A MULTIOBJECTIVE APPROACH TO ASSEMBLY LINE PART FEEDING PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

RAMAZAN KIZILYILDIRIM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

DECEMBER 2018

Approval of the thesis

A MULTIOBJECTIVE APPROACH TO ASSEMBLY LINE PART FEEDING PROBLEM

submitted by **RAMAZAN KIZILYILDIRIM** in partial fulfilment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Yasemin Serin ____________________
Head of Department, **Industrial Engineering Dept., METU**

Prof. Dr. Esra Karasakal ____________________
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. Seçil Savaşaneril Tüfekci (Head of the com.) ____________________
Industrial Engineering Dept., METU

Prof. Dr. Esra Karasakal (Supervisor) ____________________
Industrial Engineering Dept., METU

Assoc. Prof. Dr. Serhan Duran ____________________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Mustafa Kemal Tural ____________________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Diclehan Tezcaner Öztürk ____________________
Industrial Engineering Dept., Hacettepe University

Date: 07.12.2018

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

Name Last name: Ramazan Kızılyıldırım

Signature:

ABSTRACT

A MULTIOBJECTIVE APPROACH TO ASSEMBLY LINE PART FEEDING PROBLEM

Kızılyıldırım, Ramazan
M.S., Department of Industrial Engineering
Supervisor: Prof. Dr. Esra Karasakal

December 2018, 67 pages

The change in product diversity and sense of quality to increase customer satisfaction has also affected the design and management of assembly lines. Keeping the amount of stock at desired levels to provide parts to the assembly line and prevent the accumulation of stocks at assembly line has become an increasingly challenging problem for manufacturing companies that have high product diversity and high model variability. Increasing the number of vehicles might be a solution to control stock level but this leads to in-plant traffic problem. Therefore, a balance should be maintained between stock and traffic level to overcome this challenge. Although there is a comprehensive literature on assembly line optimization, assembly line part feeding problem has been addressed by relatively few studies.

In this thesis, we aim to minimize the number of tours needed to feed the assembly line and the line side stock of a company which has many production lines and produces a wide variety of products. A bi-objective mathematical model is developed to produce a transportation schedule for each vehicle driver. Since, the mathematical model is solved optimally for only small-size problems, a problem-specific heuristic algorithm is developed to solve large-size problems. The heuristic algorithm generates solutions within a very short time period for large-size problems. The heuristic algorithm is capable to produce almost all nondominated solutions.

Keywords: Assembly Line Part Feeding, Multiobjective Optimization

ÖZ

MONTAJ HATTI PARÇA BESLEME PROBLEMİ İÇİN ÇOK AMAÇLI BİR YAKLAŞIM

Kızılyıldırım, Ramazan
Yüksek Lisans Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Esra Karasakal

Aralık 2018, 67 sayfa

Müşteri memnuniyetini arttırmak için ürün çeşitliliği ve kalite anlayışında yaşanan değişim, üretim süreçlerini de etkilemiştir. Yüksek ürün çeşitliliğine ve üretim hattında yüksek model değişkenliğine sahip üretim firmaları için hat yanı stok miktarını arzu edilen seviyelerde tutarak hattı beslemek ve ara madde yığılmalarını önlemek giderek zorlaşan bir problem halini almıştır. Bu sorunu çözmek için araç sayısını arttırmak firma içi trafik problemine neden olduğu için iki amaç arasında bir denge kurulması gerekliliği ortaya çıkmaktadır. Üretim süreçlerine ilişkin kapsamlı bir literatür olmasına rağmen montaj hattı parça besleme problemi az sayıda çalışma tarafından ele alınmıştır.

Bu tezde çok sayıda üretim hattına sahip ve çok çeşitli ürün üreten bir firma için hat yanı stok miktarının ve hattı beslemek için gerekli araç tur sayısının en azlanması amaçlanmıştır. Her bir araç sürücüsü için taşıma çizelgesi oluşturmayı hedefleyen iki amaçlı bir matematiksel model geliştirilmiştir. Matematiksel model sadece çok küçük ölçekli problemleri çözebilmektedir. Bu nedenle, problem spesifik bir sezgisel algoritma geliştirilmiştir. Sezgisel algoritma büyük problemler için dahi çok kısa sürede domine edilmemiş çözümlerin tamamına yakınını üretmiştir.

Anahtar Kelimeler: Montaj Hattı Parça Besleme, Çok Amaçlı Optimizasyon

To My Family

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my academic supervisor Prof. Dr. Esra Karasakal for her guidance and criticism throughout the study.

Besides my advisors, I wish to express my appreciation to the rest of my thesis committee for their valuable comments and reviews.

Last but not least, I am thankful to my family for their support during my whole life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With increasing product variety, the complexity of the preventive maintenance problem of assembly line has increased. Contemporary part assembly plants use high variant mixed model assembly lines for manufacturing to respond to the increasing variety. On the other hand, this complicates the part feeding policy and requires a relatively large storage space at the assembly line or an increase in the frequency of in-plant traffic. In order to address these problems, different line feeding policies such as “kitting”, “Kanban-based” and “hybrid” have been developed besides the conventional “line siding” policy (Kilic and Durmusoglu 2015).

In line siding policy, parts are being delivered from a central warehouse to line in baskets at predefined time slots. This strategy is easy to execute and does not need any extra material handling before the delivery. On the other hand, a significant amount of parts is carried in the same basket to avoid any shortage. When variety is high, the total amount of the stock at the assembly line is also relatively high (Bozer and McGinnis 1992).

In part feeding system, bringing a group of parts together in predetermined quantities and then delivering them to the line is called kitting (Carlsson and Hensvold 2008). In kitting, only specific parts, which are needed for future time slots, are handled and delivered to the stations (Sali et al. 2015).

In Kanban based part feeding system, every basket is associated with a kanban card which contains all related information about parts (Sendil Kumar and Panneerselvam 2007). Stations are refilled, according to consumption, by parts that are pulled via Kanban cards (Facio et al.2013).

