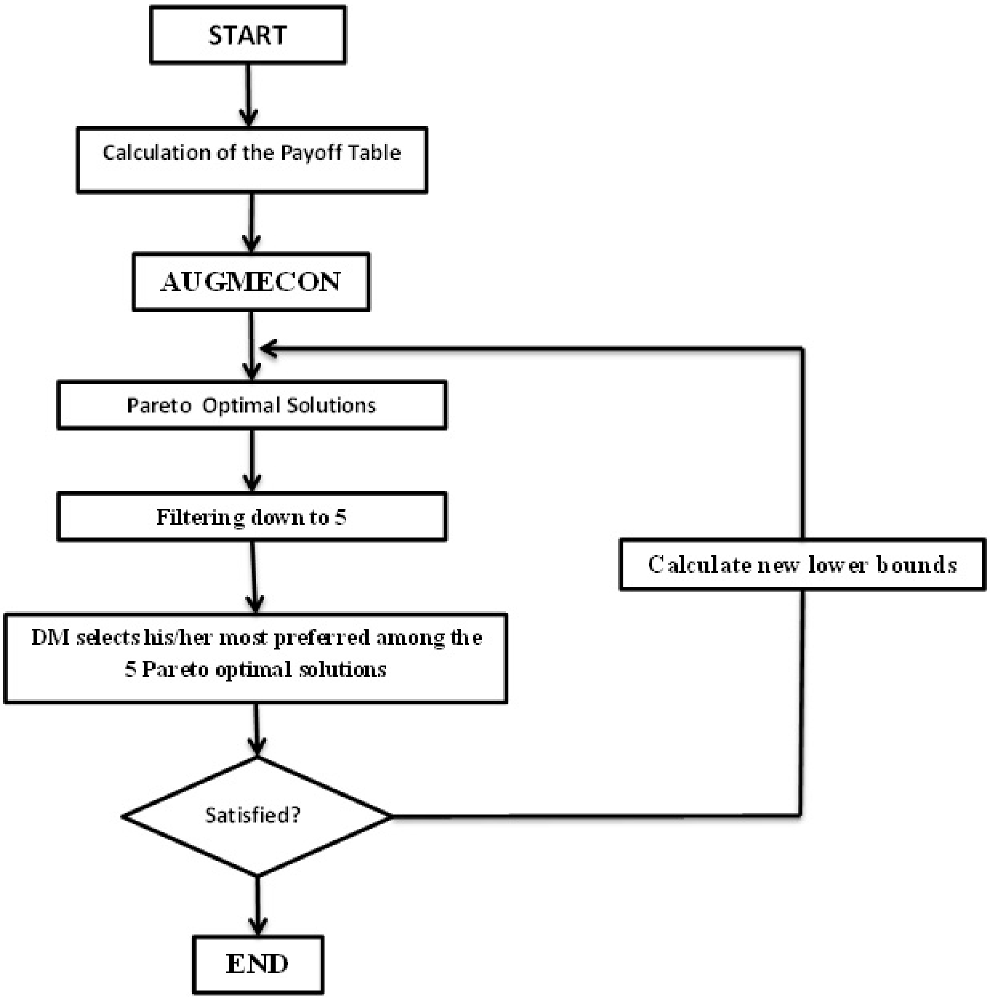

Figure 1: Flowchart of the interactive method based on AUGMECON [Mavrotas, 2009]

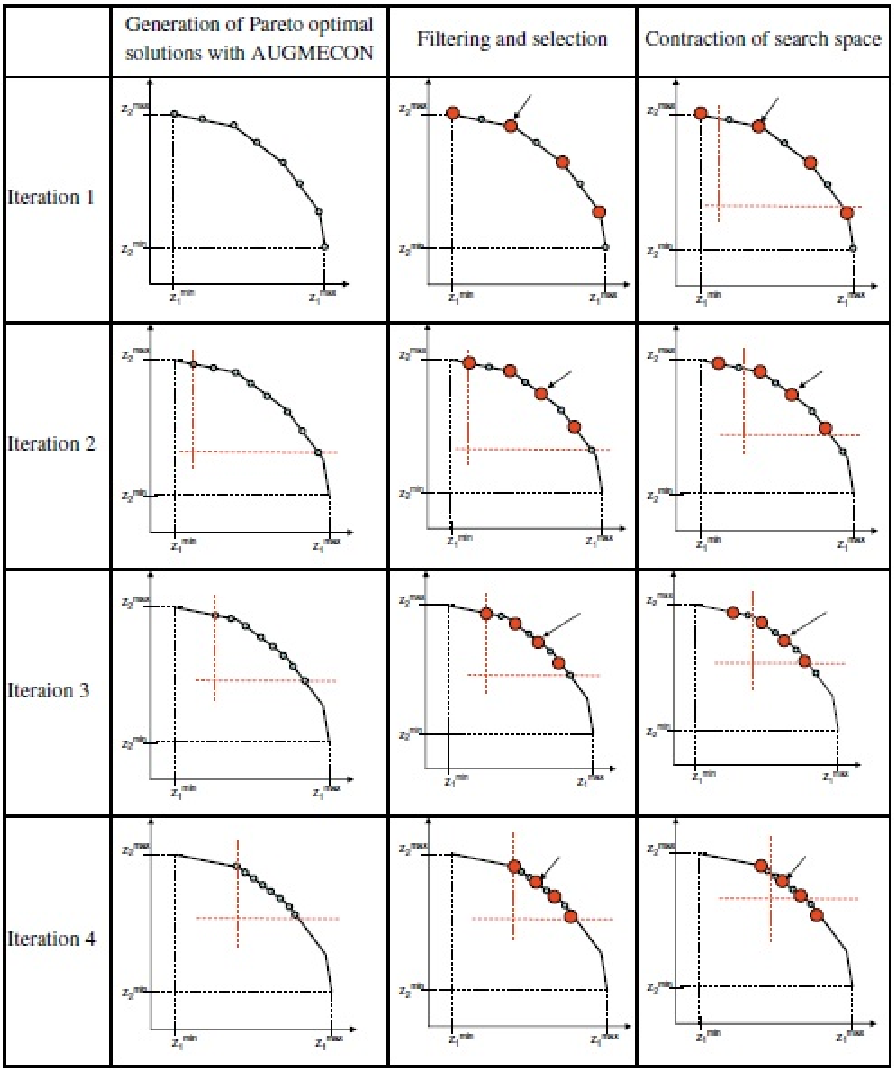

Figure 2: Graphical representation of the interactive use of AUGMECON [Mavrotas, 2009]

5 COMPUTATIONAL EXPERIMENTS

Computational experiments to evaluate the performance of the Augmented ϵ-Constraint Method (AUGMECON) and Traditional ϵ-Constraint Method are carried out.

In this section, firstly, experimental design is described and results of the experiments are given and each of the settings is discussed with run solutions. Also, an illustrative example given by Gunther et al. (1983) with $n = 35$ tasks is given to explain how the proposed method works.

In this thesis, 7 different settings are selected in order to interpret the performance of the Augmented ϵ-Constraint Method (AUGMECON) and Traditional ϵ-Constraint Method. Assembly line re-balancing with tool assignment problem is solved by these two methods, and mathematical models are coded using IBM ILOG CPLEX Optimization Studio. The code is available with examples and some supporting documentation in section 7.2.

This section includes the following sub-sections to explain the assembly line re-balancing problem with equipment assignment: In Section 5.1, experiment design is given and in Section 5.2 the results of the computational experiments are discussed. In Section 5.3, one problem instance is presented as an illustrative example to show all calculation procedures.

