HEURISTIC APPROACHES FOR ASSEMBLY LINE BALANCING AND COMPETENT WORKER ASSIGNMENT PROBLEM (ALWABP)

A Thesis

by

Muhammet Fatih Çengil

Submitted to the

Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
January 2018

HEURISTIC APPROACHES FOR ASSEMBLY LINE BALANCING AND COMPETENT WORKER ASSIGNMENT PROBLEM (ALWABP)

Approved by:

Assistant Professor, Görkem Yılmaz,
Advisor,
Department of Industrial Engineering
Özyeğin University

Associate Professor, Mehmet Güray Güler,
Department of Industrial Engineering
Yıldız Technical University

Assistant Professor, Erinç Albey,
Co-Advisor,
Department of Industrial Engineering
Özyeğin University

Associate Professor, Erhun Kundakçıoğlu,
Department of Industrial Engineering
Özyeğin University

Assistant Professor, İhsan Yanıkoğlu,
Department of Industrial Engineering
Özyeğin University

Date Approved: 27.12.2017

To My Family

ABSTRACT

In this study, we focus on assembly line balancing and worker assignment problem, where multiple production lines are executed by a shared pool of workers with varying skills and having absenteeism issues. We developed two mathematical models for the real life situation at Vestel Electronics, which is the second largest TV manufacturer in Europe. Nonlinearity in one of the developed mathematical models is resolved through McCormick relaxations and both models provide optimal results for small instances in short time. However, it is seen that the models are intractable for the real life cases. In Vestel Electronics, worker-line and task-worker assignments should be determined by considering skill of workers. Due to the absenteeism problem, line balancing should be resolved at the beginning of each shift, which brings up a rigid limit on the solution generation time. To overcome this complexity, three heuristic approaches are developed. In first heuristic, master problem is decomposed into two sub-problems and constructing a multi-pass hierarchical solution framework. In Heuristic 2, solution of first heuristic is aimed to be improved. Once second sub-problem in Heuristic 1 is solved, tasks are reassigned to stations with the given worker assignment and algorithms are run iteratively until objective value no more improves. In Heuristic 3, workers are first assigned to lines in a balanced way in terms of task abilities of the workers with an algorithm. Then, master problem is solved with this input. Experimental results reveal that our decomposition approaches provide good enough solutions within the allowed time; encouraging Vestel Electronics' production team to implement the proposed approach as an online decision support system.

Keywords: Multi Product Parallel Assembly Line Balancing Problem, Worker Assignment, Worker Skill and Absenteeism, Mixed Integer Nonlinear Programming, McCormick Relaxation, Decomposition Methods.

ÖZETÇE

Bu çalışmada işçilerin yetenekleri ve devamsızlık durumunu göz önünde bulunduran, işçilerin havuz kadrosundan çoklu üretim hatlarına atanmasını konu edinen hat dengeleme ve işçi ataması problemine odaklanılmıştır. Avrupa'nın en büyük ikinci TV üreticisi olan Vestel Elektronik A.Ş.'nin gerçek hayat durumunu yansıtan iki matematiksel model geliştirilmiştir. Modellerin bir tanesinde karşılaşılan doğrusal olmama durumu McCormick Gevşetmesi kullanılarak çözülmüştür ve iki model de küçük örnekler için kısa sürede en iyi çözümü vermektedir. Ancak, gerçek hayat örnekleri için iyi bir sonuç alınamamaktadır. Vestel'de işçi-hat ve iş-işçi atamaları işçilerin yetenekleri göz önünde bulundurularak yapılmaktadır. Devamsızlık problem nedeniyle hat dengeleme, vardiya başlangıcında yapılmalıdır ve bu durum çözüm süresine bir zaman limiti getirmektedir. Bu zorluğun üstesinden gelmek için, üç sezgisel algoritma geliştirilmiştir. İlk algoritmada ana problem iki ayrı alt problem ayrılmıştır ve çok geçişli hiyerarşik çözüm yapısı kullanılmıştır. İkinci sezgisel algoritmada, ilk sezgisel algoritmada ulaşılan çözümün geliştirilmesi hedeflenir. İlk sezgisel algoritmadaki ikinci alt problem çözüldükten sonra işler mevcut işçi ataması kullanılarak yeniden atanır ve işçilerin ve işlerin istasyonlara yeniden atanmasını sağlayan algoritmalar daha iyi çözüm bulunamayana dek ardışık olarak çalıştırılır. Üçüncü sezgisel algoritmada işçilerin işleri yapabilirliği göz önünde bulundurularak, algoritma ile hatlara atanması sağlanır. Sonrasında ana modeller bu girdi ile çalıştırılarak istenilen sürede çözüme ulaşılması beklenir. Uygulama sonuçları, alt problemlere ayırma yöntemi ile geliştirilen algoritmaların izin verilen süre içerisinde yeterince iyi sonuçlar verdiğini göstermektedir ve Vestel üretim ekibinin önerilen yaklaşımları karar destek sistemi olarak kullanma yönünde teşvik etmiştir.

Anahtar Kelimeler: Çok Ürünlü Paralel Montaj hatları Problemi, İşçi Ataması, İşçi Yeteneği ve Devamsızlığı, Karışık Tamsayılı Doğrusal Olmayan Programlama, McCormick Gevşetmesi, Alt Problemlere Ayırma Yöntemi

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisors Erinç Albey and Görkem Yılmaz for their invaluable guidance and encouragement in finishing this study.

I also sincerely thank my wife Betül Çengil for her endless patience and help. She always encouraged me during the study and I would have never completed this study without her help.

Finally, I would like to express my special gratitude to my parents who helped me in my whole life.

TABLE OF CONTENTS

ABSTRACT 4

ÖZETÇE 5

ACKNOWLEDGEMENT 6

LIST OF TABLES 9

LIST OF FIGURES 11

1. INTRODUCTION 12

2. PROBLEM DEFINITION 14

2.1. Fields of Activity for Vestel Electronics 14

2.2. Final Assembly Lines in TV Production 14

2.3. Assembly Line Balancing Activities in TV Production 15

2.4. Monthly TV Production Numbers 16

2.5. Product Types for TVs 17

2.6. Schedule for Production Plan 18

2.7. Worker Absenteeism Rates and Competencies 18

3. LITERATURE REVIEW 20

4. METHODOLOGY 26

4.1. EXACT ALGORITHMS 27

4.1.1. ALWABP - 1 27

4.1.2. ALWABP - 2 30

4.2. HEURISTIC 1 36

4.2.1. ALWABP - 1 Decomposition Part I 37

4.2.2. ALWABP - 1 Decomposition Part II 37

4.2.3. ALWABP - 2 Decomposition Part I 38

4.2.4. ALWABP - 2 Decomposition Part II 39

4.2.5. Infeasibility Check 39

4.2.5.1. Infeasibility Case I 40

4.2.5.2. Infeasibility Case II 43

4.3. HEURISTIC 2 44

4.4. HEURISTIC 3 45

5. RESULTS 49

5.1. INSTANCE GENERATION 49

5.2. COMPUTATIONAL STUDY 51

5.2.1. Output of Exact Algorithms 51

5.2.2. Output of Heuristic 1 54

5.2.3. Output of Heuristic 2 56

5.2.4. Output of Heuristic 3 57

5.2.5. Line Balance for Exact and Heuristic Algorithms 59

5.2.6. Comparison for Experience Level of the Workers Used by Proposed Algorithms 60

6. CONCLUSION 63

7. REFERENCES 65

8. VITA 68

LIST OF TABLES

Table 1: Literature Review for ALWABP 21

Table 2: Notation for ALWABP - 1 27

Table 3: Additional Notation for Vestel Specific Parameters 27

Table 4: Number of Decision Variables for (Araújo, Costa and Miralles, 2013) and Our Model 29

Table 5: Additional Notation for ALWABP-2 30

Table 6: Small Case Representing Cycle Time and Throughput of Lines 31

Table 7: Additional Notations for ALWABP – 2 Model 32

Table 8: Worker-Task Compatibility Parameters for (Araújo, Costa and Miralles, 2013) and Our Model 34

Table 9: Worker-Task Compatibility Parameters for Different Product Numbers 35

Table 10: Notation for ALWABP - 1 Decomposition Part I 37

Table 11: Notation for ALWABP - 1 Decomposition Part II 37

Table 12: Notation for ALWABP - 2 Decomposition Part I 38

Table 13: Notation for ALWABP - 2 Decomposition Part II 39

Table 14: Small Case for Worker-Task Compatibility 40

Table 15: Notation for Assignment Problem 40

Table 16: Notation for Heuristic 3 Decomposition Algorithm 45

Table 17: Worker-Task Compatibility for Small Case 46

Table 18: Optimal Worker-Line Assignment for Small Case 47

Table 19: Number of Workers Having Ability to Perform a Task in a Line 47

Table 20: Minimum and Maximum Number of Workers Having Ability to Perform a Task Among All Lines 47

Table 21: Instance Groups and Corresponding Values 49

Table 22: Objective Functions and Their Descriptions 50

Table 23: Exact Algorithm Solutions for 2 Minutes Time Limit 52

Table 24: Exact Algorithm Solutions for 60 Minutes Time Limit 53

Table 25: Time to Get First Feasible Solution 54

Table 26: Heuristic 1 Solutions for 2 Minutes Time Limit 55

Table 27: Heuristic 1 and Heuristic 2 solutions for 2 minutes time limit... 56

Table 28: Comparison between Solutions of Heuristic 2, Exact Algorithms and Heuristic 1 57

Table 29: Heuristic 3 Solutions with 2 Minutes Time Limit... 58

LIST OF FIGURES

Figure 1: Monthly Demand of TV Products in 2017 17

Figure 2: Classification of Assembly Line Balancing (Scholl, Becker and C., 2006) 20

Figure 3: Flow Chart for Heuristic 1 36

Figure 4: Flow Chart for Heuristic 2 44

Figure 5: Line Balance for Exact and Heuristic Algorithms 60

Figure 6: Percentage of High Experienced Workers Used in Exact and Heuristic Algorithms for Obj. 1 61

1. INTRODUCTION

First assembly line is started to be used by Henry Ford in the early 1900's. It is used mostly for the production of the high volume standardized products and it has a great importance for mass production. An assembly line consists of sequential workstations and work in process moves from a workstation to next workstation via transportation system such as conveyors. Workstation with the highest processing time constitutes the bottleneck and due to the sequential design, the bottleneck station sets the speed of the production, in other words the cycle time of the line. In order to manage the cycle time of the production line, one approach is distributing tasks to workstations in a balanced way such that total workload assigned to each workstation does not deviate largely from other stations. This problem is called Assembly Line Balancing Problem (ALBP) in (Becker and Scholl, 2006).

ALBP has been frequently studied in the literature. (Becker and Scholl, 2006) defines Simple Assembly Line Balancing (SALBP) as assigning the tasks to the workstations by considering precedence relationships of tasks to optimize an objective. SALBP consists of two basic types which are SALBP-I and SALBP-II. In SALBP-I, cycle time is given and objective is to minimize the number of workstation. In SALBP-II, number of workstations is given and objective is to minimize the cycle time.

In classical SALBP setting, it is assumed that processing time of a task is an exogenous parameter, and assumed to be constant. However, in real life, skills and experience of the workers are different. For this reason, extending SALBP and considering worker dependent processing times for tasks and integrating the decision of worker assignment to workstations within the problem may further ease the implementation of SALBP solution in real life cases. This integrated problem is called Assembly Line Worker Assignment and Balancing Problem (ALWABP) in (Miralles *et al.*, 2007). Similar to SALBP, ALWABP also consists of two basic types which are ALWABP-I and ALWABP-II.

This study is conducted on TV production of Vestel Electronics Company. We focus on assembly line balancing and worker assignment problem, where multiple production lines are executed by a shared pool of workers with varying skills and having absenteeism issues. Problem is in large scale in terms of multiple parallel assembly lines, high product and task variety, high number of worker. Difficulty in this problem is to solve a large scale NP-hard problem within very limited time. There are only two minutes to run and solve the problem for multiple lines and shared pool of workers.

Many other studies are conducted on ALWABP by focusing different topics and one of them is Parallel Assembly Line Worker Assignment and Balancing Problem (PALWABP). (Araújo, Costa and Miralles, 2013) has focused on PALWABP. We take inspiration from their mathematical model and designed two mathematical models which are ALWABP-I and ALWABP-II for parallel assembly lines. However, our models differentiate at some points. To reach good solutions in reasonable times, some decision variables and constraints are revised in a way to decrease their number. For ALWABP-II type, we have faced with some nonlinearity cases. Nonlinearity is solved by applying McCormick Relaxation and details are mentioned in section 5. In addition, each production system has some distinctive characteristics and these characteristics are included in the mathematical model to increase the applicability of the model in real life. Due to complexity of the real life cases, exact algorithms do not give good results in a reasonable time for the large instances. Therefore, three decomposition methods are developed as heuristic algorithms in section 5. These heuristic algorithms give good solutions in a reasonable time for large instances.

2. PROBLEM DEFINITION

In this section, general view of the TV production processes and inputs of the system will be explained.

2.1. Fields of Activity for Vestel Electronics

This study is conducted on Vestel Electronics Company of Zorlu Group, which is composed of various companies operating in different sectors: manufacturing, software and technology development, marketing, and distribution fields in the consumer electronics, household appliances, multimedia communication, LED lighting, and defense industries with € 5 billion turnover. Although almost all of these companies have ALB settings, in this study we consider TV production of Vestel Electronics Company, since TV production department has the highest worker number and highest sales turnover among all other departments of Vestel Electronics Company.