These three policies aim to deliver required parts to the assembly line with minimum associated cost. For some production systems, the hybrid combination of these three approaches could suit best for the production system. Besides decreasing related costs, creating a conducive working environment is also one of the objectives of these feeding systems. Previous studies mostly focus on the macro level comparison of different feeding policies. These macro level studies mostly try to find the best hybrid solution based on long run calculations. At the micro level, there is a need to further enlighten different aspects of each policy separately. In this study, we develop an approach to improve the line siding feeding policy.

As stated in many other studies, the assembly line part feeding problem is NP-hard (Fathi et al. 2014). Decreasing the stock to its minimum level requires high in-plant traffic or decreasing the traffic ends up with a high level of side stock. Both objectives are conflicting, and a tradeoff exists between them. In order to optimize these two objectives, transfer vehicles' loading problem and delivery schedule problem should be solved, simultaneously.

We develop a multi-objective mathematical model that minimizes total stock at the assembly line and the total number of tours required to feed the assembly line. We use epsilon constrained method to generate all nondominated solutions of the exact method. We observe that the exact method can be only used for small-size problems. In this thesis, we also develop a problem specific heuristic algorithm to solve large-size problems efficiently. Computational experiments show that heuristic algorithm generate almost all nondominated solutions. Solution time of the heuristic algorithm is less than ten seconds for tested data sets. We create a very large random dataset to challenge the solution time of the heuristic algorithm and results validate the time efficiency of the algorithm.

In the next section we want to give some important definitions related to multi-objective optimization problems.

1.1. Definitions and Some Theory for Multiobjective Optimization

Efficient, nondominated, weakly efficient and weakly nondominated points

Let x be the decision vector, X be the feasible decision region of a multi-objective problem. Let also the point $f=(f_1(x), f_2(x),..., f_p(x))$ be the corresponding point in objective space F (image of *X*) for the decision vector x where p is the number of the objectives and $f_i(x)$ is the value of the i^{th} objective.

Furthermore, assume that the original problem is as presented below.

$$"min" \qquad (f_1(x), f_2(x), f_3(x),..., f_p(x))$$

Subject to:

$$x \in X$$

For a minimization problem, the solution x is called **efficient** if there is no $x' \in X$ such that

i) $f_k(x') \leq f_k(x)$ for $k = 1,..., p$

ii) $f_k(x') < f_k(x)$ for at least one k.

If there is such an x', then x is called **inefficient**. If x is efficient, then $f=f(x)$ is called **nondominated** point. Whenever x is inefficient the corresponding point in objective space called **dominated**.

A feasible solution $x \in X$ is called weakly efficient if there is no $x' \in X$ such that

$$f_k(x') < f_k(x) \text{ for all } k = 1,..., p.$$

Then, the point $f=f(x)$ is called weakly nondominated.

The rest of the paper is structured as follows. In Section 2, a brief literature review is provided and original contribution of this thesis to literature is given. In Section 3, a detailed description of the problem is provided. In section 4, the proposed methods are presented, and the computational results for both exact and heuristic algorithms are reported in Section 5. Finally, conclusions and future research direction are discussed in Section 6.

CHAPTER 2

LITERATURE REVIEW

Digitalization of manufacturing has dramatically transformed the conventional production approaches to more automated structures. Recent changes in technology and customers' diverse product demands inevitably push companies to employ a high-variant mixed-model production line for most of the product segments. Despite the advantages, such an assembly line has high complexities due to the number of parts needed to be transferred from warehouses to the assembly line.

Early studies regarding the mass production focused on assembly line balancing and operational sequencing to improve the efficiency of the production line. A comprehensive literature review of assembly line balancing is provided by Sivasankaran and Shahabudeen (2014). As newly developed approaches have improved the efficiency of these production lines, the complexity of feeding these lines has increased proportionally.

Problems raised after changes applied to assembly line has been addressed by many studies that aim to improve different aspects of production systems such as line balancing, operational sequencing and facility layout. On the other hand, according to many recent studies, part feeding at assembly line has been addressed by relatively fewer studies (Fathi et al. 2014).

Various new feeding policies are developed in response to the increasing complexity due to the increasing number of parts needed to be transferred from warehouses to the assembly line. In addition to traditional line stocking feeding approach, policies such as Kanban and kitting are developed to reduce inventory at the assembly line (Kilic and Durmusoglu 2014).

In this thesis, we made a classification based on the classification scheme provided by Kilic and Durmusoglu (2014). We classify studies into three classes namely line stocking, Kanban and kitting. Furthermore, we classify studies that mainly compare different part feeding policies as a separate group and determine under which conditions each policy performs better than others.

2.1. Line stocking

Line stocking is mainly executed as transferring parts from a central warehouse to the storage area of the relevant workstation. In this approach, required components are being carried as an entire unit load for each component from the warehouse to the designated station. When a specific part is no longer needed, it is transferred to the warehouse again, and if the unit load is empty or not necessary in future, then it is removed from the station in order to open space at that station. (Zammori et al 2015). In this approach, the continuous use of the material is enabled but in case of high variability, there will be a high amount of stock besides the assembly line (Corakci 2008). Transfer vehicles can be different based on packaging types. For unit loads, pallets are used for transportation, and for small boxes, tugger trains will make periodical tours on a certain path. In the first case, the replenishment target is the reorder level and in the second case two bin policy is used for replenishment (Limere et al. 2012).

Salameh and Gattas (2001) formulated a model to find the optimum JIT inventory stock level by minimizing the sum of the holding cost and the shortage cost. Their key concern is to prevent any loss caused by the production interruption.

Choi and Lee (2002) compared static and dynamic part feeding policies for an automotive company. According to their definition, static part feeding is to supply parts based on predefined demand needs. In their dynamic part feeding approach, they forecast the demand based on the consumption. This approach enables the feeders to swiftly adapt in case of changes in production plans. Their single objective is to put a penalty on late and early deliveries and minimize the total penalty. Computational experiments show that their dynamic approach produces a better solution than the conventional static approach.