5.1 EXPERIMENT DESIGN

The tasks were assigned to the workstations with their equipments before the disruption. Therefore, main input of the problem is the initial assignment of the task with their equipment. Thus, this initial assignment is obtained from the simple assembly line balancing literature that are commonly used data sets. These data sets are taken from the study of Scholl and Klein [Scholl et al., 1995] on the website [Scholl, 2017]. The study of [Scholl et al., 1995] defines order strength as a complexity measure to examine the effect of precedence diagram intense. However, in this study, seven data sets are selected. They have different numbers of task in order to determine the effect of number of the tasks. Our main experiment is guided on seven of these data sets, from Mertens (1967) with $n = 7$ tasks, from Mansoor (1964) with $n = 11$ tasks, from Mitchell (1957) with $n = 21$ tasks, from Gunther et al. (1983) with $n = 35$ tasks, from Kilbridge and Wester (1962) with $n = 45$ tasks, from Hahn (1972) with $n = 53$ tasks and from Tonge (1961) with $n = 70$ tasks.

The data sets taken from above references include the processing time of each task and precedence relationship between the tasks. The number of equipments required for each task, type of equipments required for the task, the number of disrupted workstations and equipment costs are randomly generated. The precedence network diagrams for each data sets are given in section 7.1. The parameters information in the study is given in the following Table 1.

Table 1: The parameters for seven experiment data

	Parameters			
Data Sets	**# of tasks** (n)	**# of initial workstation** ($w\prime$)	**# of disrupted workstation**	**# of equipments** (l)
Mertens (1967)	7	6	2	5
Mansoor (1964)	11	4	1	5
Mitchell (1957)	21	8	2	10
Gunther et al. (1983)	35	14	2	15
Kilbridge and Wester (1962)	45	10	2	15
Hahn (1972)	53	8	2	15
Tonge (1961)	70	23	5	20

The disrupted workstations are randomly determined for each setting in the interval [2,5]. The number of equipments that is used on each experiment are distributed according to the following rule:

$$\text{The Number of Equipments} = \begin{cases} \text{Total number of tasks} \geq 60 & \text{20 equipments} \\ 30 \leq \text{Total number of tasks} < 60 & \text{15 equipments} \\ 20 \leq \text{Total number of tasks} < 30 & \text{10 equipments} \\ \text{Total number of tasks} < 20 & \text{5 equipments} \end{cases}$$

Equipment costs are randomly generated in the interval [20,120] and generated equipment cost parameters are given in Table 2 for each equipment.

Table 2: Generated equipment costs in the experiment

	Equipment Type																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Cost	50	90	120	60	45	50	20	70	100	120	90	50	50	90	120	60	100	80	90	70

After obtaining efficient points, the decision maker determines the optimal solution. Thus, decision maker's utility function was tested with different α_1 and α_2 values. The experiment is repeated under these seven different settings to three different α_1 and α_2 values for decision maker's utility function. As a result, $3 \times 7 = 21$ result combinations are obtained. α_1 and α_2 values for decision maker's utility function is defined in Table 3.

Table 3: α_1 and α_2 values used in decision maker's utility function

	α_1	α_2
Value 1	0.4	0.6
Value 2	0.7	0.3
Value 3	0.2	0.8

Traditional ϵ-Constraint Method and Augmented ϵ-Constraint Method (AUGMECON) algorithms are coded in C++ and the mathematical models are solved by IBM ILOG CPLEX 12.6. [IBM Corporation, 2012] The codes are provided in the section 7.2. The Interactive method is solved in the Microsoft Excel 2010. The algorithms and mathematical models are run on a computer with Intel(R) Core(TM) i7-4770S CPU @ 3.10 GHz, 16 GB RAM and Windows 7.

5.2 RESULTS OF THE EXPERIMENT

There are some performance measures in order to interpret the performance of The Augmented ϵ-Constraint Method against the Traditional ϵ-Constraint Method and benefit of the interactive method based on AUGMECON for decision maker. These performance measures are given below:

1. Number of models solved
2. Solution time
3. Number of efficient points
4. Number of iterations in interactive method

The number of solved models indicates how many times we solved the mathematical model in order to find efficient points. Solution time refers to the total time required for generating the set of all efficient and nonefficient solution points on each data sets. Two performance measures are number of models solved and solution times during the computational calculations assist to analyze the performance of The Augmented ϵ-Constraint Method against the Traditional ϵ-Constraint Method.

In this study, Interactive Method Based on AUGMECON is applied to select most preferred solution for decision maker. Also, number of efficient points and number of iterations are performance measures of this procedure. Number of efficient points indicates the efficient frontier that is obtained in AUGMECON. The number of iterations specifies iterations to reach the most preferred solution. These iterations are done under different α_1 ve α_2 values according to the decision maker's preferences. α_1 ve α_2 values change [0,1] interval.

Table 4 shows that the experiments with high number of tasks are harder to solve compared to the experiments with fewer number of tasks. Number of models solved and solution time increase in line with the number of tasks. The savings in the number of models and solution times obtained by AUGMECON increases as the problem size increases. For $n = 70$ and $w = 18$ as Experiment 7; the number of solved models decreases from 2293 to 35, the durations during the computational calculations decreases from 03:05:13:58 unit time to 00:45:53:45 unit time. For $n = 53$ and $w = 6$ as Experiment 6; the number of solved models decreases from 1987 to 29, the durations during the computational calculations decreases from 00:13:54:57 unit time to 00:00:18:38 unit time. Similar observations can be made for the other experiments. For $n = 11$ and $w = 3$ as Experiment 2; the number of solved models decreases from 34 to 7, but the duration times during the computational calculations increases from 00:00:03:98 unit time to 00:00:04:15 unit time. This is the only instance where solution time of the Augmented ϵ-Constraint Method exceeds the solution time of Traditional ϵ- Constraint Method. However, since the problem instance is too small,the solution times are negligible. Table 5 represents the saving of each experiments with application of the AUGMECON method.

Table 4: The experiment results of the two approaches: Traditional ϵ-Constraint Method and The Augmented ϵ-Constraint Method

	Traditional ϵ-Constraint Method		**The Augmented ϵ-Constraint Method (AUGMECON)**	
Experiment & Parameters	**# of Solved Models**	**Duration**	**# of Solved Models**	**Duration**
1	5	00:00:00:38	3	00:00:00:27
2	34	00:00:03:98	7	00:00:04:15
3	12	00:00:02:33	5	00:00:01:18
4	62	00:00:26:47	19	00:00:13:26
5	121	00:01:07:49	19	00:00:43:90
6	1987	00:13:54:57	29	00:00:18:38
7	2293	03:05:13:58	35	00:45:53:45

There is efficient frontier solutions for each experiment after the implementation of two method. Now, decision maker has to select the most preferred solution from the efficient frontier and so Interactive Method Based on AUGMECON is applied. Interactive Method is tested for three different sets of α values given in Table 3 for this purpose and results obtained are compared to the generated decision maker's utility function value. Interactive method results for each experiment are summarized in Table 6 at $\alpha_1 = 0.4$, $\alpha_2 = 0.6$, in Table 7 at $\alpha_1 = 0.7$, $\alpha_2 = 0.3$ and in Table 8 at $\alpha_1 = 0.2$, $\alpha_2 = 0.8$.

According to the Interactive Method results, if the number of efficient points is low, the number of iterations is low. Most preferred optimal solutions are changed with the decision

Table 5: Savings on each experiment with AUGMECON compared to the Traditional ϵ-Constraint Method

	Saving on # of Solved Model (%)	**Saving on Duration Time (%)**
Experiment 1	40	28.947
Experiment 2	79.412	-4.271
Experiment 3	58.333	49.356
Experiment 4	69.354	49.906
Experiment 5	84.298	34.953
Experiment 6	98.541	97.798
Experiment 7	98.474	75.195

Table 6: The results of the interactive method for each experiment at $\alpha_1 = 0.4$, $\alpha_2 = 0.6$

	# of Efficient Points	**# of iteration**	**Efficient point selected by the interactive method**		**Most preferred w.r.t. DM's underlying value function**		**Optimal**
			CT	**Cost**	**CT**	**Cost**	
Experiment 1	2	1	10	165	10	165	✓
Experiment 2	6	2	84	100	84	100	✓
Experiment 3	4	2	22	135	22	135	✓
Experiment 4	16	6	51	885	51	885	✓
Experiment 5	14	2	87	470	87	470	✓
Experiment 6	24	6	2600	580	2427	610	
Experiment 7	35	5	288	835	288	835	✓

maker's preference. The proposed interactive method is a heuristic approach. In order to measure the performance of the interactive method based on AUGMECON, the most preferred efficient solution with respect to the underlying value function of the decision maker is found. In Tables 6, 7 and 8, the solutions found by the interactive algorithm and the optimal solution found by using the underlying value function of the decision maker are given.