2.2. Final Assembly Lines in TV Production

Final assembly lines are the lines, where all work-in-process (WIP) materials of a product are assembled. Final assembly lines are composed of four different sections for TV production. First optical structure of the product, called “panel” and mechanical structure (electronic cards, cables, sockets, cable clips, etc.) are assembled, respectively. Then, aging process is applied to the products and products are tested. Finally, the product is packaged and sent to the delivery warehouse.

In each section, small work packages, called tasks are assembled. These tasks are completed at the workstations successively, via automation or labor force. Various automations are started to be used in TV production lines, but still labor force is majorly used as a resource. Number of final assembly lines actively used changes depending on the monthly order amount. Averagely, 13 assembly lines are actively used and about 600 direct workers are working on these lines in a shift. More or less, 300 workers are assigned to assembly of mechanical structure of the products and this is the most complicated section to balance in an assembly line. First reason is the number of tasks to be assembled in mechanical structure is much higher compared to other sections in a line. The other reason is precedence relationship among tasks are very strict and which tasks will be performed in which workstation is almost certain, so there is no need to balance the line for the sections other than mechanical structure. Hence, in this study, worker assignment and assembly line balancing will be taken into consideration for the section in a line where mechanical structures of the products are assembled.

2.3. Assembly Line Balancing Activities in TV Production

13 assembly lines are actively used in TV production on the average and almost 600 workers are working on these assembly lines in a shift. Workers are assigned to lines from a shared pool, so decision of which worker will be assigned to which line is needed to be made at the beginning of the shift by the supervisors in TV production.

Day shift starts at 08:00 a.m. in TV production and absent workers are listed at 08:00 a.m. by the ERP system since each employee scans their ID cards before entering the factory. From 08:00 a.m. to 08:10 a.m., total Productive Maintenance (TPM) activities which are daily management system and autonomous maintenance are applied in TV production. By using the idle time of the operators due to TPM activities, shift supervisors have 10 minutes to decide and allocate the workers to the lines and stations. Among 10 minutes, 3 minutes are reserved for daily data management and data input, 2 minutes are reserved for the running of the proposed algorithms and 5 minutes are reserved for the allocation of the workers to the lines and stations. Supervisors check absent worker list and product types in each line to make their game plan about allocation of the workers. However, there is no systematic way to make a decision for this issue and these decisions are made based on the experience and intuition of the supervisors. While assigning the workers to the jobs, it is not easy to take worker competencies into consideration by supervisors and it might result in inefficiency in TV production. Since there are too much variables and a limited time, it is hard to reach optimal allocation with human brain. An optimization tool is required to have better results by including real life issues.

2.4. Monthly TV Production Numbers

Monthly order amount changes during a year since demand pattern has an S curve as seen in Figure **1**. The demand starts to increase in September and reaches its peak level in October and November. The season including September, October, November and December is called high season, whereas period between January and July is called as low season. Monthly demand values are the monthly target production numbers due to make-to-order principle. Weekly and monthly scheduling, working hours and crew number is arranged by considering these monthly demand amounts.

Different objectives are used for low and high season in TV production lines. In high season, number of worker is increased to some level to satisfy the high demand on time and the main objective is to increase the total throughput with the given number of worker. In low season, main objective is to satisfy the low demand with the minimum number of worker.

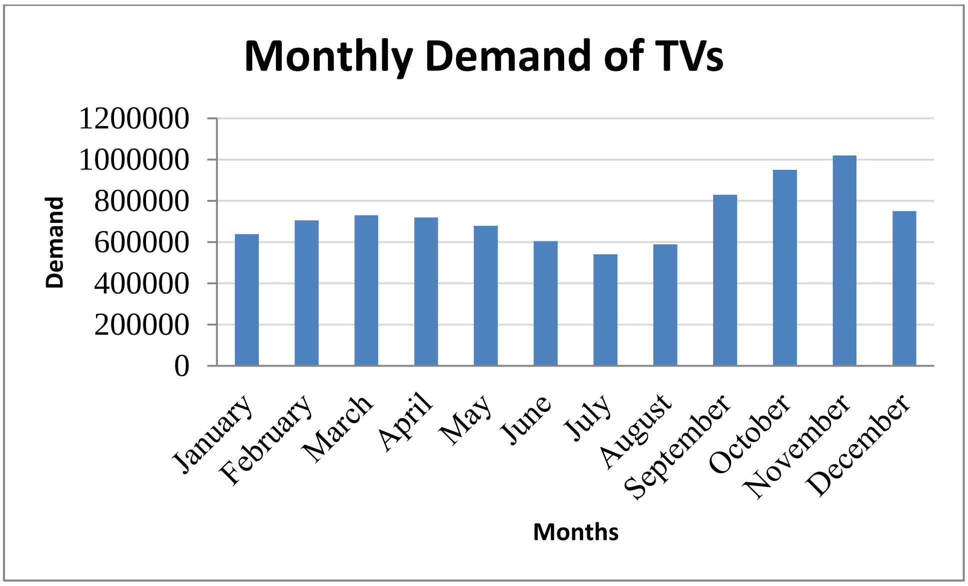

Figure 1: Monthly Demand of TV Products in 2017

2.5. Product Types for TVs

Original Equipment Manufacturing (OEM) strategy is used in Vestel Electronics Company. As an original equipment manufacturer, Vestel produces customized products for different brands and this increases the product variety dramatically. Basically, products are differentiating in some points and these are listed below.

- Size (22" – 75")
- Optical Structure (3 main structures)
- Mechanical Structure (4 main structures)
- Extra Properties (Wi-Fi, Bluetooth, DVD, etc.)

The decision of worker allocation from a shared pool to assembly lines becomes much harder in high product variety. Since task variety depends on the product variety, high product variety causes task variety to increase. In this case, line balancing also becomes harder.

2.6. Schedule for Production Plan

Orders for different product groups are received from customers and make-to-order system is used in Vestel. Long term production plan is prepared based on the certain orders for one month, and short term production schedule for two weeks is prepared by considering the arrival date of the WIP materials and due dates of the orders. TVs are produced in the final assembly lines based on the production schedule.

Production schedule for final assembly lines is performed by Production Planning Department and Final Assembly Department does not violate the schedule except shortage of material or technical problems for manufacturing of the product type. Similar types of products are tried to be scheduled successively to decrease the setup time in the line and to make worker allocation easier. When different product types are scheduled successively, worker allocation between lines becomes much harder. For instance, product type A requires extra 4 workers for screwing process compared to product type B. If product type changes from type B to type A, extra 4 workers are needed and these workers are needed to be transferred from other assembly lines with the minimum negative effect on the efficiencies of these lines. If that kind of product type changes occur several times in a shift, managing the worker allocation becomes difficult. Thus, production schedule is an important input for this study.

2.7. Worker Absenteeism Rates and Competencies

(Cohen, 2012) has analyzed the effects of the absenteeism in assembly lines and the results show that assigning inexperienced workers as substitutes for absent workers has a major potential for generating bottleneck stations. A certain number of absenteeism rates are observed in each shift in TV production. If there is an absent worker in a line, needed worker is fulfilled from other assembly lines. However, it is hard to decide from which line that worker need will be fulfilled. The reason is parameters like product types in each line, production number with different worker number for a product group and worker competencies etc. should be taken into consideration. Otherwise, total TV production number could be affected negatively.

Each worker has competences to perform different tasks e.g. screwing, cabling, and optical control test. If a worker has a competency on only screwing process, he should not be assigned to tasks other than screwing. Even if he is assigned to other tasks, he performs those tasks much slower than the standard processing time of that task. Hence, to prevent these inefficient assignments, workers should be assigned to task if and only if he has a competence for that task.

Task processing time differs among different workers although they all have a competence on that task. Competence levels for workers are classed in four categories as low level, lower-middle level, upper-middle level and high level. Processing time of the tasks are determined based on the processing speed of the workers and it is multiplied by a coefficient of 2, 1.5, 1.2 and 1 for low, lower-middle, upper-middle and high level competence, respectively. To illustrate, if processing time of a task is 20 sec. for a worker having high level competence, then it is assumed as 24 sec (20*1,2) for upper-middle level, 30 sec (20*1,5) for lower-middle level and 40 sec (20*2) for low level experienced workers. Hence, competence levels of the workers are taken into consideration when assigning workers to the tasks to have more realistic results.

3. LITERATURE REVIEW

(Scholl, Becker and C., 2006), (Becker and Scholl, 2006) have categorized the main characteristics of SALBP in detail as shown in Figure 2.

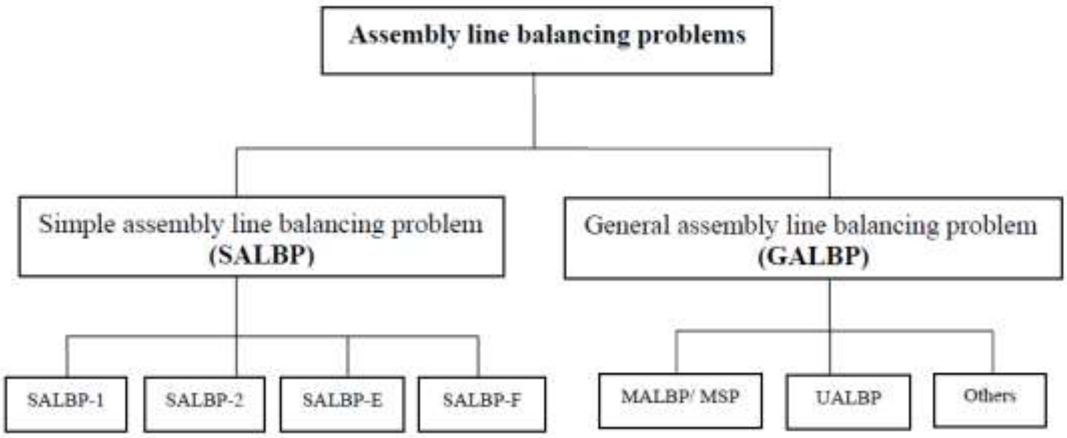

Figure 2: Classification of Assembly Line Balancing (Scholl, Becker and C., 2006)

Simple assembly line balancing problem (SALBP) is consisted of single model assembly line where only precedence constraints between tasks are considered. Generally, studies are conducted on two basic types of SALBP which are SALBP-I and SALBP-II. SALBP-I aims to minimize number of workstations with a given cycle time. SALBP-II aims to minimize cycle time for a given number of workstations.

In this study, we will focus on Assembly Line Balancing and Worker Assignment Problem (ALWABP) which is a variant of Assembly Line Balancing Problem (ALBP). ALWABP first studied by (Miralles et al., 2007) for the sheltered work centers for disabled (SWD) and it is discussed in many studies recently. Some ALWABP papers are listed in Table 1 in chronological order.

Table 1: Literature Review for ALWABP

Paper	Problem	Solution Method	Assembly Line Type
Miralles et. al. (2007)	ALWABP - 2	Mixed integer linear programming	Single line
Cardós et. al. (2008)	ALWABP - 2	Branch and Bound Algorithm	Single line
Costa et. al. (2009)	ALWABP - 2	Decomposition method	Single line
Moreira et. al. (2010)	ALWABP - 2	Constructive heuristic methods and Hybrid genetic algorithm	Single line
Blum et. al. (2011)	ALWABP - 2	Beam search algorithm	Single line
Zaman et. al. (2012)	Obj.1: ALWABP - 2 Obj.2: Max Productivity Obj.3: Min Idle Time	Genetic algorithm	Single line
Araújo et. al. (2013)	ALWABP - 2	Mixed integer linear programming and heuristic solution	Single line
Khemyong et. al. (2013)	Obj.1: ALWABP - 1 Obj.2: Min Walking Time Obj.3:Max Full Work	Mixed integer linear programming	U - shaped line
Mutlu et. al. (2013)	ALWABP - 2	Iterative genetic algorithm	Single line
Borba et. al. (2013)	ALWABP - 2	Algorithm 1: Randomized Beam Search Algorithm Algorithm 2: Branch and Bound Algorithm	Single line
Araújo et. al. (2013)	ALWABP - 2	Algorithm 1: Mixed integer linear programming Algorithm 2: The tabu search algorithm Algorithm 3: The biased random-key genetic algorithm	Parallel lines
Vilà et. al. (2014)	ALWABP - 2	Branch and Bound Algorithm	Single line
Borba et. al. (2014)	ALWABP - 2	Algorithm 1: Randomized Beam Search Algorithm Algorithm 2: Branch and Bound Algorithm	Single line
Castellucci et. al. (2014)	ALWABP - 2	Simulation Model	Single line
Moreira et. al. (2015)	ALWABP - 2	Constructive insertion heuristic	Single line
Ramezanian et. al. (2015)	ALWABP - 1 and Min Worker Cost	Imperialist competitive algorithm	Mixed-model assembly line
Moreira et. al. (2015)	ALWABP - 2	Algorithm 1: Mixed integer linear programming Algorithm 2 :constructive heuristic	Single line
Ritt et. al. (2015)	ALWABP - 2	Mixed integer linear proramming and Heuristic assignment procedures	Single line
Akyol et. al. (2016)	ALWABP - 2	Multiple Rule Based Constructive Randomized Search Algoritm	Single line
Polat et. al. (2016)	ALWABP - 2	Two-phase variable neighbourhood search algorithm	Single line
Zacharia et. al. (2016)	ALWABP - 2 and Min Workload Smoothness	Multi - objective evolutionary algorithm	Single line
Akyol et. al. (2016)	ALWABP - 2	Multiple-rule based constructive randomized search	Single line

Table 1 shows that most of the studies are related to single assembly line. Different from those, U – shaped assembly line is studied by (Khemyong and Sirovetnukul, 2013), parallel assembly lines are studied by (Araújo, Costa and Miralles, 2013) and mixed model assembly line is studied by (Ramezanian and Ezzatpanah, 2015). In this study, we work on parallel assembly lines since Vestel actively uses 13 parallel assembly lines and workers are assigned to those parallel lines from a shared pool in TV production.