Wänström and Medbo (2008) studied the design of component racks and choice of packaging types to improve the efficiency of part feeding systems. They showed that the feeding process could be improved by changing parameters for racks and package types.

Souza et al. (2008) addressed the packing problem to minimize stock beside the assembly line and the frequency of feeders. They formulated a single objective mathematical model and developed a Greedy Randomized Adaptive Search Procedure (GRASP) to solve the NP-Hard problem. Their GRASP method managed to decrease operational costs more than 50%.

Cunha and Souza (2008) modified the procedure developed by Souza et al. (2008). The modified version produces tighter bounds than the previous version of the procedure.

Golz et al. (2011) developed a heuristic solution for in-house transportation. They decompose the entire process into two main parts. In the first part, they produce the demand based on the daily production plan. In the second step, they merge the tours where it is possible under the capacity constraints. After making all possible merges, they assign the tours with an aim to minimize the number of required drivers.

Alnahhal M. and Noche B. (2013) used a mathematical model, analytic equations and a dynamic programming to address the line-side inventory, the number of trains and the variability in loading problems simultaneously. They first determine a period length and then assign feeders to provide parts to the same stations at each period during the shift. The most important objective is to minimize the number of trains and the number of trains is determined by using dynamic programming. Then, they solve a mixed integer mathematical model to minimize the total stock at and the maximum stock at the assembly line. They also keep route length close to average route length to decrease the variability.

Rao et al. (2013) developed a mathematical model and a GASA (genetic algorithm and simulated annealing) heuristic in order to find schedules for a single vehicle that feeds mixed-model assembly lines with minimum total travel cost. They use a backward tracking approach to minimize interaction between formerly scheduled materials and appending materials.

Zhou and Peng (2017) developed a mathematical model to minimize the maximum weighted inventory level in all stations and production cycles during the planning horizon. They also proposed a backtracking algorithm which yields the exact solution for small-scale instances. They developed a modified discrete artificial bee colony (MDABC) metaheuristic for real life instances.

Zhou and Xu (2018) developed two mathematical models to minimize the number of operators and the unit delivery cost for line integrated supermarkets. A dynamic programming is presented to find global optimum for small-size problems and a search algorithm for large-size problems. They first determine the number of logistic operators and stations assigned to operators. Next, they determine cyclic delivery schedules and operators.

2.2. Kitting

Kitting is the gathering of parts needed for the manufacture of a particular product then delivering this compact package to the assembly line (Zammori et al 2015). Kitting works well when the ERP data is robust, but it could fail because of weak ERP system structure. Many of the studies on this subject focus on how to optimize the kitting process. For a review see Kilic and Durmusoglu (2014).

Günther et al. (1996) developed a mixed integer programming model and a heuristic algorithm to address three aspects of kitting problem. They investigated how to arrange a right mix of components to be supplied to each station, assign jobs to stations and determine a minimum number of operators. The mathematical model fails to produce optimal solutions in reasonable time periods. Their heuristic algorithm produces optimal solutions for most of the tested datasets with a very short run time.

Chen and Wilhelm (1997) developed a linear programming model and a heuristic approach to feed the assembly line under the kitting policy. The objective was to minimize the total cost. The cost includes job earliness, job tardiness, and in-process holding costs. Their algorithm gave priority to the parts with the earliest due date. After kits are determined, the algorithm shifts the starting times to avoid earliness.

Carlsson and Hensvold (2008) found that kitting is more beneficial for high variant assembly lines based on a real-life study. Their study also took qualitative assessments into account in addition to quantitative results.

Battani et al. (2010) developed a procedure to decide on centralization and decentralization of component warehouses. The procedure was formulated as a step-by-step process based on successive linear programming optimizations to determine whether centralization or decentralization was favorable for relevant components.

Kilic and Durmusoglu (2012) developed a single objective mathematical model where the objective is to minimize the cost consisting of WIP and the number of workers for design of a kitting system.

Limère et al. (2015) developed a model to choose between kitting and line stocking by considering the walking distance of the operator. Their mathematical model also demonstrated how specific characteristics of a part influence the chances of a part being kitted.

2.3. Kanban

Kanban is developed for decentralized warehouse systems which enables delivery of parts to assembly line frequently in a short time. Kanban method aims to reduce WIP and to shorten the long travel distance to deliver parts from the central warehouse to the assembly line (SendilKumar and Panneerselvam 2007).

Shahabudeen et al. (2002) studied single card kanban system and determined the number of kanbans and the lot size. They formulated a bi-criteria objective function to maximize throughput and minimize aggregate Kanban queue. They used simulated annealing technique to solve the problem.

Jerald et al. (2006) developed an optimization technique called adaptive genetic algorithm. The algorithm has two objectives: i) minimizing the penalty cost for not meeting the delivery and ii) minimizing machine idle time. Their adaptive algorithm produces better results than the genetic algorithm.

Shahabudeen and Sivakumar (2008) developed a genetic algorithm and simulated annealing (SA) based search methods to minimize the inventory and backorder demand. Instead of traditional Kanban system with a fixed number of Kanban cards, they used an adaptive Kanban system. Their results showed that SA based algorithm yields better solutions with large reductions in CPU times.

Emde et al. (2012a) investigated loading of tow trains from a supermarket area with an objective to minimize inventory near the assembly line. In their problem, tow trains follow predefined paths and the demand of each station is determined based on the time required for the next visit. After deriving the demand, tow trains are loaded. They developed an exact polynomial-time algorithm for their problem.

Emde and Boysen (2012b) developed a mathematical model and an exact dynamic model which determines the optimal number of decentralized supermarkets. The algorithm also determines the best location for each supermarket that minimizes associated travel cost.

Emde and Boysen (2012c) developed a mathematical model and a polynomial time exact dynamic model to solve vehicle routing and scheduling problems simultaneously. Their objective is to minimize the total travel cost and the stock level at each station.

Faccio et al. (2013a) proposed a general framework for problems dealing with Kanban and supermarket systems. They used a static and a dynamic approach separately. They determined which factors have significant impacts on the performance of the feeding policy.