Table 7: The results of the interactive method for each experiment at $\alpha_1 = 0.7$, $\alpha_2 = 0.3$

	# of Efficient Points	**# of iteration**	**Efficient point selected by the interactive method**		**Most preferred w.r.t. DM's underlying value function**		**Optimal**
			CT	**Cost**	**CT**	**Cost**	
Experiment 1	2	1	9	225	9	225	✓
Experiment 2	6	2	62	320	64	230	
Experiment 3	4	2	18	235	18	235	✓
Experiment 4	16	6	51	885	51	885	✓
Experiment 5	14	2	87	470	87	470	✓
Experiment 6	24	4	2600	580	2427	610	
Experiment 7	35	5	288	835	288	835	✓

Table 8: The results of the interactive method for each experiment at $\alpha_1 = 0.2$, $\alpha_2 = 0.8$

	# of Efficient Points	**# of iteration**	**Efficient point selected by the interactive method**		**Most preferred w.r.t. DM's underlying value function**		**Optimal**
			CT	**Cost**	**CT**	**Cost**	
Experiment 1	2	1	10	165	10	165	✓
Experiment 2	6	2	94	50	94	50	✓
Experiment 3	4	2	28	115	28	115	✓
Experiment 4	16	3	102	635	68	745	
Experiment 5	14	4	87	470	87	470	✓
Experiment 6	24	2	4387	50	4387	50	✓
Experiment 7	35	4	433	755	433	755	✓

6 out of 7 experiments (85.7 %) result with the optimal solution using the interactive algorithm in Table 6 and Table 8. The interactive algorithm finds the optimal solution in 5 out of 7 experiments (71.4 %) in Table 7. Therefore, it can be concluded the interactive algorithm can find the most preferred solution of the decision maker in majority of problem instances in all weight sets.

5.3 ILLUSTRATIVE EXAMPLE

In this section, Augmented ϵ-Constrained Method is defined on Gunther et al. (1983) as an illustrative example to clarify solution approaches which is used in this research. The data set is taken from [Scholl, 2017] that is used in assembly line balancing literature. The example has 35 tasks and the precedence relationships between the tasks are given in Figure 3 with task times:

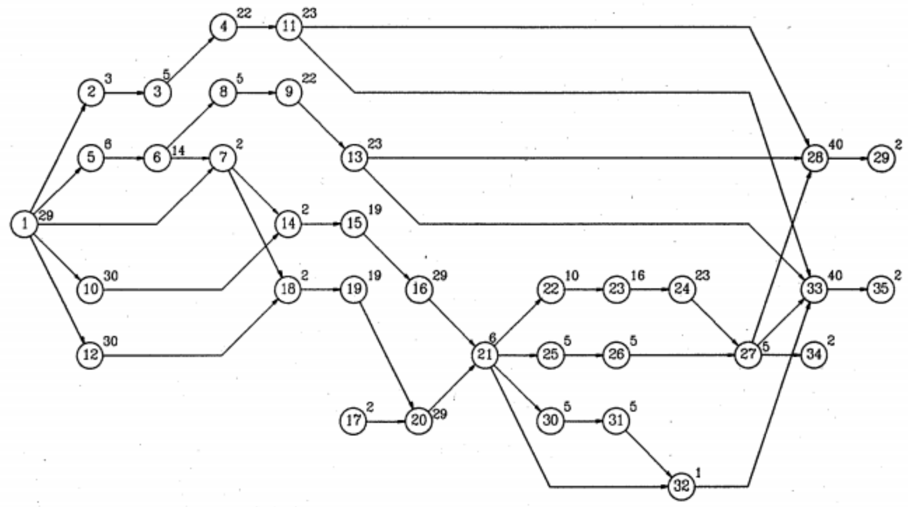

Figure 3: The precedence network of the sample problem
[Scholl et al., 1995]

In the sample problem, there are fourteen workstations in the initial configuration ($w\prime$) and optimal cycle time is 41 time units. The tasks on the assembly lines are assigned to these workstations with their equipment in order to minimize cycle time. The initial configuration of the sample is given in Figure 4 and equipment of each task is given in Table 9:

In Gunther et al.(1983) example, 2 workstations are disrupted, and so 35 tasks are reassigned to the non-disrupted workstation with their tools. The assembly line should be re-balanced since workstation 3 and workstation 8 are assumed to be disrupted workstations. Because of disruption, tasks 3, 6, 7, 8, 16, 21 and 25 have to be reassigned from the disrupted workstations

Figure 4: The initial configuration of the assembly line in the Gunther et al. (1983)

Table 9: The task and equipment assignment of the initial configuration of the sample

Workstation Number	Task	Equipment
W1	1,2,5,17	2,3,7,9,12,13
W2	3,6,7,8	2,3,4,5,7,13,14
W3	10,14	6,13,15
W4	12,18	1,2,4,12,13,15
W5	9,15	5,10
W6	4,19	6,10,13,15
W7	20	10
W8	16,21,25	1,2,5,8,9,12
W9	13,22,26	1,4,8,11,13
W10	23,24	7,10,11,12
W11	11,27,30,31,32,34	4,6,7,8,9,10,11,13,14
W12	28	4,12
W13	33	7
W14	29,35	3,12,15

to different workstations. During this reassignment process, new equipments are purchased, if required.

There are two objective functions that is presented in the mathematical model 3.1. Objective function 1 indicates the cycle time. It is converted to equality and it is assumed as a new variable. Cycle time takes a value that changes by ϵ value in each iteration. Before disruption, the cycle time of the sample is 41 time units. Therefore, trials are started with this cycle time. When the cycle time is 41 time units, there is no solution in the new configuration.

In the example, ϵ is assumed to be 1. Cycle time is increased as ϵ value and trials are made with given two approaches. According to the trials, minimum cycle time is 42 with cost of re-balancing 1545 and maximum cycle time is 102 with cost of re-balancing 635.

61 solution points are obtained in the Traditional ϵ-Constraint Method and these points are displayed in Figure 5. In the AUGMECON method, only efficient points are calculated so that number of iterations is decreased. 16 efficient points are obtained during the experiment. The graphical representation of the efficient points of the AUGMECON method is given in Figure 6.