Most of the studies in Table 1 are about ALWABP - 2. However, only a few studies have conducted on ALWABP - 1, multi objective, maximizing productivity, minimizing idle time or minimizing walking distance. (Zaman, Paul and Azeem, 2012) provides different objective functions which are ALWABP - 2, maximizing the productivity and minimizing idle time for their Genetic Algorithms (GA). (Khemyong and Sirovetnukul, 2013) has conducted their studies on U-shaped line with different objectives, ALWABP - 1, minimizing walking distance of the workers and maximizing full work. (Moreira et al., 2015) and (Moreira *et al.*, 2015) have considered ALWIBP - 1 for their studies on Assembly Line Worker Integration and Balancing Problem. (Ramezanian and Ezzatpanah, 2015) has dealt with multi objective modeling by aiming to minimize the total cycle time and operating costs for a mixed model assembly line. (Zacharia and Nearchou, 2016) has presented a multi objective evolutionary algorithm (MOEA) by minimizing the cycle time and workload smoothness.

ALBP is known as an NP-hard problem. It is hard to get optimal results in a reasonable time for the large scale problems. In ALWABP, additional decision variables and constraints are used in the models, so ALWABP is also NP-hard problem. Almost all publications in Table 1 provide metaheuristic methods, except (Miralles et al., 2007). (Miralles et al., 2007) gives an integer programming model and run the model for a small case which includes 7 workers and 18 tasks. Result of the mathematical model is good enough compared to real life scenario in terms of production number and number of worker.

However, it is hard to get optimal results for much bigger data set such as 300 workers, 13 production lines, 13 products, 45 tasks which are realized in Vestel. Although (Araújo et al., 2012), (Araújo, Costa and Miralles, 2013), (Moreira *et al.*, 2015) and (Ritt, Costa and Cristóbal, 2015) provide mixed integer programming models for ALWABP - 2, they also give some metaheuristic approaches in their studies since it is hard to reach optimal results with the exact methods for the large instances.

Most popular metaheuristic methods are genetic algorithms, branch and bound algorithms, beam search algorithms, constructive heuristic algorithms for the large instances. Other heuristic methods presented in Table 1 are tabu search algorithm, competitive algorithm, neighborhood search algorithm, evolutionary algorithm and decomposition algorithm.

(Cardós et al., 2008), (Borba and Ritt, 2013), (J., Vilà and Pereira, 2014) and (Borba, Ritt and M., 2014) have proposed branch and bound algorithm in their studies. (Cardós et al., 2008) has presented a branch and bound algorithm with three different search strategies and different parameters. It is stated that Depth First Search with Complete node development strategy gives better results among three possible strategies. (Borba and Ritt, 2013) and (Borba, Ritt and M., 2014) has proposed a task-oriented branch and bound algorithm by developing new reduction rules and lower bounds. (J., Vilà and Pereira, 2014) uses station-oriented branch and bound algorithm based on the branch, bound and remember (BB&R) scheme. They developed new lower bounds, reductions and dominance rules and results show that algorithm verifies the optimality for the large number of instances in benchmark instance set.

(Zaman, Paul and Azeem, 2012) has worked on genetic algorithm for operator assignment in a SALBP, while making assumption that skills of the operators are task independent. In our study, we are also making this assumption to make data management in real life easier. (Moreira *et al.*, 2010) has introduced a hybrid biased random-key genetic algorithm (HGA) based on the constructive heuristics they developed. They optimize the task priorities of a station-based assignment scheme inside HGA. (Mutlu *et al.*, 2013) developed an iterative genetic algorithm (IGA) with three search approaches: modified bisection search, genetic algorithm and iterated local search. Their results show that IGA is robust enough for the large set of benchmark problems. (Araújo, Costa and Miralles, 2013) has applied biased random-key genetic algorithm (BRKGA) for the parallel assembly line worker assignment and balancing problem (PALWABP).

(Blum, Miralles and C., 2011) has presented an iterative algorithm based on beam search for ALWABP - 2. In first phase, algorithm aims to find an initial cycle time for the corresponding problem. In second phase, algorithm iteratively reduces the cycle time until problem cannot be solved. For both phases, beam search algorithm is used. (Borba and Ritt, 2013) and (Borba, Ritt and M., 2014) developed a probabilistic beam search algorithm by searching for a small cycle time solving the feasibility problem ALWABP-F from different candidate cycle times.

(Moreira *et al.*, 2010) has introduced a constructive heuristic algorithm based on task and worker assignment priority rules for ALWABP-2. (Moreira et al., 2015) has proposed a constructive insertion heuristic (CIH). Algorithm first finds a SALBP-1 solution and then this solution is adapted to involve heterogeneous workers iteratively for ALWIBP-1. (Moreira *et al.*, 2015) has presented a robust insertion constructive heuristic (RICH) which is much faster than branch and bound algorithm and provides high quality solutions. (Akyol and Baykasoglu, 2016) has worked on a multiple-rule based constructive randomized search algorithm (MRBCRS) by defining thirty nine task priority rules and four worker priority rules. (Demirkol Akyol *et al.*, 2016) has proposed a MRBCRS algorithm by updating current algorithm to incorporate ergonomic risk factors for ErgoALWABP problem.

(Araújo, Costa and Miralles, 2013) has introduced a tabu search algorithm composed of two steps. First step is to generate sets of tasks that each worker can perform. Second step is to select one set of tasks from each worker so that each task is covered by at least one set. (Ramezanian and Ezzatpanah, 2015) has worked on an imperialist competitive algorithm (ICA) which is a variant of evolutionary algorithm. ICA algorithm is a socio-political based algorithm and involves imperialistic procedures such as assimilation, revolution and imperialistic competition. (Polat *et al.*, 2016) has proposed a two phase variable neighborhood search (VNS) algorithm. In first phase, tasks are assigned to workstations by applying VNS approach and in second phase, workers are assigned to workstations by applying a variable neighborhood descent method.

4. METHODOLOGY

We will present exact and heuristic algorithms to solve ALWABP problem for two different objective functions, ALWABP - 1 and ALWABP - 2. Mixed Integer Linear Programming (MILP) method is used for the exact algorithms and decomposition method is applied for the heuristic algorithms. Some assumptions are made for these methodologies and they are listed as:

- All input parameters are deterministic.
- A task cannot be split into smaller parts.
- Tasks should be processed considering the precedence relationships.
- All tasks must be processed.
- Task processing times are fixed and independent from workstations.
- Assembly line is for a single unique product.
- Only one operator is assigned to each workstation.
- Operator’s skill coefficient is task and station independent.

4.1. EXACT ALGORITHMS

4.1.1. ALWABP - 1

In this exact method, we will present the mathematical model for ALWABP-1 problem. Original mathematical model is presented by (Araújo, Costa and Miralles, 2013) and model is revised by adding some specific notation, parameters, decision variables and constraints for Vestel TV production.

Table 2: Notation for ALWABP - 1

Notation	
w	Worker
r	Product
i, j	Task
k	Line
s	Station
W	Set of workers
R	Set of products
N	Set of tasks
K	Set of lines
S	Set of stations
D_j	Set of tasks that precede task *j* in the precedence graph
I_{wi}	Set of compatible assignments for worker – task (*w*, *i*)
P_{ri}	Total processing time for task *t* of product *r*
CT_k	Cycle time (sec) of line *k*
EC_w	Experience coefficient of worker *w*
A_w	Set of absenteeism for worker *w*
X_{srik}	Binary variable. Equal to 1 only if task *i* of product *r* is assigned to station *s* of line *k*
Y_{swk}	Binary variable. Equal to 1 only if worker *w* is assigned to station *s* of line *k*
$ExpC_{ks} \in \mathbb{R}^+$	Continuous variable. Experience coefficient in station *s* of line *k*

Table 3: Additional Notation for Vestel Specific Parameters

Additional Notation for Vestel Specific Parameters	
TLS_{iks}	Set of compatibility of task between station of line (*i*, *k*, *s*)
PT_{ri}	Set of task availability in a product (*r*, *i*)
LST_{ks}	Set of station availability in a line (*k*, *s*)
SCH_{kr}	Set of schedule for line and product (*k*, *r*)

According to this notation, mathematical model can be written as

$$[\text{MIN}] \quad \sum_s \sum_w \sum_k Y_{swk} \tag{1}$$

Subject to:

$$\sum_k \sum_s X_{srik} = 1 \qquad \forall r \in R,\ \forall i \in N \tag{2}$$

$$\sum_k \sum_s Y_{swk} \le A_w \qquad \forall w \in W \tag{3}$$

$$\sum_w Y_{swk} \le 1 \qquad \forall k \in K,\ \forall s \in S \tag{4}$$

$$\sum_k \sum_s X_{srik} \cdot s \le \sum_k \sum_s X_{srjk} \cdot s \qquad \forall r \in R, \forall i, j \in N \mid i \in D_j \tag{5}$$

$$\sum_r \sum_i X_{srik} \le |N| \cdot \sum_w Y_{swk} \qquad \forall k \in K,\ \forall s \in S \tag{6}$$

$$ExpC_{ks} = \sum_w Y_{swk} / EC_w \qquad \forall k \in K,\ \forall s \in S \tag{7}$$

$$\sum_r \sum_i X_{srik} \cdot P_{ri} \le CT_k \cdot ExpC_{ks} \qquad \forall k \in K,\ \forall s \in S \tag{8}$$

$$X_{srik} + Y_{swk} \le 1 \qquad \forall w \in W, \forall r \in R,\ \forall i \in N, \forall k \in K\ \forall s \in S \big| I_{wi} = 0 \tag{9}$$

$$\sum_r X_{srik} \le TLS_{iks} \qquad \forall i \in N,\ \forall k \in K,\ \forall s \in S \tag{10}$$

$$\sum_k \sum_s X_{srik} \le PT_{ri} \qquad \forall i \in N,\ \forall r \in R \tag{11}$$

$$\sum_r \sum_i X_{srik} \le M \cdot LST_{ks} \qquad \forall k \in K,\ \forall s \in S \tag{12}$$

$$\sum_i \sum_s X_{srik} \le M \cdot SCH_{kr} \qquad \forall k \in K,\ \forall r \in R \tag{13}$$

Objective function (1) minimizes the total number of worker. Constraint (2) guarantees that each task is assigned to one station. Constraint (3) ensures that workers can be assigned to a station if s/he is available in the shift. Constraint (4) enforces at most one worker is assigned to a station. Constraint (5) implies that precedence relationship among tasks cannot be violated. Constraint (6) ensures that tasks are assigned to a station only if there is a worker in that station. Constraint (7) and (8) define the cycle time for each line as the maximum execution time among each station by considering worker competency level. Constraint (9) guarantees that a worker is assigned to a task if that worker has a competence for that task. Constraints (10), (11), (12) and (13) are Vestel specific constraints. Constraint (10) ensures that a task is assigned to a station only if that task is compatible with that station. Constraint (11) implies that a task is assigned to a station only if product includes that task. Constraint (12) guarantees that a worker is assigned to a station only if that workstation is available in that line since number of station in each line changes from 18 to 21. Constraint (13) defines that products are produced in final assembly lines by considering the schedule.

Table 4: Number of Decision Variables for (Araújo, Costa and Miralles, 2013) and Our Model

			# of Decision Variables
Araujo et. al.	**X[s][w][r][i][k]**		45.630.000
	s	20	
	w	300	
	r	13	
	i	45	
	k	13	
Our Model	**X[s][r][i][k]**		152.100
	s	20	
	r	13	
	i	45	
	k	13	

To decrease the complexity of the model, number of decision variables is aimed to be decreased. When we modify the model of (Araújo, Costa and Miralles, 2013) to our problem, decision variable of X includes five indices which are worker, line, workstation, product and task. Decision variable number is calculated for this case in Table 4 which is over 45 million.

However, we revise the decision variable and corresponding constraints by excluding worker index from this decision variable. In this way, decision variable number decreases from 45 million to 150 thousand. This arrangement contributes to get better solutions in a shorter time period.

4.1.2. ALWABP - 2

In this exact method, we will present the mathematical model for ALWABP-2 problem. Common constraints with ALWABP - 1 algorithm will be indicated by their numbers instead of rewrite it entirely.