Faccio et al. (2013b) minimized the total cost function composed of inventory costs, handling costs, and stock-out costs for feeding a multiple mixed-model assembly-line system.

Fathi et al. (2014) developed a multi-objective mathematical model and a heuristic algorithm. The objectives are to minimize inventory at each station and to minimize the number of tours required to feed the assembly line. In their study the transfer vehicle uses a predefined path to deliver parts. This structure enables the predetermination of exact demand of each station at the arrival of the vehicle.

Lolli et al. (2015) used simulation to compare different scenarios and find the number of operators required to avoid inline shortages. They analyze the scenarios with respect to the number of kanbans simultaneously taken in charge by operators.

Battani et al. (2015) addressed the design of the automated part logistic system in which a supermarket used as a warehouse. They also developed an analytic model to select the most appropriate transportation system.

Bortolini et al. (2015) developed an analytical cost model to optimize the Kanban number through the minimization of the total cost function.

2.4. Comparison of Different Feeding Policies

Bozer and McGinnis (1992) developed a descriptive model to compare the kitting and the line stocking policies. In their specific example, they showed that floor spaced requirements and the average WIP decreases with kitting.

Karlsson and Thoresson (2011) developed a guiding manual to manage the transformation of feeding policy for automobile companies from sequenced material flows into kitted material flow. They determined which parts should be delivered by a kitting policy.

Caputo and Pelagagge (2011) compared three different approaches based on cost (personnel and equipment) and performance (WIP) by using quantitative benchmarks and descriptive models. They showed that hybrid methods yield better results in terms of the total cost than pure feeding approaches.

Caputo et al. (2013) developed an integer linear programming model to choose the optimal feeding policy for each part. The model calculates an average cost as a function of cost generating actions and items for each policy. Their results show that applying the same feeding policy to all parts may yield poor results in terms of total cost. They proposed that part feeding policy of each part should be determined separately.

Caputo et al. (2015a) developed a model to measure the efficiency of kitting policy based on various cost items such as safety stock-holding cost, WIP holding cost, cost for floor occupation at workstations. Their descriptive model considers resource size and computation of systems' economic performances for kitting. The model also provides a quantitative benchmark to compare the efficiency of kitting with other part feeding policies.

Caputo et al. (2015b) developed an optimization model to choose cost minimizing part feeding policy. Their model determines the most cost-efficient part feeding policy for each part. They showed that for different parts different hybrid policies could yield better results.

Caputo et al. (2015c) developed an analytical model to compare JIT and line stocking part feeding policies. Their model includes some new critical cost factors such as error cost. They showed that for different parts different hybrid policies could yield better results.

Caputo et al. (2016) compared different part feeding policies. They developed a parametric model for three feeding policies and mapped areas where each feeding policy is more efficient.

Caputo et al. (2018) made a sensitivity analysis and a parametric analysis to explore the impact of part features on total delivery cost for different feeding policies. They mapped the areas where each feeding policy is more efficient, and this enables to choose the best feeding policy for each part.

Hanson et al. (2012) compared the time required to fetch the parts for kitting and line stocking policies using ANOVA. They showed that kitting has shorter fetching time than line stocking. They also discussed other advantages and disadvantages of both policies such as space requirement, pre-sorting time etc.

Hanson and Brolin (2013) compared the efficiency of kitting system with continuous supply based on their real-life observations. They tried to show that under which conditions the kitting system was more efficient than the continuous supply.

Faccio (2014) developed a decision-making tool to determine under which condition each part feeding policy performs better. They applied their model to a case study and compared the efficiency of kitting, kanban and hybrid feeding policies. They considered the impact of product mix variations and model varieties on part feeding policy and tried to find the breakeven points to determine under which conditions each part feeding policy performs better.

Sali et al. (2015) compared different part feeding policies based on the total cost which includes part preparation before assembly, picking, in-plant transportation and storage costs. Their multi-scenario analysis shows that parameters used in the cost function have an impact on the performance of the feeding policy. Analysis of scenarios shows under which conditions a feeding policy yields a better performance.

Sali and Sahin (2016) developed a comprehensive mathematical model to choose the most appropriate feeding policy for each individual component. The model calculates an average cost as a function of cost generating factors.

Usta et al. (2017) developed a hierarchical clustering analysis and used activity-based cost methodology to compare the performance of kitting and hybrid feeding policies. Based on the different scenario analysis, their results point out that hybrid policies yield better performance.

2.5. Contribution to Literature

Many of the previous studies have a single objective that aims to improve the internal part feeding. Number of tours and stock level at each station are among the mostly addressed objectives either directly or indirectly. The only study that aims to improve both in-plant traffic and assembly line stocks with a mathematical programming model was conducted by Fathi et al. (2014). Since, they use a single objective model, to the best of our knowledge, our study is the first study that attempts to generate all nondominated solutions.

In this thesis, we use two objective functions without a priori information on paths. Since the paths and travel durations are not predefined, the exact demand of each station cannot be determined in advance. We develop a new model which splits the working period into one-hour slots and determines the demand for each of these one-hour slots. As a result, a certain demand is obtained for each time slot. Furthermore, previous studies assume all baskets are compatible and this allows transfer of different baskets together. We also address the incompatible basket problem.

In order to clearly demonstrate the aim of our study, we made a new classification table. The detailed classification based on objectives and implemented methodologies are presented in Table 1. Studies without a mathematical model are classified under the last column. Studies with a mathematical model are classified into two groups according to whether the objective function is single or multi-objective. As the main concern of this study is number of tours (NT) and stock level (SL), the objective functions other than these are classified as other (O).

As seen from the table, most of the studies have single objectives. 16 studies that focus on line stocking, address NT and SL objectives more than the other two groups. Some of these studies have different objectives and we classified them under the last column by labelling them as “Other”. Studies are classified under kitting mainly try to improve the kitting process. The questions they address are how to determine the kit size, where to store kits, when to start kitting process for a specific kit etc. Kanban studies mostly address SL and O objectives. Generally, they aim to increase the number of the kanbans per cycle. In line stocking, the main target is to minimize either the number of tours or the amount of stock besides the line.