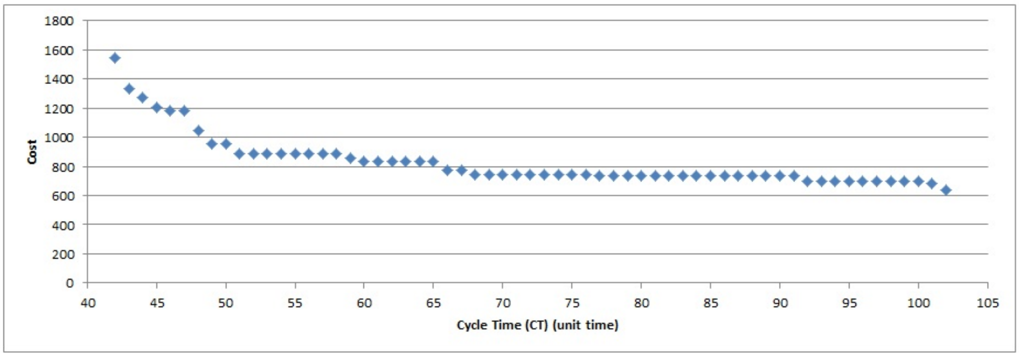

Figure 5: The calculated points in the Traditional ϵ-constraint Method

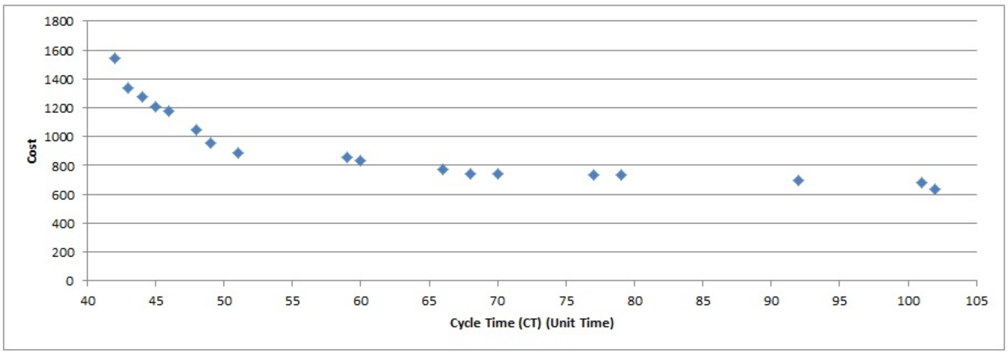

Figure 6: The efficient points in the AUGMECON Method

Although the number of models solved in the Traditional ϵ-constraint Method is 61, the number of models solved in the AUGMECON method is reduced to 16. The gain is approximately

70% with AUGMECON method. Also, the duration of during the computational calculations is decreased and saving on time is 50% with this method. There are 16 efficient points on Gunther (1983) sample (Experiment 4) and these points are given in Table 10.

Table 10: The efficient points of the sample

Point #	**Cycle Time**	**Cost**
1	42	1545
2	43	1335
3	44	1275
4	45	1205
5	46	1180
6	48	1045
7	49	955
8	51	885
9	59	855
10	60	835
11	66	775
12	68	745
13	77	735
14	92	695
15	101	685
16	102	635

After obtaining efficient solutions, decision maker selects the most preferred solution with the Interactive Method. The obtained Pareto optimal solutions are filtered using the Forward and Reverse Filtering Process. α_1 and α_2 values that is given Table 3 are used in the sample. Iterations numbers and most preferred solutions are reported at Table 6, Table 7 and Table 8. The calculation steps of the sample are represented to below at $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$.

For first iteration:

- R_i and π_i are computed for two objective vectors.

$$R_1 = CT_{16} - CT_1 = 102 - 42 = 60 \tag{32}$$

$$R_2 = Cost_1 - Cost_{16} = 1545 - 635 = 910 \tag{33}$$

$$\pi_1 = \frac{1}{R_1}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.93814433 \tag{34}$$

$$\pi_2 = \frac{1}{R_2}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.061856 \tag{35}$$

- L_p distances between the points are calculated and calculation results are tabulated in Table 11.

- d is calculated to select points.

$$d = \frac{D}{5-1} = \frac{79.6042}{5-1} = 19.901 \tag{36}$$

Point 1 and Point 16 are selected as bound. From left to right, first value that exceed 19.901 is selected. From down to up, first value that exceed 19.901 is selected. All points are selected using these procedures. Point 1, Point 4, Point 8, Point 13 and Point 16 are selected based on d value.

- Decision maker's utility function is computed with $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$.

$$f(1) = \left[0.4\frac{42-42}{102-42} - 0.6\frac{1545-635}{1545-635}\right] = 0.6 \tag{37}$$

$$f(4) = \left[0.4\frac{45-42}{102-42} - 0.6\frac{1205-635}{1545-635}\right] = 0.39582 \tag{38}$$

$$f(8) = \left[0.4\frac{51-42}{102-42} - 0.6\frac{885-635}{1545-635}\right] = 0.22484 \tag{39}$$

$$f(13) = \left[0.4\frac{77-42}{102-42} - 0.6\frac{735-635}{1545-635}\right] = 0.29927 \tag{40}$$

Table 11: L_p distances between the points at iteration 1

D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-	13.023	16.806	21.218	22.887	31.435	37.081	41.688	45.562	47.052	52.682	55.169	59.903	70.460	76.769	79.604
2		-	3.828	8.257	9.992	18.541	24.169	28.829	33.269	34.797	40.810	43.381	48.936	60.665	67.655	70.274
3			-	4.430	6.168	14.713	20.342	25.001	29.545	31.081	37.182	39.770	45.542	57.575	64.708	67.289
4				-	1.808	10.289	15.912	20.578	25.322	26.866	33.099	35.709	41.790	54.215	61.600	64.051
5					-	8.557	14.199	18.840	23.513	25.058	31.299	33.911	40.043	52.557	59.998	62.421
6						-	5.645	10.289	15.640	17.189	23.750	26.389	33.284	46.611	54.480	56.653
7							-	4.718	11.237	12.711	19.450	22.055	29.583	43.427	51.563	53.516
8								-	7.731	8.991	15.630	18.147	26.096	40.219	48.511	50.282
9									-	1.552	8.222	10.843	18.446	32.502	40.781	42.573
10										-	6.742	9.344	17.106	31.244	39.567	41.298
11											-	2.6389	10.612	24.888	33.303	34.865
12												-	8.465	22.726	31.180	32.614
13													-	14.288	22.726	24.255
14														-	8.465	10.088
15															-	3.231
16																-

$$f(16) = \left[0.4\frac{102-42}{102-42} - 0.6\frac{635-635}{1545-635}\right] = 0.4 \tag{41}$$

Decision maker selects Point 8 since minimum utility value is 0.22484.

- Lower bound is calculated for cycle time and cost based on Point 8.

$$LB_1^2 = 51 - 0.5 \times 51 - 42 = 46.5 \tag{42}$$

$$LB_2^2 = 885 - 0.5 \times 885 - 635 = 760 \tag{43}$$

Points are selected between point 6 and point 11 based on lower bounds.

For second iteration:

- R_i and π_i are computed for two objective vectors.

$$R_1 = CT_{11} - CT_6 = 66 - 48 = 18 \tag{44}$$

$$R_2 = Cost_6 - Cost_{11} = 1045 - 775 = 270 \tag{45}$$

$$\pi_1 = \frac{1}{R_1}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.9375 \tag{46}$$

$$\pi_2 = \frac{1}{R_2}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.0625 \tag{47}$$

- L_p distances between the points are calculated and calculation results are tabulated in Table 12.

- d is calculated to select points.

$$d = \frac{D}{3-1} = \frac{23.8649}{3-1} = 11.9324 \tag{48}$$

Table 12: L_p distances between the points at iteration 2

D	**6**	**7**	**8**	**9**	**10**	**11**
6	-	5.7025	10.3880	15.7278	17.2866	23.8649
7		-	4.7598	11.2673	12.7514	19.5081
8			-	7.7308	8.9976	15.6531
9				-	1.5625	8.2502
10					-	6.7604
11						-

Since there are 6 points remaining, we have to select 3 points. Point 6, Point 9 and Point 11 are selected based on d value.

- Decision maker's utility function is computed with $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$.

$$f(6) = \left[0.4\frac{48-42}{102-42} - 0.6\frac{1045-635}{1545-635}\right] = 0.31033 \tag{49}$$

$$f(9) = \left[0.4\frac{59-42}{102-42} - 0.6\frac{855-635}{1545-635}\right] = 0.25839 \tag{50}$$

$$f(11) = \left[0.4\frac{66-42}{102-42} - 0.6\frac{775-635}{1545-635}\right] = 0.25231 \tag{51}$$

Decision maker selects point 9 since utility value is 0.25231.