Table 5: Additional Notation for ALWABP-2

Additional Notation for ALWABP-2	
H	Coefficient representing the total seconds in an hour
X_{srik}	Binary variable. Equal to 1 only if task *i* of product *r* is assigned to station *s* of line *k*
Y_{swk}	Binary variable. Equal to 1 only if worker *w* is assigned to station *s* of line *k*
$ExpC_{ks} \in \mathbb{R}^{+}$	Continuous variable. Experience coefficient in station *s* of line *k*
$CT_k \in \mathbb{R}^{+}$	Continuous variable. Cycle time (sec) of line *k*

In general, objective function in ALWABP-2 is to minimize cycle time. However, (Araújo, Costa and Miralles, 2013) has presented that instead of minimizing cycle time, maximizing total throughput is more applicable objective function for parallel assembly lines. The reason will be explained with a small case study. Assume that we have two parallel production lines and their cycle times are 25 sec, 30 sec and their throughput rates are 144 TV/hr and 120 TV/hr respectively. Total throughput rate will be 264 TV/hr. Throughput needs to be calculated by dividing cycle time (CT) to coefficient representing the total seconds in an hour (H).

Table 6: Small Case Representing Cycle Time and Throughput of Lines

	Cycle Time (sec)	Throughput (TV/hr)
Line 1	25	144
Line 2	30	120

$$[\text{MAX}] \qquad \sum_k \frac{H}{CT_k} \tag{1.a}$$

Objective is to maximize total throughput.

Subject to:

Constraints (2), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13) in ALWABP - 1 model are also valid for this model.

$$\sum_k \sum_s Y_{swk} = A_w \qquad \forall w \in W \tag{3.a}$$

$$\sum_r \sum_i X_{srik} \cdot P_{ri} \leq CT_k \cdot ExpC_{ks} \qquad \forall k \in K,\ \forall s \in S \tag{8}$$

Objective (1.a) is to maximize total throughput. Constraint (3) is replaced with (3.a) and it enforces workers to be assigned to exactly one station if s/he is available in the shift.

Due to objective function (1.a) and constraint (8), model becomes nonlinear. In this scenario, objective function is **nonlinear** since a decision variable is in the denominator of a division ($\frac{H}{CT_k}$). Similarly, constraint related to total processing time in a station cannot exceed the cycle time is also **nonlinear** since two decision variables are multiplied ($CT_k \cdot ExpC_{ks}$).

To convert this nonlinear model to linear model, some decision variables are needed to be added and some calculations should be made.

$$\frac{H}{CT_k} = TR_k$$

$$\sum_r \sum_i X_{srik} \cdot P_{ri} \leq \frac{H}{TR_k} \cdot ExpC_{ks}$$

$$(\sum_r \sum_i X_{srik} \cdot P_{ri}) \cdot TR_k \leq H \cdot ExpC_{ks}$$

$$0 \leq H \cdot ExpC_{ks} - TR_k \cdot (\sum_r \sum_i X_{srik} \cdot P_{ri})$$

$$Z_{ks} = \sum_r \sum_i X_{srik} \cdot P_{ri}$$

$$0 \leq H \cdot ExpC_{ks} - TR_k \cdot Z_{ks}$$

$$Mc_{ks} = TR_k \cdot Z_{ks} \qquad \text{NONLINEAR!!}$$

$$0 \leq H \cdot ExpC_{ks} - Mc_{ks}$$

It is **still nonlinear** since two continuous decision variables are multiplied. Then, **McCormick Relaxation** will be applied to get linear model.

Table 7: Additional Notations for ALWABP – 2 Model

Additional Notations for ALWABP – 2 Model	
TR_k^L	Lower bound for throughput of line *k*
TR_k^U	Upper bound for throughput of line *k*
Z_{ks}^L	Lower bound for total processing time in station *s* of line *k*
Z_{ks}^U	Upper bound for total processing time in station *s* of line *k*
$TR_k \in \mathbb{R}^+$	Continuous Variable. Hourly throughput of the line *k*
$Mc_{ks} \in \mathbb{R}^+$	Continuous Variable. McCormick relaxation for multiplication of throughput and total processing time in station *s* of line *k*
$Z_{ks} \in \mathbb{R}^+$	Continuous Variable. McCormick relaxation for the total processing time in station *s* of line *k*

$$\text{MAX} \quad \sum_k TR_k \tag{14}$$

Subject to:

Constraints (2), (4), (5), (6), (7), (9), (10), (11), (12), (13) in ALWABP - 1 model are also valid for this model. Additionally, constraints below will be added.

$$\sum_k \sum_s Y_{swk} = A_w \qquad \forall w \in W \tag{3.a}$$

$$0 \le H \cdot ExpC_{ks} - Mc_{ks} \qquad \forall k \in K,\ \forall s \in S \tag{15}$$

$$Z_{ks} = \sum_r \sum_i X_{srik} \cdot P_{ri} \qquad \forall k \in K,\ \forall s \in S \tag{16}$$

$$Mc_{ks} \ge TR_k^L \cdot Z_{ks} + TR_k \cdot Z_{ks}^L - TR_k^L \cdot Z_{ks}^L \qquad \forall k \in K,\ \forall s \in S \tag{17}$$

$$Mc_{ks} \ge TR_k^U \cdot Z_{ks} + TR_k \cdot Z_{ks}^U - TR_k^U \cdot Z_{ks}^U \qquad \forall k \in K,\ \forall s \in S \tag{18}$$

$$Mc_{ks} \le TR_k^U \cdot Z_{ks} + TR_k \cdot Z_{ls}^L - TR_k^U \cdot Z_{ks}^L \qquad \forall k \in K,\ \forall s \in S \tag{19}$$

$$Mc_{ks} \le TR_k \cdot Z_{ks}^U + TR_k^L \cdot Z_{ks} - TR_k^L \cdot Z_{ks}^U \qquad \forall k \in K,\ \forall s \in S \tag{20}$$

$$TR_k \ge TR_k^L \qquad \forall k \in K \tag{21}$$

$$TR_k \le \cdot TR_k^U \qquad \forall k \in K \tag{22}$$

$$Z_{ks} \ge Z_{ks}^L \qquad \forall k \in K,\ \forall s \in S \tag{23}$$

$$Z_{ks} \le Z_{ks}^U \qquad \forall k \in K,\ \forall s \in S \tag{24}$$

Objective (14) is to maximize total throughput. Constraint (3) is replaced with (3.a) and it enforces workers to be assigned to exactly one station if s/he is available in the shift. Constraints (15) - (24) are designed for McCormick Relaxation.

(Araújo, Costa and Miralles, 2013) has also nonlinear objective function and to linearize the model, their objective function is converted into maximizing throughput. In our case, we need to linearize both objective function and the 8th constraint which is cycle time cannot be exceeded in a workstation. To make linearization, different method from (Araújo, Costa and Miralles, 2013) is used, McCormick Relaxation. The reason why different linearization method is preferred is the definition of the worker-task compatibility parameter. (Araújo, Costa and Miralles, 2013) defines the task-worker compatibility parameter as p_{wi} representing the time spent by worker *w* to perform task *i*. Instead of this parameter, we define worker-task compatibility with three different parameters I_{wi}, EC_w, P_{ri}. The reason is to be able to manage the data complexity in real life.

Table 8 compares the number of parameters needed for the worker-task compatibility between (Araújo, Costa and Miralles, 2013) and our model. For each worker, processing time of each task of each product is determined and 1.350.000 parameters are needed to be prepared in the model of (Araújo, Costa and Miralles, 2013). However, in our model only 18.300 parameters are used for worker-task compatibility. Since our aim is to develop an applicable mathematical model for Vestel TV Production, data set should be manageable.

Table 8: Worker-Task Compatibility Parameters for (Araújo, Costa and Miralles, 2013) and Our Model

<table>
<tr><th></th><th></th><th></th><th># of Parameters</th><th># of Parameters in the Model</th></tr>
<tr><td rowspan="4">(Araújo, 2011)</td><td colspan="2">P[w][r][i]</td><td rowspan="4">1.350.000</td><td rowspan="4">1.350.000</td></tr>
<tr><td>w</td><td>300</td></tr>
<tr><td>r</td><td>100</td></tr>
<tr><td>i</td><td>45</td></tr>
<tr><td rowspan="8">Our Model</td><td colspan="2">I[w][i]</td><td rowspan="3">13.500</td><td rowspan="8">18.300</td></tr>
<tr><td>w</td><td>300</td></tr>
<tr><td>i</td><td>45</td></tr>
<tr><td colspan="2">P[r][i]</td><td rowspan="3">4.500</td></tr>
<tr><td>r</td><td>100</td></tr>
<tr><td>i</td><td>45</td></tr>
<tr><td colspan="2">EC[w]</td><td rowspan="2">300</td></tr>
<tr><td>w</td><td>300</td></tr>
</table>

It is mentioned that Vestel is an OEM / ODM company and it produces not only its own brand, but also different brands. Consider that Vestel made a new agreement with a customer and this customer will probably order different type of products in terms of size, mechanical structure, technology level, etc. Shortly, number of products in Vestel can change easily. Table 9 shows how number of parameters for worker-task compatibility is affected with the increase of product number for both mathematical model. 20% increase in product number increases the number of parameters as 270.000 for (Araújo, Costa and Miralles, 2013), whereas it increases the number of parameters for our model only 900. These numbers also show that data management for worker-task compatibility is easier in our model.

Table 9: Worker-Task Compatibility Parameters for Different Product Numbers

	# of Parameters for Worker-Task Compatilibity	
# of products (r)	(Araújo, 2011)	Our Model
100	1.350.000	18.300
110	1.485.000	18.750
120	1.620.000	19.200

4.2. HEURISTIC 1

In this section, we will propose a decomposition method as a heuristic algorithm. Structure of decomposition method is figured as a flow chart in Figure 3. ALWABP problem is decomposed into sub-problems in a hierarchical framework. First sub-problem is related to assigning tasks to workstations and output of this sub-problem is checked for infeasibility cases by an assignment problem algorithm. If an infeasible case is realized, a feedback mechanism steps in and a cut generation is added into first sub-problem. This sub-problem is solved again and feedback mechanism proceeds iteratively until a feasible case is achieved. Then, next sub-problem is solved by assigning workers to corresponding tasks and workstations to reach final results.

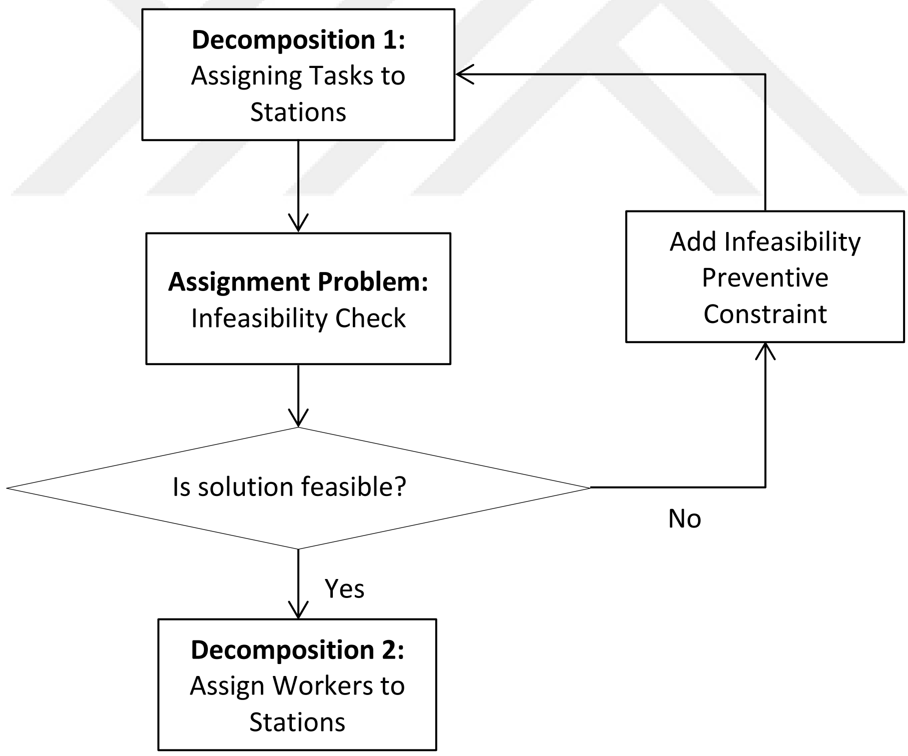

Figure 3: Flow Chart for Heuristic 1

4.2.1. ALWABP - 1 Decomposition Part I

Table 10: Notation for ALWABP - 1 Decomposition Part I

X_{srik}	Binary variable. Equal to 1 only if task *i* of product *r* is assigned to station *s* of line *k*
Y_{sk}	Binary variable. Equal to 1 only if station *s* of line *k* is used

$$[\text{MIN}] \quad \sum_{s}\sum_{k} Y_{sk} \tag{1.b}$$

Subject to:

Constraints (2), (5), (10), (11), (12), (13) in ALWABP - 1 model are also valid for this model.

$$\sum_{r}\sum_{i} X_{srik} \le |N| \cdot Y_{sk} \qquad \forall k \in K,\ \forall s \in S \tag{6.b}$$

$$\sum_{r}\sum_{i} X_{srik} \cdot P_{ri} \le CT_{k} \cdot Y_{sk} \qquad \forall k \in K,\ \forall s \in S \tag{8.b}$$

Objective function (1b) minimizes the total station number actively used. Constraint (6b) ensures that tasks are assigned to a station only if that station is actively used. (8.b) define the cycle time for each line as the maximum execution time among each station that line.