Table 1. Literature review matrix

		Single objective MP			Multi-objective MP			Other Approaches		
		NT	SL	O	NT	SL	O	NT	SL	O
Line stocking	Salameh and Gattas (2001)									X
	Choi and Lee (2002)							X	X	
	Cunha and Souza (2008)	X	X							
	Souza et al. (2008)	X	X							
	Wänström and Medbo (2008)									X
	Golz et al. (2011)	X								
	Alnahhal M. and Noche B. (2013)					X	X	X		X
	Rao et al. (2013)		X	X						
	Zhou and Peng (2017)		X							
	Zhou B. and Xu J. (2018)			X						
Kitting	Günther et al. (1996)			X						
	Chen and Wilhelm (1997)			X						
	Carlsson and Hensvold (2008)									X
	Battani et al. (2010)									X
	Kilic and Durmusoglu (2012)			X						
	Limère et al. (2015)			X						
Kanban	Shahabudeen et al. (2002)									X
	Jerald et al. (2006)									X
	Shahabudeen P. and Sivakumar (2008)								X	X
	Emde et al. (2012a)		X							
	Emde and Boysen (2012b)			X						
	Emde and Boysen (2012c)		X	X						
	Faccio et al.(2013a)									X
	Faccio et al.(2013b)									X
	Fathi et al. (2014)				X	X				
	Lolli et al. (2015)									X
	Battani et al. (2015)									X
	Bortolini et al. (2015)									X
Comparison of Policies	Bozer and McGinnis (1992)									X
	Karlsson and Thoresson (2011)									X
	Caputo and Pelagagge (2011)									X
	Hanson et al. (2012)									X
	Hanson and Brolin (2013)									X
	Faccio (2014)									X
	Caputo et al. (2013)			X						
	Caputo et al.(2015a)									X
	Caputo et al.(2015b)			X						
	Caputo et al.(2015c)									X
	Caputo et al.(2016)									X
	Sali et al. (2015)									X
	Sali and Sahin (2016)			X						
	Usta et al. (2017)									X
	Caputo et al.(2018)									X

NT: Number of tours. SL: Stock at the line. O: Other.

CHAPTER 3

PROBLEM DEFINITION

The problem in this thesis is defined based on problem symptoms at two Turkish plants which manufacture refrigerator and dishwasher. Both plants have high level of stocks at the assembly line and high internal traffic. The assembly lines are mixed model and the conveyors are moving at a constant speed. There is only one warehouse with a single I/O point. There are many forklift drivers whose job is to feed associated assembly lines. They use forklifts to feed the assembly lines. Each forklift driver is free to make his own delivery plan for the given daily production plan. The parts are being transferred from a central warehouse to the assembly lines. Each driver is responsible to deliver a set of parts. Each part type is used at a specific station.

We address a general problem where a part could be used at more than one stations. Forklifts have limited capacity and it is not suitable to transfer different basket types in the same tour. The capacity of forklift regarding number of baskets that could be delivered is different for different basket types. The drivers do not pursue a predetermined path.

In this thesis, to determine the aggregated demand of each station the total working time is divided into slots. Demand is satisfied by delivering the demand of each slot in the previous time slots.

As desired, the amount of stock at the assembly line could be minimized by increasing the frequency of the forklifts. However, this will increase the internal traffic. In this thesis, we aim to minimize both amount of stock at the assembly line and internal traffic.

CHAPTER 4

PROPOSED METHODS

In this section, a mixed integer linear programming (MILP) model is formulated. The model gives optimal loading and delivery schedule. In the second part, a problem specific heuristic algorithm is developed to overcome the shortcoming of the mathematical model for large size problems.

We divide total working time in a shift into one-hour slots. Based on the daily production schedule and the speed of the assembly line, we determine demand of parts in each time slot. Based on the time studies, we find an average tour duration. Using the average tour time, we calculate maximum number of tours that could be performed in a time slot. The forklift drivers do not have to perform all tours in a time slot. They may deliver all demand with fewer tours than the maximum allowed number of tours.

Parts are assumed to be delivered in different basket types. It is not possible to deliver different basket types on the same tour.

The demand of a station at a time slot must be transferred in previous time slots. For example, the demand of the 3rd time slot could be transferred either in the 1st or the 2nd time slot. The stock level at the end of the 2nd time slot has to be enough to satisfy the demand of the 3rd time slot.

4.1. Exact Method

Parameters

$M = \{1,...,n_m\}$	Set of parts
$F = \{1,...,n_F\}$	Set of forklifts
$S = \{1,...,n_S\}$	Set of stations
D_{mst}	demand for *part* $m \in M$ at station $s \in S$ at time slot t
WM_f	maximum weight allowed to be transferred by forklift $f \in F$
W_m	weight of unit load of part $m \in M$ (kg)
I_{ms}	amount of part $m \in M$ remained at station $s \in S$ from previous shift
K	A very large positive constant
n	number of time slots at one shift time
h	maximum number of tours could be made by *a* forklift
basketT_m	basket type of part *m*

Decision Variables

α_{mst} = number of baskets of stocks of part $m \in M$ at station $s \in S$ at the end of time slot t

β_{mst} = stock of part $m \in M$ at station $s \in S$ at the end of time slot t

χ_{fmstr} = number of unit loads of part $m \in M$ carried to station $s \in S$ at tour *r* of time slot t by forklift $f \in F$

$$\vartheta_{fmtr} = \begin{cases} 1, \text{ if forklift } f \in F \text{ transfers part } m \in M \text{ on tour r at time slot t} \\ 0, \text{ otherwise} \end{cases}$$

$$\delta_{ftr} = \begin{cases} 1, \text{ if forklift } f \in F \text{ makes tour r at time slot t} \\ 0, \text{ otherwise} \end{cases}$$

α_{mst} is the rounded-up value of B_{mst}. For example, let's assume we have two baskets of part m at station s at time slot t. If we use 60% of parts in one basket, then B_{mst} will be equal to 1.4. But we know that the number of baskets is 2 at that station and we could track the stock in terms of basket by setting α_{mst} equal to 2.