- Lower bound is calculated for cycle time and cost based on Point 11.

$$LB_1^3 = 66 - 0.5 \times 66 - 42 = 54 \tag{52}$$

$$LB_2^3 = 775 - 0.5 \times 775 - 635 = 705 \tag{53}$$

Points are selected between point 9 and point 13 based on lower bounds.

For third iteration:

- R_i and π_i are computed for two objective vectors.

$$R_1 = CT_{13} - CT_9 = 77 - 59 = 18 \tag{54}$$

$$R_2 = Cost_9 - Cost_{13} = 855 - 735 = 120 \tag{55}$$

$$\pi_1 = \frac{1}{R_1}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.86957 \tag{56}$$

$$\pi_2 = \frac{1}{R_2}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.13043 \tag{57}$$

- L_p distances between the points are calculated and calculation results are tabulated in Table 13.

Table 13: L_p distances between the points at iteration 3

D	**9**	**10**	**11**	**12**	**13**
9	-	2.74981	12.0804	16.3434	22.1355
10		-	9.4057	13.6455	19.7144
11			-	4.2821	10.8956
12				-	7.9340
13					-

- d is calculated to select points.

$$d = \frac{D}{3-1} = \frac{22.135}{3-1} = 11.0678 \tag{58}$$

Point 9, Point 10 and Point 13 are selected based on d value.

- Decision maker's utility function is computed with $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$.

$$f(9) = \left[0.4\frac{59-42}{102-42} - 0.6\frac{855-635}{1545-635}\right] = 0.25839 \tag{59}$$

$$f(10) = \left[0.4\frac{60-42}{102-42} - 0.6\frac{835-635}{1545-635}\right] = 0.25187 \tag{60}$$

$$f(13) = \left[0.4\frac{77-42}{102-42} - 0.6\frac{735-635}{1545-635}\right] = 0.29927 \tag{61}$$

Decision maker selects point 9 since utility value is 0.25231.

- Lower bound is calculated for cycle time and cost based on Point 10.

$$LB_1^4 = 60 - 0.5 \times 60 - 42 = 51 \tag{62}$$

$$LB_2^4 = 835 - 0.5 \times 835 - 635 = 735 \tag{63}$$

Points are selected between point 8 and point 13 based on lower bounds.

For forth iteration:

- R_i and π_i are computed for two objective vectors.

$$R_1 = CT_{13} - CT_8 = 77 - 51 = 26 \tag{64}$$

$$R_2 = Cost_8 - Cost_{13} = 885 - 735 = 150 \tag{65}$$

$$\pi_1 = \frac{1}{R_1}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.85227 \tag{66}$$

$$\pi_2 = \frac{1}{R_2}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.14773 \tag{67}$$

- L_p distances between the points are calculated and calculation results are tabulated in Table 14.

- d is calculated to select points.

$$d = \frac{D}{3-1} = \frac{31.3377}{3-1} = 15.6688 \tag{68}$$

Point 8, Point 11 and Point 13 are selected based on d value.

Table 14: L_p distances between the points at iteration 4

D	**8**	**9**	**10**	**11**	**12**	**13**
8	-	8.1319	10.6487	20.6760	25.2519	31.3377
9		-	3.0750	13.2386	17.9694	23.4435
10			-	10.2330	14.9418	20.6919
11				-	4.7483	11.0819
12					-	7.8114
13						-

- Decision maker's utility function is computed with $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$.

$$f(8) = \left[0.4\frac{51-42}{102-42} - 0.6\frac{885-635}{1545-635}\right] = 0.22484 \tag{69}$$

$$f(11) = \left[0.4\frac{66-42}{102-42} - 0.6\frac{775-635}{1545-635}\right] = 0.25231 \tag{70}$$

$$f(13) = \left[0.4\frac{77-42}{102-42} - 0.6\frac{735-635}{1545-635}\right] = 0.29927 \tag{71}$$

Decision maker selects point 8 since utility value is 0.22484.

- Lower bound is calculated for cycle time and cost based on Point 8.

$$LB_1^5 = 51 - 0.5 \times 51 - 42 = 46.5 \tag{72}$$

$$LB_2^5 = 885 - 0.5 \times 855 - 635 = 760 \tag{73}$$

Points are selected between point 6 and point 11 based on lower bounds.

For fifth iteration:

- R_i and π_i are computed for two objective vectors.

$$R_1 = CT_{11} - CT_6 = 66 - 48 = 18 \tag{74}$$

$$R_2 = Cost_6 - Cost_{11} = 1045 - 775 = 270 \tag{75}$$

$$\pi_1 = \frac{1}{R_1}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.9375 \tag{76}$$

$$\pi_2 = \frac{1}{R_2}\left[\frac{1}{R_1} + \frac{1}{R_2}\right]^{-1} = 0.0625 \tag{77}$$

- L_p distances between the points are calculated and calculation results are tabulated in Table 15.

Table 15: L_p distances between the points at iteration 5

D	**6**	**7**	**8**	**9**	**10**	**11**
6	-	5.7025	10.3880	15.7278	17.2866	23.8649
7		-	4.7598	11.2673	12.7514	19.5081
8			-	7.7308	8.9976	15.6531
9				-	1.5625	8.2502
10					-	6.7604
11						-

- d is calculated to select points.

$$d = \frac{D}{3-1} = \frac{23.8649}{3-1} = 11.9324 \tag{78}$$

Point 6, Point 8 and Point 11 are selected based on d value.

- Decision maker's utility function is computed with $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$.

$$f(8) = \left[0.4\frac{48-42}{102-42} - 0.6\frac{1045-635}{1545-635}\right] = 0.31033 \tag{79}$$

$$f(11) = \left[0.4\frac{51-42}{102-42} - 0.6\frac{885-635}{1545-635}\right] = 0.22484 \tag{80}$$

$$f(13) = \left[0.4\frac{66-42}{102-42} - 0.6\frac{775-635}{1545-635}\right] = 0.25321 \tag{81}$$

Decision maker selects point 8 since utility value is 0.22484.

- Lower bound is calculated for cycle time and cost based on Point 8.

$$LB_1^5 = 51 - 0.5 \times 51 - 42 = 46.5 \tag{82}$$

$$LB_2^5 = 885 - 0.5 \times 855 - 635 = 760 \tag{83}$$

As a result of these iterations, we conclude that the most preferred solution has 51 unit cycle time and 855 unit cost at $\alpha_1 = 0.4$, $\alpha_2 = 0.6$. The results of the interactive method solution is represented in Table 16.

Table 16: The results of the interactive method on Gunther sample

	Interactive Method		**Utility Function**	
	CT	**Cost**	**CT**	**Cost**
$\alpha_1 = 0.4$ $\alpha_2 = 0.6$	51	885	51	885
$\alpha_1 = 0.7$ $\alpha_2 = 0.3$	51	885	51	885
$\alpha_1 = 0.2$ $\alpha_2 = 0.8$	102	635	68	745

Most preferred solution depends on the decision maker's preferences. When $\alpha_1 = 0.4$, $\alpha_2 = 0.6$ and $\alpha_1 = 0.7$, $\alpha_2 = 0.3$; optimal cycle time is 51 time units and optimal cost is 885 cost units. However, for $\alpha_1 = 0.2$ and $\alpha_2 = 0.8$; the cost preference is more increased to cycle time preference so that the most preferred solution is changed. It has 102 time unit and 635 cost unit.

6 CONCLUSION

In this thesis, assembly line re-balancing problem with tool assignment is considered. The disruption in some of the workstations cause an infeasible assembly line. As a result of this disruption, the tasks in the disrupted workstations should be reassigned to the non-disrupted workstations with required tools regarding the precedence relationship. There are stability and efficieny measures to reach re-balanced line. The cycle time is efficiency measure to obtain maximum production rate. Also, the number of non-disrupted workstation is stability measure in order to assign tasks.