4.2.2. ALWABP - 1 Decomposition Part II

Table 11: Notation for ALWABP - 1 Decomposition Part II

X_{srik}	Parameter. Equal to 1 only if task *i* of product *r* is assigned to station *s* of line *k*
Y_{swk}	Binary variable. Equal to 1 only if worker *w* is assigned to station *s* of line *k*
$ExpC_{ks} \in \mathbb{R}^{+}$	Continuous variable. Experience coefficient in station *s* of line *k*

$$[\text{MIN}] \quad \sum_{s}\sum_{w}\sum_{k} Y_{swk} \tag{1}$$

Subject to:

Constraint (3), (4), (6), (7), (8) are also valid for this model.

$$X_{srik} + Y_{swk} \leq 1 \qquad \forall w \in W, \forall r \in R, \ \forall i \in N, \forall k \in K, \forall s \in S \middle| I_{wi} = 0 \ \&\& X_{srik} = 1 \tag{9.b}$$

Constraint (9.b) guarantees that a worker is assigned to a task if that worker has a competence for that task. This constraint is revised to decrease the total constraint number in such a way that constraint is used only for the case if a task is assigned to that station.

4.2.3. ALWABP - 2 Decomposition Part I

Table 12: Notation for ALWABP - 2 Decomposition Part I

X_{srik}	Binary variable. Equal to 1 only if task i of product r is assigned to station s of line k
$TR_k \in \mathbb{R}^+$	Continuous Variable. Hourly throughput of the line *k*
$Mc_{ks} \in \mathbb{R}^+$	Continuous Variable. McCormick relaxation for multiplication of throughput and total processing time in station *s* of line *k*
$Z_{ks} \in \mathbb{R}^+$	Continuous Variable. McCormick relaxation for the total processing time in station *s* of line *k*

$$\text{MAX} \quad \sum_{k} TR_k \tag{14}$$

Subject to:

Constraints (2), (5), (10), (11), (12), (13), (14), (15), (16), (17), (18), (19), (20), (21), (22), (23) in ALWABP - 2 model are also valid for this model.

Objective (14) is to maximize total throughput.

4.2.4. ALWABP - 2 Decomposition Part II

Table 13: Notation for ALWABP - 2 Decomposition Part II

X_{srik}	Parameter. Equal to 1 only if task *i* of product *r* is assigned to station *s* of line *k*
Y_{swk}	Binary variable. Equal to 1 only if worker *w* is assigned to station *s* of line *k*
$ExpC_{ks} \in \mathbb{R}^{+}$	Continuous variable. Experience coefficient in station *s* of line *k*
TR_k	Continuous variable. Hourly throughput of the line *k*

$$\text{MAX} \quad \sum_{k} TR_k \tag{14}$$

Subject to:

Constraint (3.a), (4), (6), (7), (15) in ALWABP - 2 are also valid for this model.

$$X_{srik} + Y_{swk} \leq 1 \qquad \forall w \in W, \forall r \in R, \ \forall i \in N, \forall k \in K, \forall s \in S \middle| I_{wi} = 0 \ \&\& X_{srik} = 1 \tag{9.b}$$

Objective (14) is to maximize total throughput. Constraint (9.b) guarantees that a worker is assigned to a task if that worker has a competence for that task.

4.2.5. Infeasibility Check

In heuristic 1, decomposition part 1 is run firstly and output of this model is input into decomposition part 2. However, there is a possibility for infeasibility in this scenario since tasks are assigned to stations by disregarding the workers. We have faced with two infeasibility cases during this process.

4.2.5.1. Infeasibility Case I

A small case representing the infeasibility situation will be proposed in this section.

Table 14: Small Case for Worker-Task Compatibility

Worker \ Task	1	2	3	4	5
1	1	1	0	0	0
2	0	1	1	1	1

Let's say worker - task compatibility is assumed as in Table 14. Since Decomposition part 1 does not regard the worker competency, tasks can be assigned to stations as T1, T2 and T3 in station 1, T4 and T5 in station 2. When Decomposition-2 model is run with this input, solution becomes infeasible. The reason is none of the workers have a competency for all tasks in station 1 which are T1, T2, and T3. To prevent such situations, we have developed an assignment problem algorithm.

Assignment Problem

Table 15: Notation for Assignment Problem

β_{wks} **	Cost incurred by assigning worker *w* to line *k* of station *s*
X_{srik}	Parameter. Equal to 1 only if task *i* of product *r* is assigned to station *s* of line *k*
Y_{swk}	Binary variable. Equal to 1 only if worker *w* is assigned to station *s* of line *k*

** β_{wks} represents the cost incurred by assigning worker *w* to line *k* of station *s* and it is calculated by an algorithm presented after assignment problem model.

$$[\text{MIN}] \quad \sum_{w}\sum_{k}\sum_{s} Y_{swk} \cdot \beta_{wks} \tag{25}$$

Subject to:

$$\sum_{k}\sum_{s} Y_{swk} = A_w \qquad \forall w \in W \tag{26}$$

$$\sum_{w} Y_{swk} \leq 1 \qquad \forall k \in K,\ \forall s \in S \qquad (27)$$

Objective (25) is to minimize total cost incurred by assigning workers to the stations of the lines. Constraint (26) forces that each worker must be assigned to a line if s/he is available in the shift. Constraint (27) ensures that at most one worker can be assigned to a station.

Infeasibility Check and Feedback Algorithm

Solve ALWABP Decomposition 1 model: Let X_{srik} be the optimal solution.
input: an instance (*R,N,K,S,W*), task – station assignment X_{srik} , absenteeism of the workers A_w and competency of the workers I_{wi}
for $k = 1 \ldots K$ **do**
$r = k$
  for $s = 1 \ldots S$ **do**
  Check = ∅ , count = 0
    for $i = 1 \ldots N$ **do**
      if $X_{srik} = 1$ **then**
      Check = Check U { *i* } , count = count +1
        for $w = 1 \ldots W$ **do**
        if count = 1 && $I_{wcheck_0} = 1$ **then**
        $\beta_{wks} = 1$
        else
        b = 1
          for a = 0 ... count - 1
            if $I_{wcheck_a} = I_{wcheck_{a+1}}$ && $I_{wcheck_a} = 1$ **then**
            b = b+1
          end for
          if b = count **then**
          $\beta_{wks} = 1$
        if $\beta_{wks} = 0$ **then**
        $\beta_{wks} = \infty$
        end for
    end for
  end for
end for
Solve Assignment Problem: Let Y_{swk} be the optimal solution
forall $k \in K, s \in S$
infeasiblecheck$_{ks}$ =0
end for
forall $i \in N$, $r \in R, k \in K$, $s \in S$, $w \in W$
  if $Y_{swk} * \beta_{wks}$ >= Very Big Number && $X_{srik} = 1$ **then**
  infeasiblecheck$_{ks}$ = 1
end for
Taskfeedback = ∅
forall $i \in N$, $r \in R, k \in K$, $s \in S$
  if infeasiblecheck$_{ks}$ = 1**then**
  Taskfeedback = Taskfeedback U { *i* }
end for
if Taskfeedback = ∅ **then**
Solve ALWABP Decomposition 2 model.
else
Solve ALWABP Decomposition 1 model and assignment problem algorithm until Taskfeedback = ∅. Then Solve ALWABP Decomposition 2 model.

Output of the decomposition part I which is assignment of tasks to workstations is input into this algorithm. Then, workstations are clustered representing which workstations include which tasks. Information of which workers can perform all tasks in clusters is determined and listed. To illustrate, at the beginning of this section, we assumed that T1, T2 and T3 are in station 1, T4 and T5 are in station 2. Algorithm first checks whether worker 1 can perform all tasks in station 1 or not, and then checks for worker 2 and so on. For each worker and workstation clusters, this information is controlled. If none of the workers can perform all tasks in a workstation, then cost incurred to assign a worker to that station (β_{wks}) takes a value of very big number, otherwise takes value of zero. After all cost parameters are determined, assignment algorithm is run. Since aim of the assignment problem is to minimize the total cost incurred from worker - workstation assignment, model tries to assign workers to suitable workstations as far as possible. To be clear, we will give an example from our small case stated in the beginning of this section. In this small case, worker 1 will be assigned to workstation 1 and worker 2 will be assigned to workstation 2. In this scenario, cost parameter for worker 1 and workstation 1 becomes very big number since worker 1 cannot perform task 3 and cost parameter becomes zero for the assignment of worker 2 and workstation 2. In such an infeasibility case, we are determining these tasks and give feedback to decomposition part I model with constraint (28). For example, in this scenario, constraint will not allow to assign T1, T2 and T3 to same workstation since none of the workers can perform all these tasks. This information is held in “Taskfeedback” cluster in the algorithm.

n_{ks}	Number of tasks assigned to station s of line k in “Taskfeedback” cluster

$$X_{sri_1k} + X_{sri_2k} + \ldots + X_{sri_nk} \leq n_{ks} - 1 \qquad \forall i \in \text{Taskfeedback},\ \forall k \in K,\ \forall s \in S,\ \forall r \in R \qquad (28)$$

4.2.5.2. Infeasibility Case II

In ALWABP-2 decomposition part I, tasks are assigned to workstations by disregarding the workers. Since aim of this model is to maximize the total throughput, tasks are tried to be spread different workstations to decrease the total processing time in each workstation. In this case, it is realized that total number of workstations that tasks are assigned is higher than the total number of worker available in the shift. For this reason, we add constraints (29) and (30) to handle such infeasibility cases.

SU_{ks}	Binary variable. 1 only if station *s* of line *k* is actively used, 0 otherwise.

$$\sum_r \sum_i X_{srik} \leq M * SU_{ks} \qquad \forall k \in K,\ \forall s \in S \qquad (29)$$

$$\sum_k \sum_s SU_{ks} \leq \sum_w A_w \qquad (30)$$

Constraint (29) determines whether stations are actively used or not. Constraint (30) guarantees that total workstation number used in decomposition part I model should be less than or equal to total available number of worker.

4.3. HEURISTIC 2

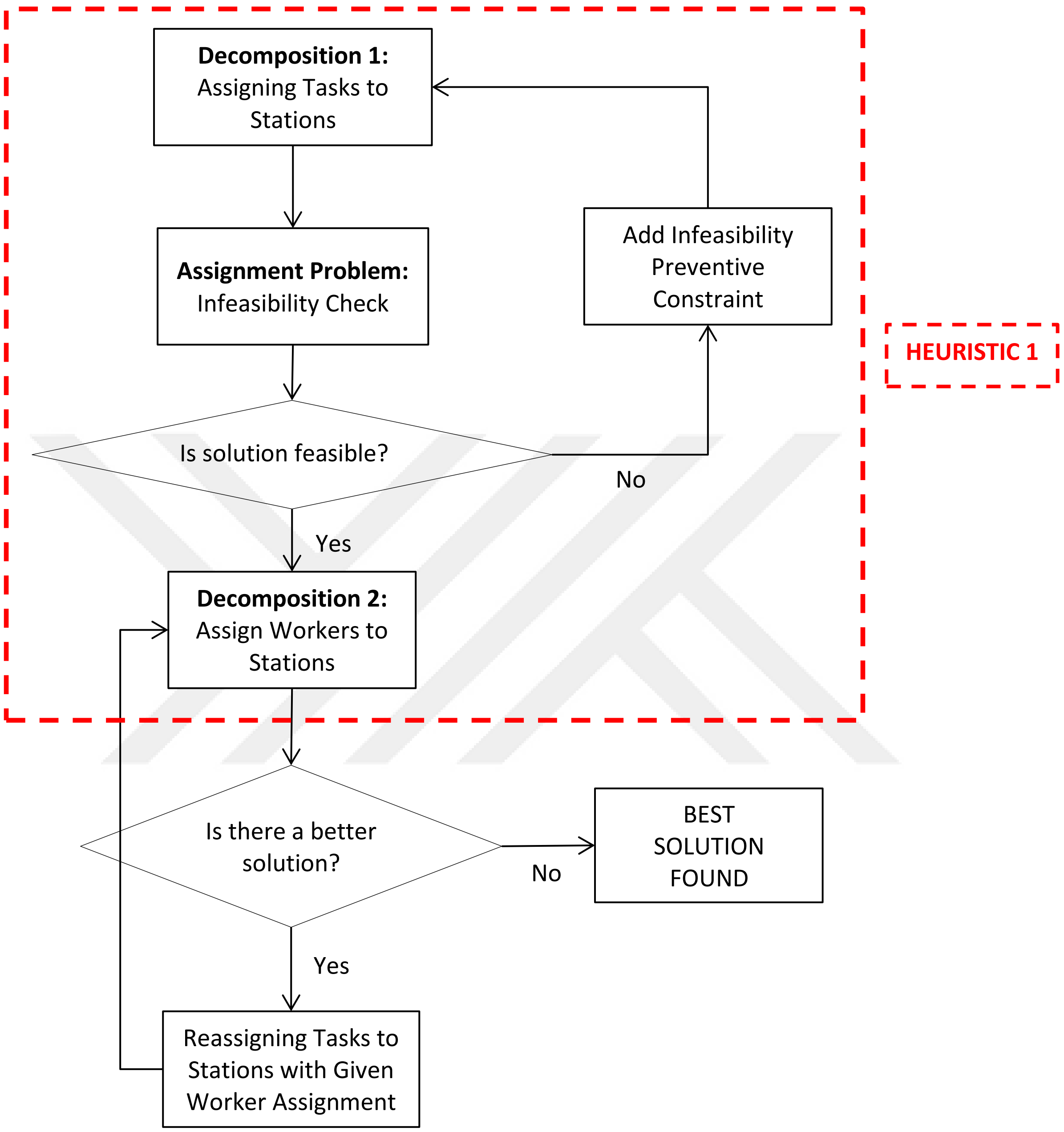

Figure 4: Flow Chart for Heuristic 2

In Heuristic 1 part I tasks are assigned to workstations by disregarding the workers, then workers are assigned to those tasks and workstations. There might be some inefficiencies with the result of Heuristic 1 since workers are disregarded in part I. Heuristic 2 is designed to improve the result of Heuristic 1 ALWABP-2 problem. By using the output of worker -

workstation assignment in Heuristic 1 part II, exact algorithm for ALWABP-2 is run again. The only difference for exact algorithm for ALWABP-2 is worker - workstation assignment is known and input as parameter this time. In this way, tasks are reassigned to workstations with the given worker - workstation assignment. These two algorithms are run iteratively until objective function value no further improves. Pseudo code for Heuristic 2 is given below.