Mathematical Model

$$min \sum_{f \in F} \sum_{t=1}^{n} \sum_{r=1}^{h} \delta_{ftr} \tag{i}$$

$$min \sum_{m \in M} \sum_{s \in S} \sum_{t=1}^{n} \alpha_{mst} \tag{ii}$$

Subject to:

$$\sum_{s \in S} \sum_{m \in M} \chi_{fmstr} * \mathrm{W}_m \le WM_f \qquad \forall f \in F, t = 1,...,n, r = 1,...,h \tag{1}$$

$$\beta_{ms(t-1)} \ge D_{mst} \qquad \forall m \in M, s \in S, t = 1,...,n \tag{2}$$

$$\alpha_{mst} \ge \beta_{mst} \qquad \forall m \in M, s \in S, t = 1,...,n-1 \quad \sum_{l=t+1}^{n} D_{msl} > 0 \tag{3}$$

$$\beta_{ms(t-1)} + \sum_{f \in F} \sum_{r=1}^{h} \chi_{fmstr} - D_{mst} = \beta_{mst} \qquad \forall m \in M, \mathrm{s} \in S, t = 1,...,n \tag{4}$$

$$\beta_{ms0} = I_{ms} \qquad \forall m \in M, \mathrm{s} \in S \tag{5}$$

$$\sum_{s \in S} \chi_{fmstr} \le \vartheta_{fmtr} * K \qquad \forall f \in F, m \in M, t = 1,...,n, r = 1,...,h \tag{6}$$

$$\sum_{m \in M} \vartheta_{fmtr} \le \delta_{ftr} * K \qquad \forall f \in F, t = 1,...,n, r = 1,...,h \tag{7}$$

$$\vartheta_{fmtr} + \vartheta_{fltr} \le 1 \qquad \forall f \in F, m \in M, l \in M, basketT_m \ne basketT_l, t = 1,...,n, \mathrm{r} = 1..h \tag{8}$$

$$\delta_{ftr} \le \delta_{ltm} \qquad \forall f \in F, l \in F, t = 1,...,n, r = 1,...,h, \mathrm{m} = 1,...,h, f > l, r > m \tag{9}$$

$$\alpha_{mst} \ge 0, \text{integer} \qquad \forall m \in M, \mathrm{s} \in S, t = 1,...,n \tag{10}$$

$$\beta_{mst} \ge 0 \qquad \forall m \in M, \mathrm{s} \in S, t = 0,...,n \tag{11}$$

$$\chi_{fmstr} \ge 0, \text{integer} \qquad \forall f \in F, m \in M, \mathrm{s} \in S, t = 1,...,n, r = 1,...,h \tag{12}$$

$$\delta_{ftr} \in \{0,1\} \qquad \forall f \in F, t = 1,...,n, r = 1,...,h \tag{13}$$

$$\vartheta_{fmtr} \in \{0,1\} \qquad \forall f \in F, m \in M, t = 1,...,n, \mathrm{r} = 1..h \tag{14}$$

The first objective function (i) aims to minimize the total number of baskets of the stock at the assembly line. The second objective function (ii) aims to minimize the total number of the tours.

Constraint (1) ensures that a load of any forklift does not exceed the capacity of that forklift. Constraint (2) forces the model to keep the stock level at least as large as the demand to prevent any shortage. Constraint (3) determines the number of baskets for the stocks. This constraint has to be satisfied if the part will be used in the remaining time slots. If $\sum_{l=t+1}^{n} D_{msl} > 0$ in constraint (3) is satisfied, then this means that there is demand for part m at station s in the remaining time slots. If there isn't any demand for that part, then α_{mst} will take the value of zero to minimize the stock objective. Constraint (4) is a balance constraint that calculates the stock by subtracting consumed parts from the delivered parts. Constraint (5) initialize starting stock level. Constraints (6) and (7) establish link between variables χ_{fmstr}, ϑ_{fmtr} and δ_{ftr}. Constraint (8) ensures that different basket types cannot be transferred on the same tour. Constraint (9) removes the symmetries and restricts the solution space for forklifts and tours.

This model has similarities with capacitated lot sizing problem both in constraints and objective function. But the concepts of both problems are quite different as well. Indeed, the capacitated lot sizing models could be used before using this model to determine which products will be produced in each time slot and this model could be used to determine schedule of parts.

In the mathematical model, it is assumed that each part could be used by more than one station. This assumption necessitates tracking the station information. For example, let's assume that β_{122}=0.3 and β_{132}=0.4 and their sum will be β_{122} +β_{132} =0.7. If we don't have station information, then the third constraint will force α_{12} (α_{mt} without station index) to be equal to 1. But we know that in total there exist two baskets in the stock, one is at station 2 and the other one is at station 3.

We use ε-constraint method to solve the mathematical model. In the next part, we give a definition of the ε-constraint method and an example to show how we apply it to our problem.

ε-constraint Method

In this method, all other objectives are transferred into the constraints except one as shown below.

$$
\begin{aligned}
&\mathit{min} \quad && f_1(\mathrm{x}) \\
&\text{Subject to:} && \\
& && f_2(\mathrm{x}) <= \mathrm{e}_2 \\
& && f_3(\mathrm{x}) <= \mathrm{e}_3 \\
& && \quad \ldots \\
& && f_p(\mathrm{x}) <= \mathrm{e}_p \\
& && x \in X
\end{aligned}
$$

x is variable, and X is set of feasible solutions. $f_i(\mathrm{x})$ is the i^{th} objective function. e_i is the bound for i^{th} objective function.

In this thesis, we address a bi-objective problem where values of both objective functions are integer, and total number of tours is expected to be in a limited range. This enables us to find all efficient points if the problem size is not very large. We first generate the solution that gives the minimum stock level. This solution also gives us an upper bound on the total number of the tours. Next, we use tour objective as a constraint and the upper bound as RHS. Then we decrease the RHS of the tour constraint one each time and resolve the problem. The termination condition is to reach an infeasible case which indicates that demand cannot be satisfied with such a small number of tours. We provide an example of the method for a small dataset of our problem below.