Two algorithms are presented to obtain Pareto frontier of the problem. First algorithm is Traditional ϵ-Constrained Method and the second is Augmented ϵ-Constrained Method. First method finds all points in the feasible region using computational experiments. The second method calculates only efficient points in the feasible region. These two methods are compared to each other according to number of solved models and solution times as the performance measures. The savings on the computational calculations is changed between 40 percent and 98.5 percent through the AUGMECON method.

The Interactive method provides to select most preferred solution from the efficient points. The experiments is applied in three different decision maker preferences and the results are tested. The interactive method can find optimal solution with respect to the majority with high percentage.

There are many gaps in the assembly line balancing literature so that our study makes a great contribution to the literature. In future research, new efficiency, stability and performance measures can be identified to rebalance. The number of disrupted workstations can be changed to observe the effect of the measures.

Another future research can be testing different layouts of assembly lines such as U-shaped assembly line, parallel assembly line and flexible assembly line. Product can be selected as a mixed model on the re-balancing line. The new proposed algorithm can be generated for solving assembly line re-balancing problem. The genetic algorithm and tabu search method may suggested to apply for these problems. Further changes can be changing the input parameters, change in demand pattern, change in task times, technological restrictions or product types.

7 APPENDIX

7.1 PRECEDENCE DIAGRAMS

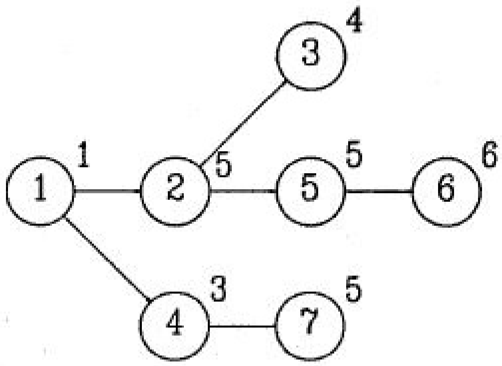

Figure 7: The precedence diagram of Mertens sample

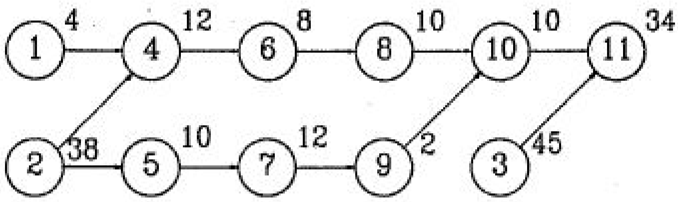

Figure 8: The precedence diagram of Mansoor sample

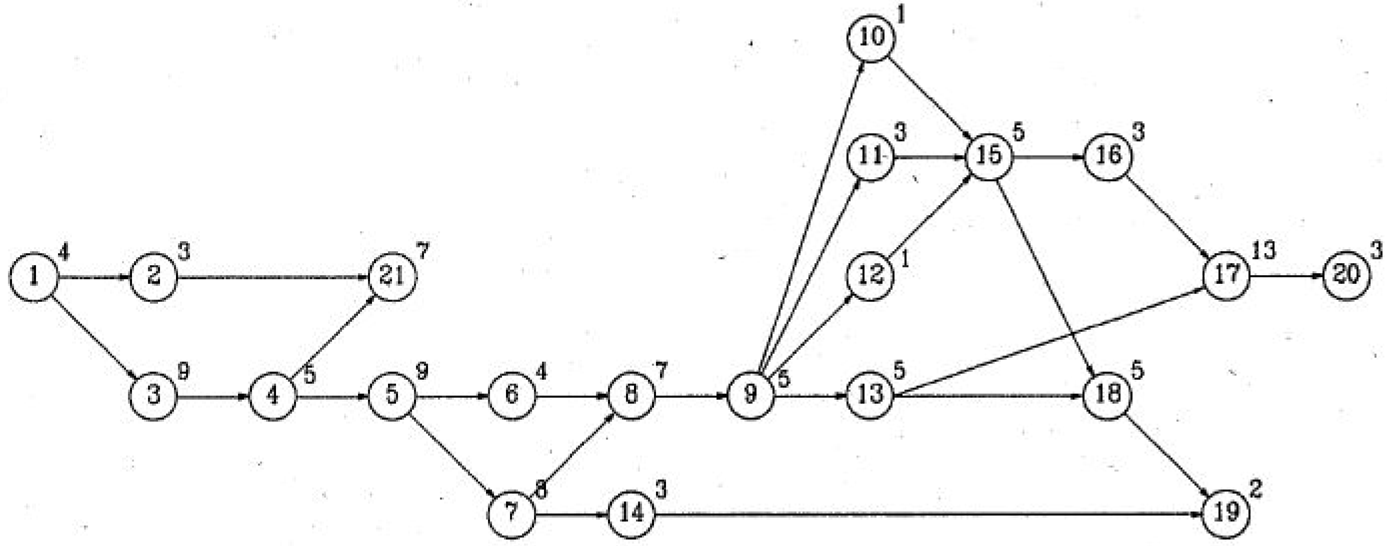

Figure 9: The precedence diagram of Mitchell sample

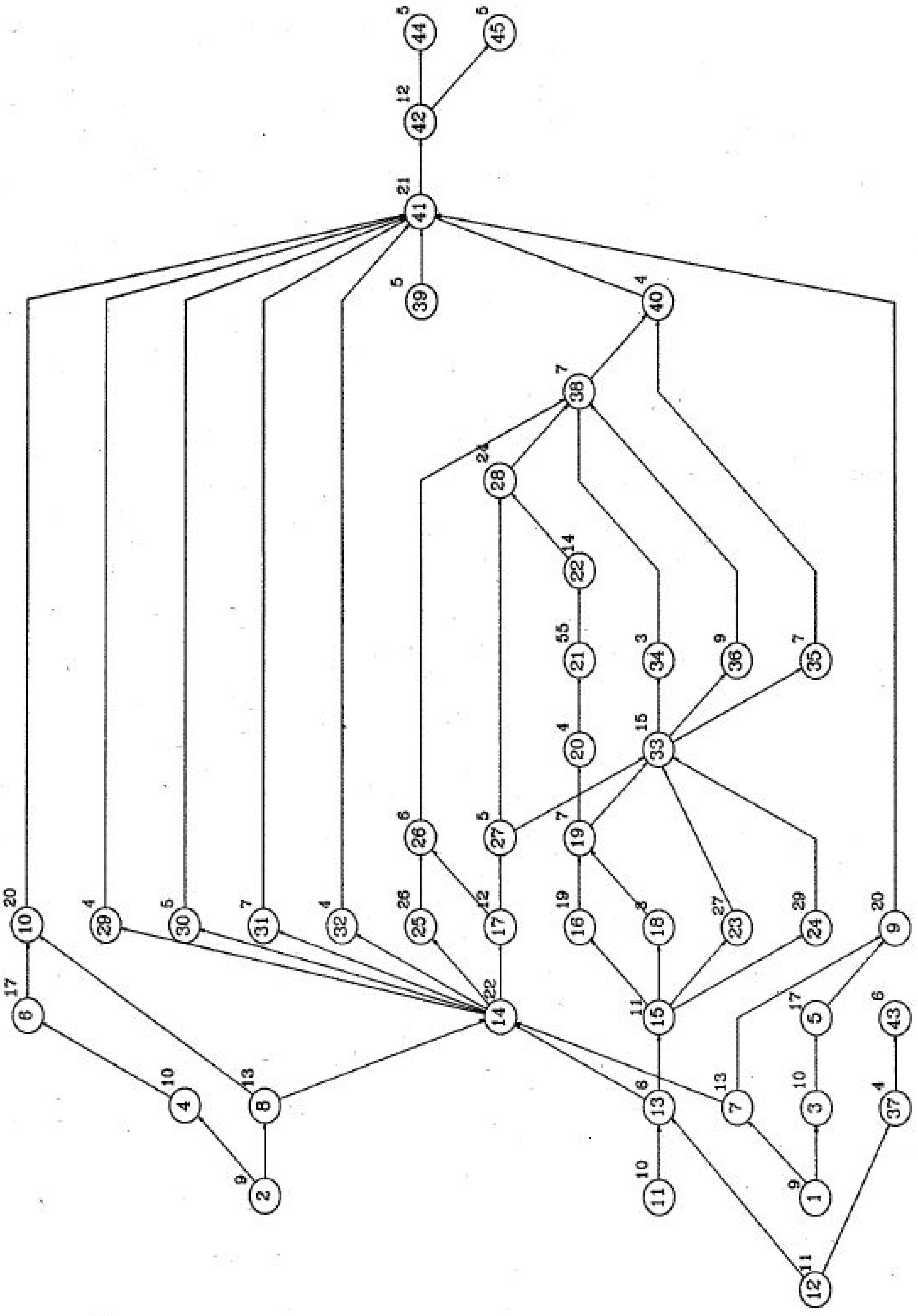

Figure 10: The precedence diagram of Kilbridge-Wester sample

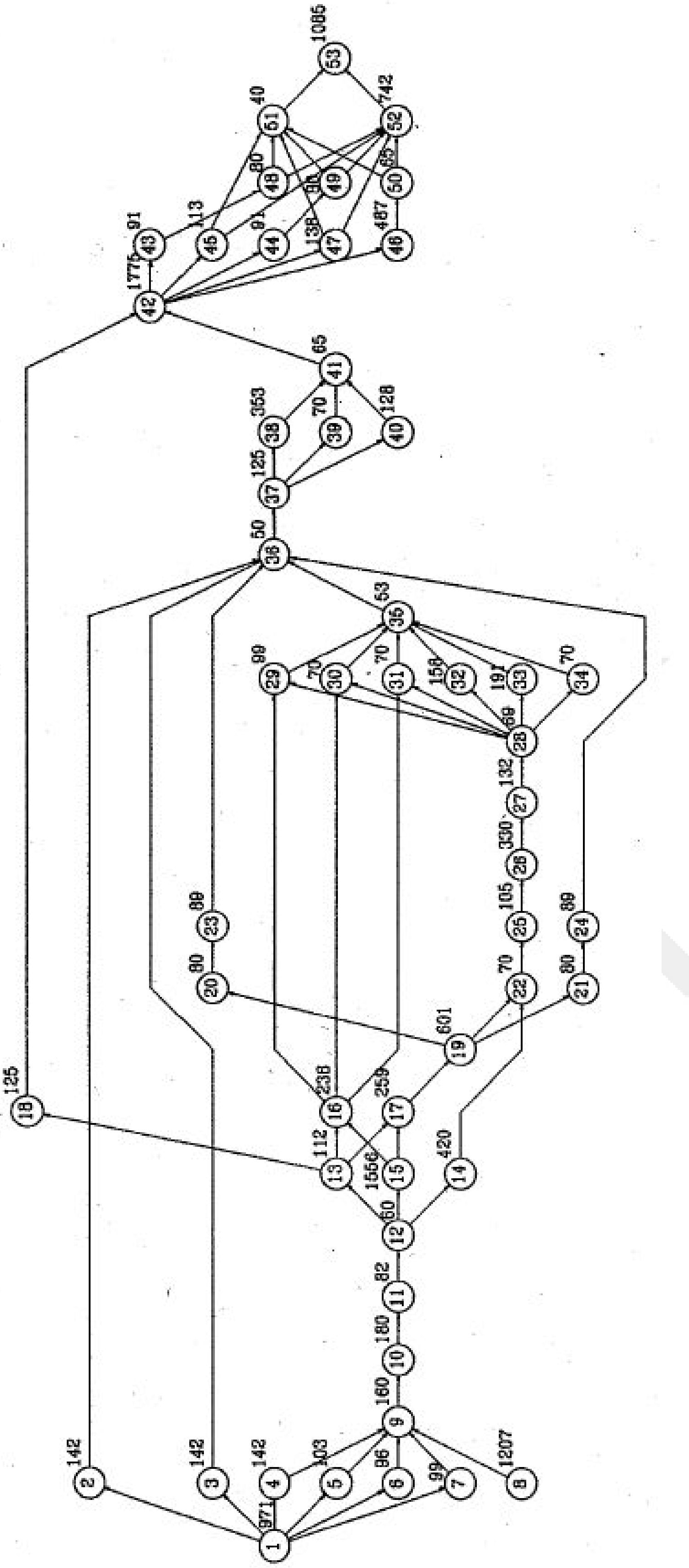

Figure 11: The precedence diagram of Hahn sample

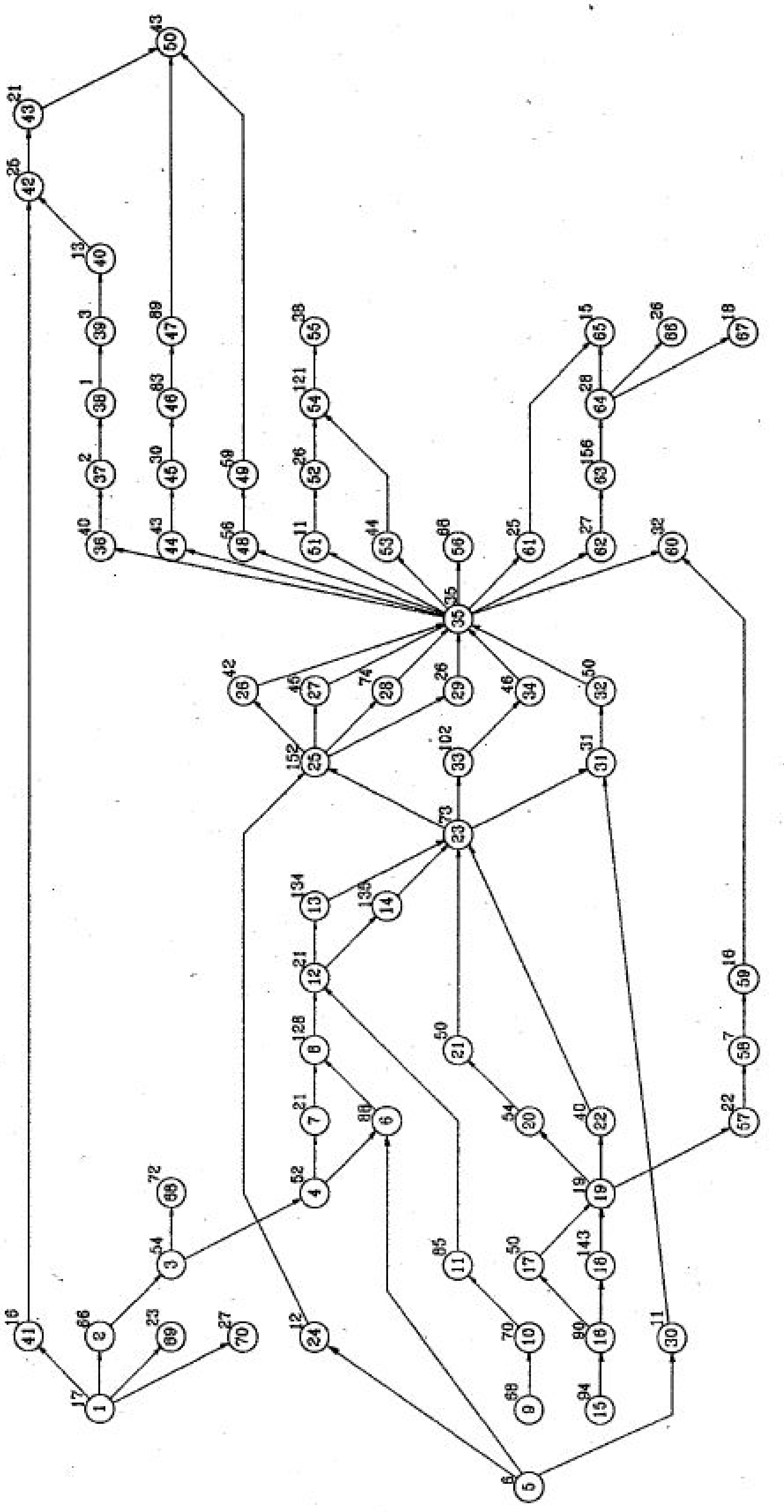

Figure 12: The precedence diagram of Tonge sample

7.2 CODES

7.2.1 TRADITIONAL ϵ - CONSTRAINT MODEL

```
range task=1..35;
range equipment=1..15;
range workstation=1..12;
int processtime[task]=...;
int cost[equipment]=...;
int l[task,equipment]=...;
int precedence[task,task]=...;
int u[equipment,workstation]=...;
dvar boolean x[task,workstation];
dvar boolean y[equipment,workstation];
minimize sum(k in workstation)sum(j in equipment)cost[j]*y[j,k];
subject to
{
forall(i in task)
{
sum(k in workstation) x[i,k]==1;
}
forall(k in workstation)
{
sum(i in task)processtime[i]* x[i,k]¡=41;
}
forall(i in task,j in equipment,k in workstation : l[i][j]==1)