Heuristic 2

1. Solve Heuristic 1: Let Y_{swk} be the optimal solution.
2. **if** objective function of Heuristic 1 part 1 gives better solution compared to Decomposition 2 **then**
3. **input:** Y_{swk}
4. Solve exact algorithm for ALWABP-2 by using Y_{swk} as input. Let X_{srik} be the optimal solution.
5. **input:** X_{srik}
6. Solve Heuristic 1 part 2 model.
7. Solve these two algorithms iteratively until objective function value no further improves.

Heuristic 2 is only valid for ALWABP-2 since reassigning tasks with the given worker assignment does not improve the objective function of ALWABP-1.

4.4. HEURISTIC 3

Workers are assigned to assembly lines from a shared pool in Vestel TV Production. In Heuristic 3, a decomposition algorithm is developed to assign workers to assembly lines. After decision of which operators work in which line is known, this information is input into exact algorithms as constraint and exact algorithms are run to get results.

Table 16: Notation for Heuristic 3 Decomposition Algorithm

WL_{wk}	Binary variable. Equal to one only if worker *w* is assigned to line *k*
LT_{ki}	Continuous variable. Number of worker having competency for task *i* in line *k*
Max_i	Continuous variable. Maximum worker number within all lines having competency for task *i*
Min_i	Continuous variable. Minimum worker number within all lines having competency for task *i*

[MIN] $\sum_i Max_i - Min_i$ (31)

Subject to:

$\sum_k WL_{wk} = 1$ $\forall w \in W$ (32)

$\sum_w WL_{wk} \cdot I_{wi} = LT_{ki}$ $\forall k \in K, \forall i \in N$ (33)

$Min_i \leq LT_{ki}$ $\forall k \in K, \forall i \in N$ (34)

$Max_i \geq LT_{ki}$ $\forall k \in K, \forall i \in N$ (35)

Objective (31) is to minimize the total differences between maximum and minimum worker numbers within all lines having competency for each task. Constraint (32) ensures that each worker must be assigned to a line. Constraint (33) states the number of workers having competency for each task for each line. Constraints (34) and (35) determine the minimum and maximum worker number within all lines having competency for tasks.

A numerical small case will be presented to explain working principle of Heuristic 3 clearly. Let's assume that 9 workers, 6 tasks and 3 lines are included in this small case.

Table 17: Worker-Task Compatibility for Small Case

Worker / Task	1	2	3	4	5	6
1	1	1	1	-	-	-
2	1	1	1	-	-	-
3	1	1	1	-	-	-
4	-	-	1	1	1	-
5	-	-	1	1	1	-
6	-	-	1	1	1	-
7	-	-	-	1	1	1
8	-	-	-	1	1	1
9	-	-	-	1	1	1
Total	**3**	**3**	**6**	**6**	**6**	**3**

Table 17 shows the worker task compatibility matrix for the small case scenario. It shows that there exist 3 workers having competency for task 1, 3 workers for task 2, 6 workers for task 3 and so on. In heuristic 3, our aim is to assign workers to lines in a balanced way. For example, it would be better if there is one worker having competency for task 1 in each line. Similarly, it is better to have one worker for task 2, two workers for task 3 and so on if possible. Therefore, one of the optimal worker – line assignment for this small case study is given in Table 18.

Table 18: Optimal Worker-Line Assignment for Small Case

	Line 1	**Line 2**	**Line 3**
Worker	1,4,7	2,5,8	3,6,9

Table 19: Number of Workers Having Ability to Perform a Task in a Line

LT_{ki}	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Line 1	1	1	2	2	2	1
Line 2	1	1	2	2	2	1
Line 3	1	1	2	2	2	1

Table 20: Minimum and Maximum Number of Workers Having Ability to Perform a Task Among All Lines

Task	Min_i	Max_i	$Max_i - Min_i$
1	1	1	0
2	1	1	0
3	2	2	0
4	2	2	0
5	2	2	0
6	1	1	0

Table 19 shows how many workers can perform each task in each line and Table 20 represents the minimum and maximum number of workers having ability to perform a task among all lines. To illustrate, there are exactly one worker in each line having ability to perform task 1. Therefore, both minimum and maximum values take the value of 1 and difference becomes 0. Difference between minimum and maximum values are 0 for all tasks and it is seen that workers are assigned to lines in a balanced way.

After worker – line assignment is determined by the algorithm, constraint (36) is added to exact algorithm for ALWABP-1 and constraint (37) is added to exact algorithm for ALWABP-2.

$$Y_{swk} \leq WL_{wk} \qquad \forall w \in W, \forall k \in K, \forall s \in S \qquad (36)$$

$$Y_{swk} = WL_{wk} \qquad \forall w \in W, \forall k \in K, \forall s \in S \qquad (37)$$

Constraint (36) and (37) guarantees that workers can be assigned to only specified assembly lines for ALWABP-1 and ALWABP-2, respectively.

5. RESULTS

In this section, results of the exact and heuristic algorithms will be shared. Firstly, different instances will be generated and then computational studies will be made with these instances for each proposed algorithms.

5.1. INSTANCE GENERATION

Proposed models and algorithms will be tested for different instance groups to check their performances. Four different instance groups are determined for the computational study and they are listed in Table 21.

Table 21: Instance Groups and Corresponding Values

Name	Description	Value 1	Value 2	Value 3
A	Percentage of the absent workers	5%	10%	-
B	Percentage of the high level experienced workers	70%	80%	-
C-1	Cycle cime (sec)	22	24	-
C-2	Lower and Upper Bound for Throughput (TV/hr)	150 / 200	180 / 220	-
D	Number of lines	2	7	13

First instance group is the percentage of the absent workers (A) in each shift and two different values are determined for this parameter. The reason is that Vestel works with 5% absenteeism rate on the average in real case. If it is needed to work in some special dates especially in religious holidays, absenteeism rate increases up to 10% and it is analyzed what happens if absent rate increases to that level.

Second instance group is the percentage of the high level experienced workers (B). There are four different levels for the experience of the workers which are low, lower-middle, upper-middle and high level. High level experienced worker rate is about 70% in Vestel in high season. In low season, some of the low level experienced workers are voluntarily made redundant and high level experienced worker level increases to 80%.

Third instance group is cycle time (C) and throughput of the lines. Instances for cycle time will be used for the ALWABP-1 models and instances for throughput will be used for the ALWABP -2 models. Two different values are determined for both cycle time and throughput by considering the low and high season.

Last instance group is the line number (D) for TV production. This instance group is generated to observe the performance of proposed models in different sizes. Normally, 13 production lines are actively used in Vestel, but 13 production lines require, 260 workers, 13 products, 45 tasks and 21 workstations. Therefore, size of the problem increases dramatically for 13 lines and there is a risk to reach good results in a reasonable time. Proposed models will be run for three different line numbers which are 2, 7 and 13 to see the robustness of the algorithms in different sizes.

Totally, there exist 24 instance groups and they are run for two different objective functions. First objective function is to minimize the total number of worker, to be represented as Obj. 1 in Table 22 and the second objective function is to maximize total throughput, to be represented as Obj. 2 in Table 22.

Table 22: Objective Functions and Their Descriptions

Name	Description
Obj. 1	Minimize Total Number of Worker
Obj. 2	Maximize Total Throughput

5.2. COMPUTATIONAL STUDY

Proposed models and algorithms are run for 24 instances and 2 different objectives in CPLEX 12.7 on an i7 – 7700HQ CPU PC with a 2.80 GHz and 16 GB installed memory (RAM). Proposed algorithms are needed to give good solutions within two minutes during the idle time of the workers at the beginning of the shift. Actually, total idle time is 10 minutes at the beginning of the shift, but 3 minutes are reserved for daily data management and data input, 2 minutes are reserved for the running of the proposed algorithms and 5 minutes are reserved for the allocation of the workers to the lines and stations. Thus, output of the proposed models and algorithms will be presented for two minutes in the next sections.

5.2.1. Output of Exact Algorithms

Exact algorithms are run for the 24 instances and two different objective functions with two minutes time limit. Total run time, absolute gap and the objective function values for each instance are listed in Table 23.

Based on Table 23, exact algorithms with Obj. 1 and Obj. 2 provide optimal solutions for almost all instances including two lines. Exact algorithms give solutions with gap values between 6.3 % and 9.5% for Obj. 1 and between 0.4% and 28.7% for Obj. 2 for instances having 7 lines. However, exact algorithms give no solutions for instances including 13 lines and these are represented as N.S. in Table 23. Since real life case includes 13 lines, necessity of the heuristic algorithms are clearly shown.

Table 23: Exact Algorithm Solutions for 2 Minutes Time Limit

	Exact Algorithm (2 min)					
Instances	**Obj. 1**	**Solution Time (sec)**	**Gap (%)**	**Obj. 2**	**Solution Time (sec)**	**Gap (%)**
A1B1C1D1	27	6	-	400	1	-
A1B1C2D1	25	9	-	435	4	-
A1B2C1D1	27	14	-	400	2	-
A1B2C2D1	25	10	-	430	11	-
A2B1C1D1	27	6	-	400	2	-
A2B1C2D1	25	6	-	428	120	0,6
A2B2C1D1	27	20	-	400	4	-
A2B2C2D1	25	6	-	427	120	0,8
A1B1C1D2	99	120	8,4	1393	120	0,4
A1B1C2D2	87	120	6,3	1406	120	9,3
A1B2C1D2	98	120	7,6	1393	120	0,4
A1B2C2D2	89	120	9,5	1357	120	13,1
A2B1C1D2	99	120	8,6	1393	120	0,4
A2B1C2D2	87	120	6,5	1357	120	12,8
A2B2C1D2	98	120	8,1	1340	120	4,4
A2B2C2D2	89	120	9,4	1177	120	28,7
A1B1C1D3	N.S.	120	N.S.	N.S.	120	N.S.
A1B1C2D3	N.S.	120	N.S.	N.S.	120	N.S.
A1B2C1D3	N.S.	120	N.S.	N.S.	120	N.S.
A1B2C2D3	N.S.	120	N.S.	N.S.	120	N.S.
A2B1C1D3	N.S.	120	N.S.	N.S.	120	N.S.
A2B1C2D3	N.S.	120	N.S.	N.S.	120	N.S.
A2B2C1D3	N.S.	120	N.S.	N.S.	120	N.S.
A2B2C2D3	N.S.	120	N.S.	N.S.	120	N.S.

To have an applicable algorithm for Vestel TV production, algorithm is needed to give good solutions in two minutes, but we also analyzed exact algorithms for one hour to test how good is the solutions of our proposed heuristic algorithms. As it is seen in Table 24, objective function values improve in some points, but optimal results cannot be reached for most of the instances having 7 lines. When solutions of the exact algorithms with 2 minutes and 60 minutes are compared, the main difference is observed in instances having 13 lines. Exact algorithms give solution for instances having 13 lines within 60 minutes with some gap values and these solutions will be used to compare with the solutions of the heuristic algorithms.

Table 24: Exact Algorithm Solutions for 60 Minutes Time Limit

	Exact Algorithm (60 min)					
Instances	**Obj. 1**	**Solution Time (sec)**	**Gap (%)**	**Obj. 2**	**Solution Time (sec)**	**Gap (%)**
A1B1C1D1	27	7	-	400	2	-
A1B1C2D1	25	14	-	435	4	-
A1B2C1D1	27	11	-	400	2	-
A1B2C2D1	25	10	-	430	10	-
A2B1C1D1	27	7	-	400	2	-
A2B1C2D1	25	5	-	428	331	-
A2B2C1D1	27	17	-	400	4	-
A2B2C2D1	25	6	-	428	421	-
A1B1C1D2	97	3600	5,9	1393	84	-
A1B1C2D2	86	3600	4,7	1482	3600	3
A1B2C1D2	97	3600	5,6	1393	302	-
A1B2C2D2	86	3600	4,5	1471	3600	4
A2B1C1D2	97	3600	5,8	1393	298	-
A2B1C2D2	86	3600	4,4	1446	3600	5,7
A2B2C1D2	97	3600	5,8	1386	3600	0,9
A2B2C2D2	86	3600	4,6	1416	3600	6,8
A1B1C1D3	180	3600	8	2593	2044	-
A1B1C2D3	162	3600	8	2710	3600	5
A1B2C1D3	179	3600	6,9	2593	3600	0,3
A1B2C2D3	163	3600	6,1	2630	3600	8,1
A2B1C1D3	179	3600	7,8	2593	3600	0,3
A2B1C2D3	162	3600	7	2500	3600	14,1
A2B2C1D3	179	3600	7,9	2540	3600	2,3
A2B2C2D3	162	3600	7	2471	3600	14,4

For the instances having 13 lines, time needed to find first feasible solution is also important and this information is shared in Table 25. Average time to get first feasible solution is 5.5 minutes for Obj. 1 and 3.5 minutes for Obj. 2.