Let us assume that the original models is (model 1) and model for ε-constraint method is (model 2). As it is seen from model 2, we added $\partial * \mathrm{f(x)}_1$ term to the objective function. This version of the model is called as modified ε-constraint model. ∂ is a very small positive constant and the added term prevent the model to end up with a weakly nondominated solution.

Model 1	Model 2
$Min\ f(\mathrm{x})_1$ (total stock level)	$Min\ f(\mathrm{x})_1 + \partial * \mathrm{f(x)}_2$
$Min\ f(\mathrm{x})_2$ (total number of tours)	subject to
subject to	$f(\mathrm{x})_2 \leq \varepsilon$
$x \in X$	$x \in X$

Dataset 1.1 (given in Appendix B) is used to illustrate the ε-constraint method. At the first step, we take ε as a big number and find the solution that minimizes the total stock ($f(\mathrm{x})_1$). The value of the first objective function is 40 and the value of the second objective function is 18 for the first solution.

After finding the first solution, we update the RHS value as18-0.1 (ε=0.1) and find the minimum value of the first objective under these conditions which is 41. Since the value of the second objective function is 17 at this solution, we set the RHS to 17- 0.1 and find the next nondominated solution.

Then, the next nondominated solution is generated as (42, 16) and the RHS is updated as 16- 0.1 to find the next solution.

All other solutions are generated by following the same procedure. At point (65, 12), we set the RHS to 12- 0.1. Since there is no feasible solution on the left side of this point, we end up with an infeasible problem. At this point, we terminate the method and provide all generated solutions as the set of nondominated points in Table 2.

Table 2. All nondominated points for dataset 1.1

Number of Baskets as Stock	Number of Tours
40	18
41	17
42	16
43	15
46	14
55	13
65	12

The mathematical model can solve small-size problems in reasonable time periods when we try to find the solution that minimizes the total stock at the line. On the other hand, when we try to minimize the total number of the tours the solution time could be too long for even small-size problems. Due to the complexity of the assembly line part feeding problem, it could be impossible to solve it for large size problems. Here we developed a problem specific heuristic algorithm to overcome this difficulty. A detailed description is given below.

4.2. Heuristic Algorithm

Below we have listed steps to prepare data to use in the proposed algorithm

1) Determine the demand of each time slot
2) Classify the demand of each time slot based on the basket types (this will help us to solve the basket type constraint before even starting to solve the problem)

Overview of the algorithm is given below.

Step 1: Check cumulative number of tours demanded for each of the time slots and if it exceeds the cumulative number of maximum tours allowed until that time slot then the problem is infeasible. Otherwise, go to step 2.

Step 2: If there is demand for the same part at same station in a future time slot then subtract the remaining stock of deliveries from the demand of that future time slot and find net demand. If the inventory stock is bigger than the demand, then subtract the demand from inventory and delete that demand. Go to step 3.

Step 3: Find the time slots whose required number of tours exceed maximum tour limit. Choose the time slot that has the highest basket demand among these violating slots. Break the tie in favor of the highest time slot index. At this step, calculate the cost of moving one tour of each basket type to previous time slots. Choose the tour that brings lowest extra stock and break the tie by selecting the one that decreases number of tours. The second tie breaking rule is selecting the one with lowest basket type index. If necessary, move the baskets of the tour by splitting to more than one previous time slot. If there is a fractionally loaded tour for a basket type, choose this one as a candidate for that basket type. If the candidate could not be moved, then the algorithm tries to choose another candidate among other basket types and it chooses the one with lowest extra stock and applies the tie breaking rules whenever there are candidates with same level of stock. Do this until all parts are assigned to tours and a feasible solution is obtained. Go to step 4.

Step 4: The solution obtained at the end of the third step, has the minimum stock level among all solutions generated by the algorithm. Keep this solution as the first solution point generated by the heuristic algorithm. Try to move fractionally loaded tours to previous time slots and merge them with fractionally loaded tours of those previous time slots to decrease the number of the tours. If there is any tour gain after merges, then for those fractionally loaded tours calculate the cost of merging in terms of stock level and move the one that has minimum cost. Break the tie by selecting the one with the highest time slot index and lowest basket type index. Do this whenever a gain from the total number of tours is possible.

The heuristic algorithm is depicted in Figure 1. Each of the colored boxes corresponds to one of the steps that are explained above and the corresponding step is labelled on the corners of the boxes.

Below, we also provide steps of the algorithm and provide a flowchart to illustrate algorithm.

Steps of the Algorithm

D_{cst} demand for basket type c at station s ∈S at time slot t

$ctour_{tc}$ cumulative number of tours required until time slot t for basket type c

$tour_{tc}$ number of tours required for basket type c to deliver the demand of the time slot t

v_c capacity of forklift for basket type c

Step1: Calculate all cumulative number of tours required until each time slot $ctour_{tc} = \frac{\sum_{s \in S} \sum_{t=1}^{l} D_{cst}}{v_c}$. If it exceeds tour limit, terminate otherwise go to step 2.

Step 2: Determine the net demand and go to step 3.

Step 3: Find time slots with the highest number of tours: max { $tour_{tc} = \frac{\sum_{s \in S} D_{cst}}{v_c}$ }. Break the tie in favor of the highest time slot index. If it doesn't exceed tour limit go to step 4. Otherwise, shift the tour with lowest extra stock (break the tie for tour gain and the lowest basket type index) and go to step 3. A detailed flowchart of this step is given in Figure 2.

Step 4: Find the tours that provide one tour gain after shifting. If there isn't such a tour, then terminate the algorithm. Otherwise, shift the tour that causes a minimum stock increase. Break the tie for highest time slot index and minimum basket type index respectively. Repeat step 4. A detailed flowchart of this step is given in Figure 3.