{
x[i,k]¡=u[j,k]+y[j,k];
}
forall(i in task, h in task : precedence[i][h]==1)
{
sum(k in workstation)k*x[i,k]¡=sum(k in workstation)k*x[h,k];
}
}
```

7.2.2 AUGMENTED ϵ - CONSTRAINT MODEL

```
range task=1..35;
range equipment=1..15;
range workstation=1..12;
int processtime[task]=...;
int cost[equipment]=...;
int l[task,equipment]=...;
int precedence[task,task]=...;
int u[equipment,workstation]=...;
dvar boolean x[task,workstation];
dvar boolean y[equipment,workstation];
dvar float+ s;
dvar int CT;
minimize sum(k in workstation)sum(j in equipment)cost[j]*y[j,k]-0.0001*s;
subject to
{
forall(i in task)
{
sum(k in workstation) x[i,k]==1;
}
forall(k in workstation)
{
sum(i in task)processtime[i]* x[i,k]¡=CT;
}
forall (k in workstation)
{
CT+s==83;
}
forall(i in task,j in equipment,k in workstation : l[i][j]==1)
{
x[i,k]¡=u[j,k]+y[j,k];
}
forall(i in task, h in task : precedence[i][h]==1)
```

{

*sum(k in workstation)k*x[i,k]¡=sum(k in workstation)k*x[h,k];*

}

}

References

[Bérubé et al., 2009] Bérubé, J.-F., Gendreau, M., and Potvin, J.-Y. (2009). An exact ε-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits. *European Journal of Operational Research*, 194(1):39–50.

[Celik et al., 2014] Celik, E., Kara, Y., and Atasagun, Y. (2014). A new approach for rebalancing of u-lines with stochastic task times using ant colony optimisation algorithm. *International Journal of Production Research*, 52(24):7262–7275.

[Chutima and Chimklai, 2012] Chutima, P. and Chimklai, P. (2012). Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge. *Computers & Industrial Engineering*, 62(1):39–55.

[Gamberini et al., 2009] Gamberini, R., Gebennini, E., Grassi, A., and Regattieri, A. (2009). A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem. *International Journal of Production Research*, 47(8):2141–2164.

[Gamberini et al., 2006] Gamberini, R., Grassi, A., and Rimini, B. (2006). A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem. *International Journal of Production Economics*, 102(2):226–243.

[Grangeon et al., 2011] Grangeon, N., Leclaire, P., and Norre, S. (2011). Heuristics for the re-balancing of a vehicle assembly line. *International Journal of Production Research*, 49(22):6609–6628.

[Grzechca, 2014] Grzechca, W. (2014). Assembly line balancing problem with reduced number of workstations. *IFAC Proceedings Volumes*, 47(3):6180–6185.

[Hwang and Yoon, 1981] Hwang, C.-L. and Yoon, K. (1981). Multiple attribute decision making, vol. 186 of. *Lecture Notes in Economics and Mathematical Systems.*

[IBM Corporation, 2012] IBM Corporation (2012). IBM ILOG CPLEX Studio IDE.

[Kottas and Lau, 1973] Kottas, J. F. and Lau, H.-S. (1973). A cost-oriented approach to stochastic line balancing1. *AIIE Transactions*, 5(2):164–171.

[Lyu, 1997] Lyu, J. (1997). A single-run optimization algorithm for stochastic assembly line balancing problems. *Journal of Manufacturing Systems*, 16(3):204–210.