Table 25: Time to Get First Feasible Solution

	Time to Get First Feasible Solution	
Instances	**Obj. 1 (sec)**	**Obj. 2 (sec)**
A1B1C1D3	263	299
A1B1C2D3	222	424
A1B2C1D3	197	162
A1B2C2D3	416	161
A2B1C1D3	154	158
A2B1C2D3	512	161
A2B2C1D3	716	199
A2B2C2D3	168	179
Average	**331**	**218**

5.2.2. Output of Heuristic 1

When Heuristic 1 algorithm is run, solutions in Table 26 are reached. All instances are solved without any gap value within 2 minutes, except A2B2C1D3. Infeasibility case for this instance is different than presented in section 4.2.5. In Obj. 1, algorithm tries to minimize the total station number by increasing the load of the stations near to cycle time in step 1 and workers are not considered in this step. In step 3, workers are tried to be assigned to stations and corresponding tasks, but their experience coefficient is multiplied by the total task time in that station. For A2B2C1D3 instance, cycle time is given as 22 seconds and if total task time in a station is 22 sec, then only workers having high level competence can be assigned to that station since its experience coefficient is 1. Similarly, if total task time in a station is 15 sec, only workers having high level and middle-high level competence can be assigned to that station since their experience coefficents are 1 and 1.25, respectively and cycle time is not exceeded when these workers are assigned to that task 15 (15*1) and 18.75 (15*1.25). In A2B2C1D3 instance, absenteeism number is on the higher number which is 10% and experience level is on the lower number which is 70%. When workers are assigned to stations, cycle time is exceeded in some stations since high and middle-high experience worker number is relatively lower compared to other instances. Infeasibility case is because of this reason and to prevent such infeasilibities, this case will be studied in the future studies.

Table 26: Heuristic 1 Solutions for 2 Minutes Time Limit

	Heuristic 1					
Instances	**Obj. 1**	**Solution Time (sec)**	**Gap (%)**	**Obj. 2**	**Solution Time (sec)**	**Gap (%)**
A1B1C1D1	27	2	-	400	1	-
A1B1C2D1	25	1	-	396	1	-
A1B2C1D1	27	2	-	380	1	-
A1B2C2D1	25	1	-	390	1	-
A2B1C1D1	27	2	-	360	1	-
A2B1C2D1	25	1	-	383	3	-
A2B2C1D1	27	2	-	360	1	-
A2B2C2D1	25	1	-	362	3	-
A1B1C1D2	97	19	-	1389	5	-
A1B1C2D2	86	21	-	1410	25	-
A1B2C1D2	97	21	-	1339	7	-
A1B2C2D2	86	19	-	1363	32	-
A2B1C1D2	97	19	-	1344	7	-
A2B1C2D2	86	17	-	1360	65	-
A2B2C1D2	97	20	-	1293	9	-
A2B2C2D2	86	21		1304	78	-
A1B1C1D3	179	101	-	2584	34	-
A1B1C2D3	162	113	-	2633	82	-
A1B2C1D3	179	102	-	2517	62	-
A1B2C2D3	162	96	-	2618	88	-
A2B1C1D3	179	101	-	2571	23	-
A2B1C2D3	162	108	-	2579	71	-
A2B2C1D3	Infeasible	Infeasible	-	2507	46	-
A2B2C2D3	162	99	-	2524	93	-

When solutions of the Heuristic 1 with 2 minutes time limit are compared to solutions of the exact algorithms with 60 minutes time limit, Heuristic 1 gives same or better solutions for all instances in Obj. 1 within much shorter times. However, Heuristic 1 gives same or better solutions for only 6 instances out of 24 instances in Obj. 2. On the average, difference between objective function values of Heuristic 1 and Exact Algorithm with 60 minutes are only 5% and it means that Heuristic 1 gives averagely 5% worse solutions in Obj. 2.

5.2.3. Output of Heuristic 2

The aim of Heuristic 2 is to improve the solutions of Heuristic 1 - Obj. 2 algorithms and solutions are exhibited in Table 27. Heuristic 2 improves the solutions of Heuristic 1 5% on the average.

Table 27: Heuristic 1 and Heuristic 2 solutions for 2 minutes time limit

	Heuristic 1			Heuristic 2		
Instances	**Obj. 2**	**Solution Time (sec)**	**Gap (%)**	**Obj. 2**	**Solution Time (sec)**	**Gap (%)**
A1B1C1D1	400	1	-	400	2	-
A1B1C2D1	396	1	-	427	5	-
A1B2C1D1	380	1	-	400	3	-
A1B2C2D1	390	1	-	424	4	-
A2B1C1D1	360	1	-	400	7	-
A2B1C2D1	383	3	-	420	5	-
A2B2C1D1	360	1	-	387	3	-
A2B2C2D1	362	3	-	401	4	-
A1B1C1D2	1389	5	-	1393	8	-
A1B1C2D2	1410	25	-	1468	52	-
A1B2C1D2	1339	7	-	1386	19	-
A1B2C2D2	1363	32	-	1460	68	-
A2B1C1D2	1344	7	-	1393	20	-
A2B1C2D2	1360	65	-	1454	113	-
A2B2C1D2	1293	9	-	1373	20	-
A2B2C2D2	1304	78	-	1424	99	-
A1B1C1D3	2584	34	-	2593	49	-
A1B1C2D3	2633	82	-	2757	120	-
A1B2C1D3	2517	62	-	2586	101	-
A1B2C2D3	2618	88	-	2746	116	-
A2B1C1D3	2571	23	-	2593	53	-
A2B1C2D3	2579	71	-	2679	94	-
A2B2C1D3	2507	46	-	2580	115	-
A2B2C2D3	2524	93	-	2661	116	-

Table 28 shows that Heuristic 2 gives close solutions compared to Exact Algorithm with 60 minutes time limit for instances having 2 lines and 7 lines. However, solutions of Heuristic 2 are better compared to Exact Algorithm with 60 minutes time limit for 13 lines and it shows that Heuristic 2 is robust enough.

Table 28: Comparison between Solutions of Heuristic 2, Exact Algorithms and Heuristic 1

	Exact Algorithm (2 min)	Exact Algorithm (60 min)	Heuristic 1	Heuristic 2
Instances	**Obj. 2**	**Obj. 2**	**Obj. 2**	**Obj. 2**
A1B1C1D1	400	400	400	400
A1B1C2D1	435	435	396	427
A1B2C1D1	400	400	380	400
A1B2C2D1	430	430	390	424
A2B1C1D1	400	400	360	400
A2B1C2D1	428	428	383	420
A2B2C1D1	400	400	360	387
A2B2C2D1	427	428	362	401
A1B1C1D2	1393	1393	1389	1393
A1B1C2D2	1406	1482	1410	1468
A1B2C1D2	1393	1393	1339	1386
A1B2C2D2	1357	1471	1363	1460
A2B1C1D2	1393	1393	1344	1393
A2B1C2D2	1357	1446	1360	1454
A2B2C1D2	1340	1386	1293	1373
A2B2C2D2	1177	1416	1304	1424
A1B1C1D3	N.S.	2593	2584	2593
A1B1C2D3	N.S.	2710	2633	2757
A1B2C1D3	N.S.	2593	2517	2586
A1B2C2D3	N.S.	2630	2618	2746
A2B1C1D3	N.S.	2593	2571	2593
A2B1C2D3	N.S.	2500	2579	2679
A2B2C1D3	N.S.	2540	2507	2580
A2B2C2D3	N.S.	2471	2524	2661

5.2.4. Output of Heuristic 3

Table 29 shows the solutions of Heuristic 3 for Obj. 1 and Obj. 2 and almost all instances are solved without any gap value within 2 minutes. When solutions of the Heuristic 3 with 2 minutes time limit is compared to Exact Algorithm with 60 minutes time limit, Heuristic 3 is worse only %2 for Obj. 1 and worse only %1 for Obj. 2. This results show that Heuristic 3 is also robust enough to solve the problems in Vestel TV production.

Table 29: Heuristic 3 Solutions with 2 Minutes Time Limit

	Heuristic 3					
Instances	**Obj. 1**	**Solution Time (sec)**	**Gap (%)**	**Obj. 2**	**Solution Time (sec)**	**Gap (%)**
A1B1C1D1	28	4	-	400	1	-
A1B1C2D1	25	2	-	430	2	-
A1B2C1D1	28	3	-	400	2	-
A1B2C2D1	25	3	-	423	5	-
A2B1C1D1	28	2	-	393	2	-
A2B1C2D1	25	2	-	416	4	-
A2B2C1D1	28	4	-	386	3	-
A2B2C2D1	25	2	-	405	5	-
A1B1C1D2	97	15	-	1386	12	-
A1B1C2D2	87	15	-	1464	19	-
A1B2C1D2	99	15	-	1380	15	-
A1B2C2D2	88	14	-	1445	21	-
A2B1C1D2	Infeasible	Infeasible	-	1340	120	1,5
A2B1C2D2	89	120	7,3	1398	120	6
A2B2C1D2	Infeasible	Infeasible	-	1333	120	1,9
A2B2C2D2	Infeasible	Infeasible	-	1378	120	7,4
A1B1C1D3	181	54	-	2553	39	-
A1B1C2D3	166	89	-	2682	48	-
A1B2C1D3	184	67	-	2553	42	-
A1B2C2D3	166	45	-	2669	111	-
A2B1C1D3	183	61	-	2540	45	-
A2B1C2D3	166	55	-	2655	92	-
A2B2C1D3	186	120	5,2	2506	46	-
A2B2C2D3	168	50	-	2614	120	2,8

Heuristic 3 gives infeasible solutions in Obj. 1 models for 3 instances having 7 lines which are A2B1C1D2, A2B2C1D2 and A2B2C2D2. The reason for infeasibility is similar to realized in Heuristic 1. In Heuristic 3, workers first assigned to lines and then lines are balanced to optimize the objective function. Experience of the workers are not taken into consideration in the algorithm assigning workers to lines, so number of middle-low level and low level experienced workers might higher in some lines. In this case, it is not possible to balance the line without exceed the cycle time. This infeasibility cases might be studied in future researches.

5.2.5. Line Balance for Exact and Heuristic Algorithms

In this section, line balance of the exact and heuristic algorithms will be analyzed. Line balance is observed as about 70% in real life and it is expected to be improved by proposed algorithms. Figure 5 shows a whisker and box plot representing line balance of the proposed algorithms for instances A1B1C1D3 and A2B1C2D3. In this plot output of the Exact Algorithm with 60 minutes time limit and Heuristic 1, 2 and 3 with 2 minutes limit are used for specified instances. These instances have 13 lines and to create the whisker and box plot, line balance of each line is needed to be calculated first. After line balances are calculated for 13 lines, variation of line balances among 13 lines are used to create the plot in Figure 5.

Line balance for Exact Algorithms (E.A.), Heuristic 1 (H1), Heuristic 2 (H2) and Heuristic 3 (H3) is shown in Figure 5. Line balance for proposed algorithms mostly varies from 82% to 95% and on the average it is about 90%. Results are much better compared to real life line balance values. The worst line balance value is observed in Exact Algorithm for Obj. 2 in A2B1C2D3 instance. When it is checked from Table 24, obtained solution is far away from the optimal solution and relative gap is about 14%.

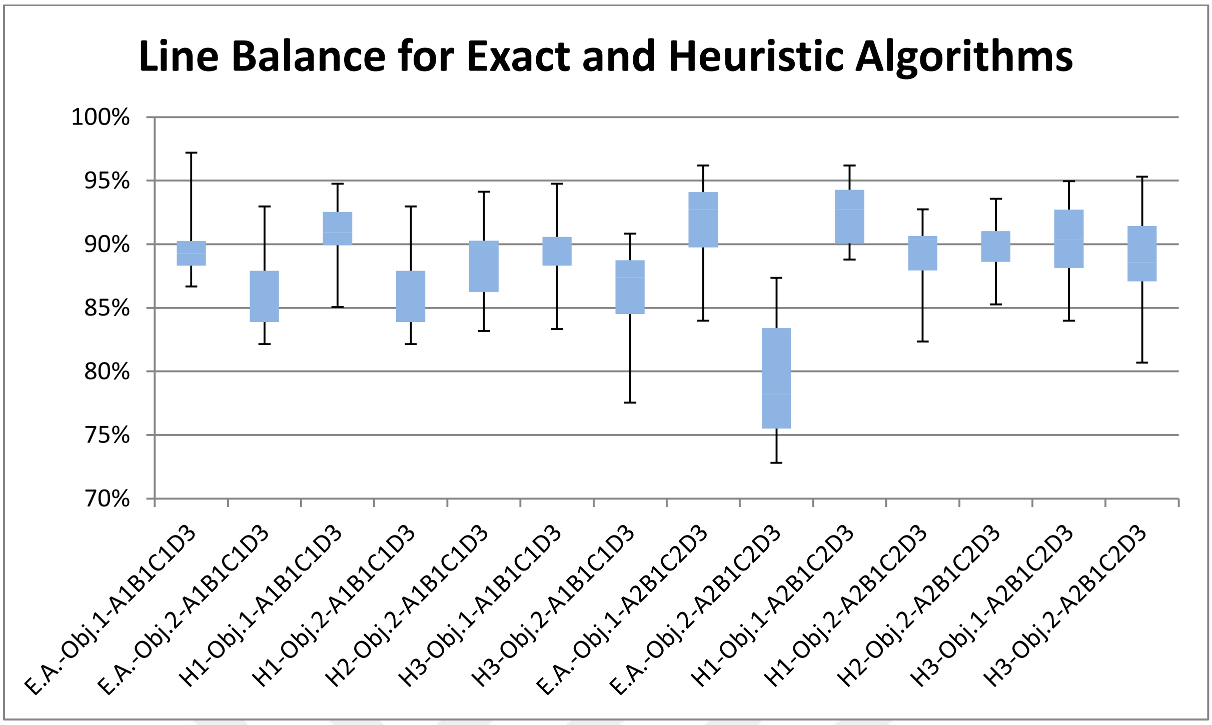

Figure 5: Line Balance for Exact and Heuristic Algorithms

5.2.6. Comparison for Experience Level of the Workers Used by Proposed Algorithms

In Obj.1, exact algorithms and proposed algorithms tries to minimize the number of worker and it is expected that algorithms prefer to use high experience workers first since they perform the same task in a shorter time than workers having other experience levels. To observe this case, output of the instances having 13 lines are analyzed for proposed algorithms and it is shown in Figure 6. In Obj. 2, all available workers are used in stations, hence only Obj. 1 will be analyzed for this case. Since Heuristic 2 is designed for only Obj. 2, it is not included in Figure 6.