[Mavrotas, 2009] Mavrotas, G. (2009). Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2):455–465.

[Pape, 2015] Pape, T. (2015). Heuristics and lower bounds for the simple assembly line balancing problem type 1: Overview, computational tests and improvements. *European Journal of Operational Research*, 240(1):32–42.

[Rada-Vilela et al., 2013] Rada-Vilela, J., Chica, M., Cordón, Ó., and Damas, S. (2013). A comparative study of multi-objective ant colony optimization algorithms for the time and space assembly line balancing problem. *Applied Soft Computing*, 13(11):4370–4382.

[Raj et al., 2016] Raj, A. V., Mathew, J., Jose, P., and Sivan, G. (2016). Optimization of cycle time in an assembly line balancing problem. *Procedia Technology*, 25:1146–1153.

[Saif et al., 2014] Saif, U., Guan, Z., Liu, W., Zhang, C., and Wang, B. (2014). Pareto based artificial bee colony algorithm for multi objective single model assembly line balancing with uncertain task times. *Computers & Industrial Engineering*, 76:1–15.

[Sarin et al., 1999] Sarin, S. C., Erel, E., and Dar-El, E. M. (1999). A methodology for solving single-model, stochastic assembly line balancing problem. *Omega*, 27(5):525–535.

[Scholl, 2017] Scholl, A. (2017). Assembly line balancing data sets and research topics. `https://assembly-line-balancing.de`.

[Scholl et al., 1995] Scholl, A. et al. (1995). *Data of assembly line balancing problems*. Techn. Hochsch., Inst. für Betriebswirtschaftslehre.

[Simaria and Vilarinho, 2004] Simaria, A. S. and Vilarinho, P. M. (2004). A genetic algorithm based approach to the mixed-model assembly line balancing problem of type ii. *Computers & Industrial Engineering*, 47(4):391–407.

[Steuer, 1986] Steuer, R. E. (1986). *Multiple criteria optimization: theory, computation, and applications*. Wiley.

[Wang et al., 2011] Wang, H., Zhu, X., Wang, H., Hu, S. J., Lin, Z., and Chen, G. (2011). Multi-objective optimization of product variety and manufacturing complexity in mixed-model assembly systems. *Journal of Manufacturing Systems*, 30(1):16–27.

[Yang et al., 2013] Yang, C., Gao, J., and Sun, L. (2013). A multi-objective genetic algorithm for mixed-model assembly line rebalancing. *Computers & Industrial Engineering*, 65(1):109–116.

[Yoosefelahi et al., 2012] Yoosefelahi, A., Aminnayeri, M., Mosadegh, H., and Ardakani, H. D. (2012). Type ii robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model. *Journal of Manufacturing Systems*, 31(2):139–151.

[Yuguang et al., 2016] Yuguang, Z., Bo, A., and Yong, Z. (2016). A pso algorithm for multi-objective hull assembly line balancing using the stratified optimization strategy. *Computers & Industrial Engineering*, 98:53–62.

[Zha and Yu, 2014] Zha, J. and Yu, J.-j. (2014). A hybrid ant colony algorithm for u-line balancing and rebalancing in just-in-time production environment. *Journal of Manufacturing Systems*, 33(1):93–102.