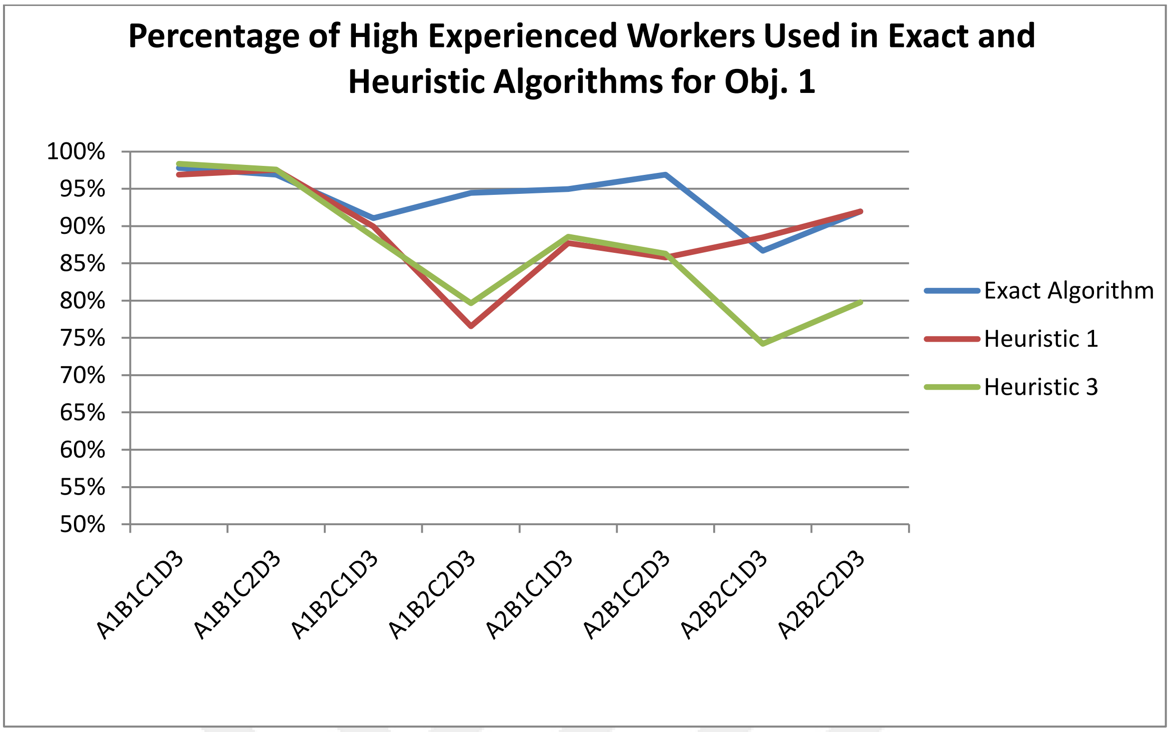

Figure 6: Percentage of High Experienced Workers Used in Exact and Heuristic Algorithms for Obj. 1

It is seen from Figure 6 that workers with high experienced level usage is higher in exact algorithm which is 94%. The reason might be that exact algorithm considers both task assignment and worker assignment at the same time; therefore, it gives the priority to high experienced worker directly to minimize the total number of worker used. However, in Heuristic 1, since problem is decomposed into sub-problems, first task assignment is done without considering worker competency and then worker assignment is done to corresponding tasks. After first sub-problem, total task time is determined for each station and when next sub-problem is run, it is not a problem to use high or low level experienced worker for a station except exceeding the cycle time. To illustrate, assume that total task time is 10 seconds in a station and cycle time is 20 seconds. In this case, when sub-problem assigning workers to stations is run, even low level experienced worker can be used in this station since its experience coefficient is 2 and cycle time cannot be exceeded (10x2 = 20 sec). That’s why percentage of high experienced workers used is lower in Heuristic 1 which is 89% compared to exact algorithm. In Heuristic 3, first workers are assigned to lines and then each line is balanced internally with the given worker set. In this case, experience level of workers is not considered during worker - lines assignment, so different cases might be realized. For instance, high experienced worker assigned to might be 95% in line 1 whereas 70% in line 2. In this case, while balancing line 2 with the given cycle time, workers are needed to be used other than high level experience. This is the reason why Heuristic 3 has the lowest high experienced worker usage ratio, 87%, among other algorithms.

6. CONCLUSION

This study is conducted on the real life problem of TV production of Vestel Electronics Company. Workers are needed to be assigned to lines and lines are needed to be balanced by considering absenteeism and competency of the workers within 2 minutes time limit at the beginning of the shift. This is needed to be solved. This problem is called as ALWABP in literature and two types of ALWABP are handled which are ALWABP-1 and ALWABP-2 by considering the seasonality in demand. In ALWABP-1, objective is to minimize total number of worker and in ALWABP-2, objective is to maximize total throughput. Exact algorithms are developed to solve the real life problem. Some nonlinear cases are observed in ALWABP-2 of exact algorithm and it is handled by using McCormick Relaxation method. Exact algorithms give good solutions for small instances, but they give no solutions for real life cases. Then, three heuristic algorithms are developed to handle this situation. In first heuristic, master problem is decomposed into two sub-problems and a multi-pass hierarchical solution frame work is constructed. Feasibility of the first sub-problem is checked by a novel assignment model and a cut is added to first sub-problem in case of infeasibility. This continues until feasibility is attained. After feasibility is guaranteed, second sub-problem is solved to obtain final solution for the master problem. In Heuristic 2, solution of first heuristic is aimed to be improved for ALWABP-2 type. Once second sub-problem in Heuristic 1 is solved, tasks are reassigned to workstations with the given worker assignment and algorithms run iteratively until no more improvements on objective value is reached. In Heuristic 3, an algorithm is developed to assign workers to lines in a balanced way by considering task abilities of the workers. Then, exact algorithms are run with the input of worker-line assignment. Computation results show that all heuristic approaches provide good enough solutions for the real life problem within allowed time limit. Heuristic approaches are encouraging Vestel Electronics' production team to implement the proposed approach as an online decision support system.

7. REFERENCES

A, S. (2006) ‘A survey on problems and methods in generalised assembly line balancing’, *European Journal of Operations Research*, 168, p. 694.

A, S. (2007) ‘A classification of assembly line balancing problems’, *European Journal of Operations Research*, 183, p. 674.

Akyol, S. D. and Baykaso\uglu, A. (2016) ‘A multiple-rule based constructive randomized search algorithm for solving assembly line worker assignment and balancing problem’, *Journal of Intelligent Manufacturing*. doi: 10.1007/s10845-016-1262-6.

Araújo, F. F. B., Costa, A. M. and Miralles, C. (2013) ‘Balancing parallel assembly lines with disabled workers’, *CoRR*, abs/1304.1. Available at: http://arxiv.org/abs/1304.1423.

Becker, C. and Scholl, A. (2006) ‘A survey on problems and methods in generalized assembly line balancing’, *European Journal of Operational Research*, 168(3), pp. 694–715. doi: 10.1016/j.ejor.2004.07.023.

Blum, C., Miralles, C. and C., M. (2011) ‘On Solving the Assembly Line Worker Assignment and Balancing Problem via Beam Search’, *Computers & Operations Research*, 38(1), p. 328. doi: 10.1016/j.cor.2010.05.008.

Borba, L. and Ritt, M. (2013) ‘Exact and Heuristic Methods for the Assembly Line Worker Assignment and Balancing Problem’, *CoRR*, abs/1308.0. Available at: http://arxiv.org/abs/1308.0299.

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

C., M. *et al.* (2012) ‘Two Extensions for the ALWABP: Parallel Stations and Collaborative Approach’, *International Journal of Production Economics*, 140(1), pp. 483–495. doi: https://doi.org/10.1016/j.ijpe.2012.06.032.

Demirkol Akyol, S. *et al.* (2016) ‘ErgoALWABP: a multiple-rule based constructive randomized search algorithm for solving assembly line worker assignment and balancing problem under ergonomic risk factors’, *Journal of Intelligent Manufacturing*. doi: 10.1007/s10845-016-1262-6.

J., P., Vilà, M. and Pereira, J. (2014) 'A Branch-and-bound Algorithm for Assembly Line Worker Assignment and Balancing Problems', *Computers & Operations Research*, 44(Supplement C), pp. 105–114. doi: https://doi.org/10.1016/j.cor.2013.10.016.

Khemyong, P. and Sirovetnukul, R. (2013) 'Task and worker assignment in the shared-machine U-shaped assembly line', in *2013 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 325–329. doi: 10.1109/IEEM.2013.6962427.

M., C. *et al.* (2007) 'Advantages of Assembly Lines in Sheltered Work Centres for Disabled. A Case Study', *International Journal of Production Economics*, 110(1), p. 187. doi: 10.1016/j.ijpe.2007.02.023.

M., C. *et al.* (2008) 'Branch and Bound Procedures for Solving the Assembly Line Worker Assignment and Balancing Problem: Application to Sheltered Work Centres for Disabled', *Discrete Applied Mathematics*, 156(3), p. 352. doi: 10.1016/j.dam.2005.12.012.

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

Moreira, M. C. O. *et al.* (2010) 'Simple heuristics for the assembly line worker assignment and balancing problem', *CoRR*, abs/1003.3. Available at: http://arxiv.org/abs/1003.3676.

Moreira, M. C. O. *et al.* (2015) 'Robust assembly line balancing with heterogeneous workers', *Computers & Industrial Engineering*, 88(Supplement C), pp. 254–263. doi: https://doi.org/10.1016/j.cie.2015.07.004.

Mutlu, Ö. *et al.* (2013) 'An Iterative Genetic Algorithm for the Assembly Line Worker Assignment and Balancing Problem of Type-II', *Computers & Operations Research*, 40(1), p. 418. doi: 10.1016/j.cor.2012.07.010.

Polat, O. *et al.* (2016) 'A two-phase variable neighbourhood search algorithm for assembly line worker assignment and balancing problem type-II: an industrial case study', *International Journal of Production Research*. Taylor & Francis, 54(3), pp. 722–741. doi: 10.1080/00207543.2015.1055344.

Ramezanian, R. and Ezzatpanah, A. (2015) 'Modeling and solving multi-objective mixed-model assembly line balancing and worker assignment problem', *Computers & Industrial*

Engineering, 87(Supplement C), pp. 74–80. doi: https://doi.org/10.1016/j.cie.2015.04.017.

Ritt, M., Costa, A. and Cristóbal, M. (2015) 'The assembly line worker assignment and balancing problem with stochastic worker availability', *International Journal of Production Research*, 54, pp. 1–16.

Scholl, A., Becker, C. and C., B. (2006) 'State-of-the-art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing', *European Journal of Operational Research*, 168(3), p. 666. doi: 10.1016/j.ejor.2004.07.022.

Zacharia, P. T. and Nearchou, A. C. (2016) 'A population-based algorithm for the bi-objective assembly line worker assignment and balancing problem', *Engineering Applications of Artificial Intelligence*, 49(Supplement C), pp. 1–9. doi: https://doi.org/10.1016/j.engappai.2015.11.007.

Zaman, T., Paul, S. K. and Azeem, A. (2012) 'Sustainable operator assignment in an assembly line using genetic algorithm', *International Journal of Production Research*. Taylor & Francis, 50(18), pp. 5077–5084. doi: 10.1080/00207543.2011.636764.

8. VITA

Muhammet Fatih Çengil received the bachelor's degree in Industrial Engineering Department of Middle East Technical University, Ankara, Turkey in 2014. After graduation, he started to work in Vestel Electronics Company as a Process Development Engineer and now he is working as TV Production Specialist in TV Production Department. He started to Master of Science Program in Industrial Engineering, Department of Özyeğin University in 2014. He conducted his M.Sc. study under supervision of Asst. Prof. Görkem Yılmaz and Asst. Prof. Erinç Albey.