The first three steps use an insertion heuristic to find the first feasible solution. While applying the insertion heuristic, the violating tours and the remaining tours could be thought as two separate groups. Without the violating tours, the remaining tours already create a feasible base with minimum number of stocks by assigning each tour to the closest previous time slot. At each iteration; one tour among the violating tours is inserted into the assigned base by choosing the cheapest insertion option in terms of stock level increment.

At the fourth step, another heuristic approach is used to construct the other solution points. The saving method is used to merge tours and decrease the total number of tours. The heuristic finishes when no more tour gain is possible.

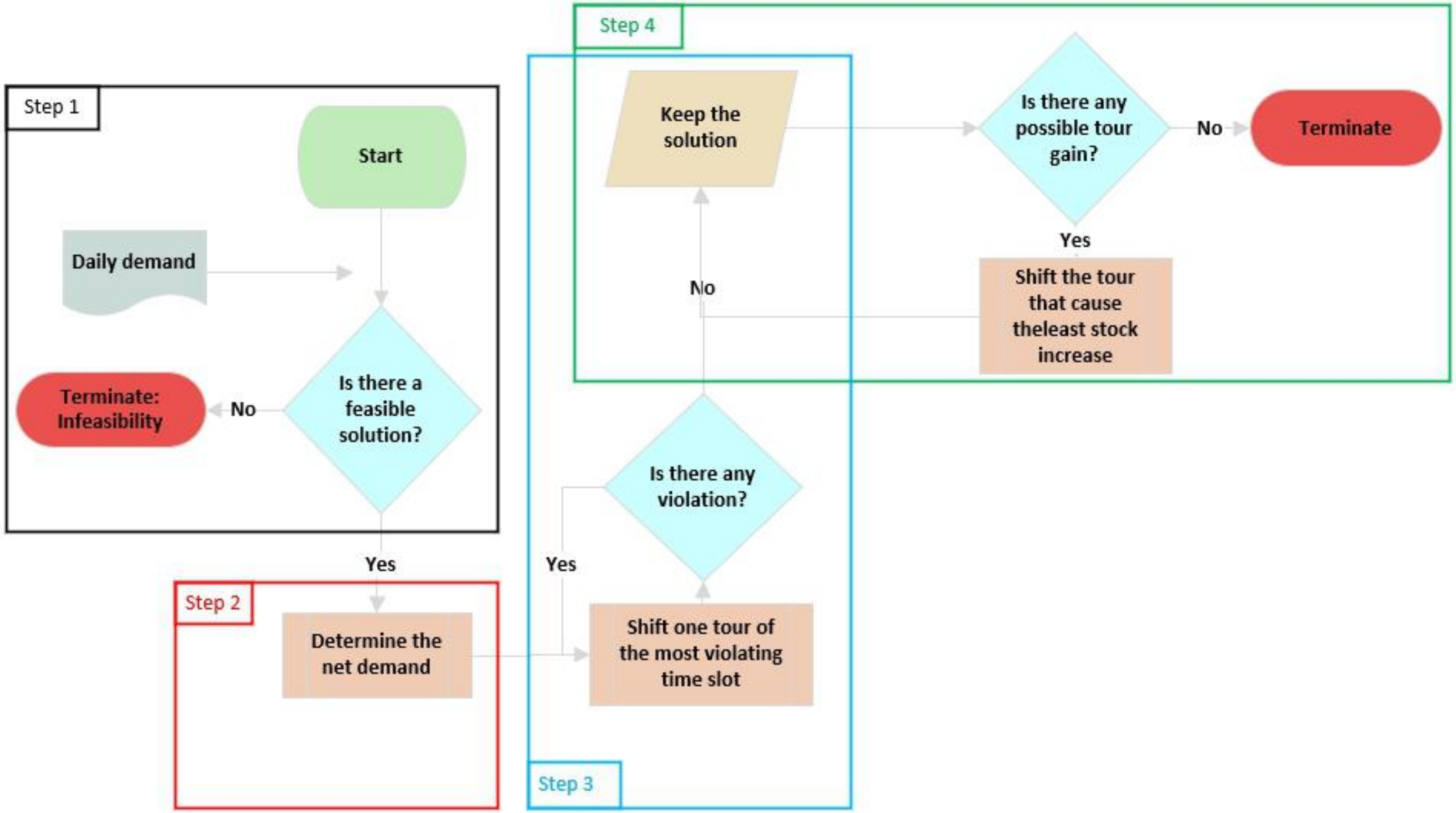

Figure 1. Flowchart of the heuristic algorithm

Figure 2 shows how infeasibilities are eliminated, and the first feasible solution is obtained. If the number of tours required to deliver demand of a time slot exceeds the number of the tours allowed, then this time slot is called violating time slot. At this step, the most violating time slot is determined. The algorithm shifts one tour of this time slot to previous time slots. After the shift, the algorithm again finds the most violating time slot and makes a shift again until all violating tours are eliminated. Shifting a tour to previous time slots increases the total stock. For example, assume that we have a tour loaded with 5 baskets and scheduled to the 4^{th} time slot. If this tour is shifted to the 3^{rd} time slot, the total stock will increase by 5.

The algorithm chooses the largest time index and smallest basket type as tie breaking rule. Choosing the smallest time index could eliminate some possible merges in the future steps. So, the largest time index rule ensures that these possible merges are not eliminated. For example, assume there are partially loaded tours at time slots 3, 4, and 5. Three of them are from the same basket type and forklift capacity is enough to transfer three of them together. If we choose the 5^{th} time slot as a candidate, then the first merge will happen at the 4^{th} time slot. But if we choose the 4^{th} time slot as a candidate slot then we will never merge the 4^{th} and 5^{th} time slots' demand at time slot 4. By doing this, we may miss some of the solutions. Therefore, the tie breaking rule is in favor of the largest time index.

The second tie break is in favor of the smallest basket type. We arrange our data in a way that smallest basket type has the highest capacity in terms of number baskets that could be transferred by a forklift. In this case, the total capacity is filled with many small baskets. So, we expect that these small units will provide more merging opportunity by being able to be distributed to many previous time slots.

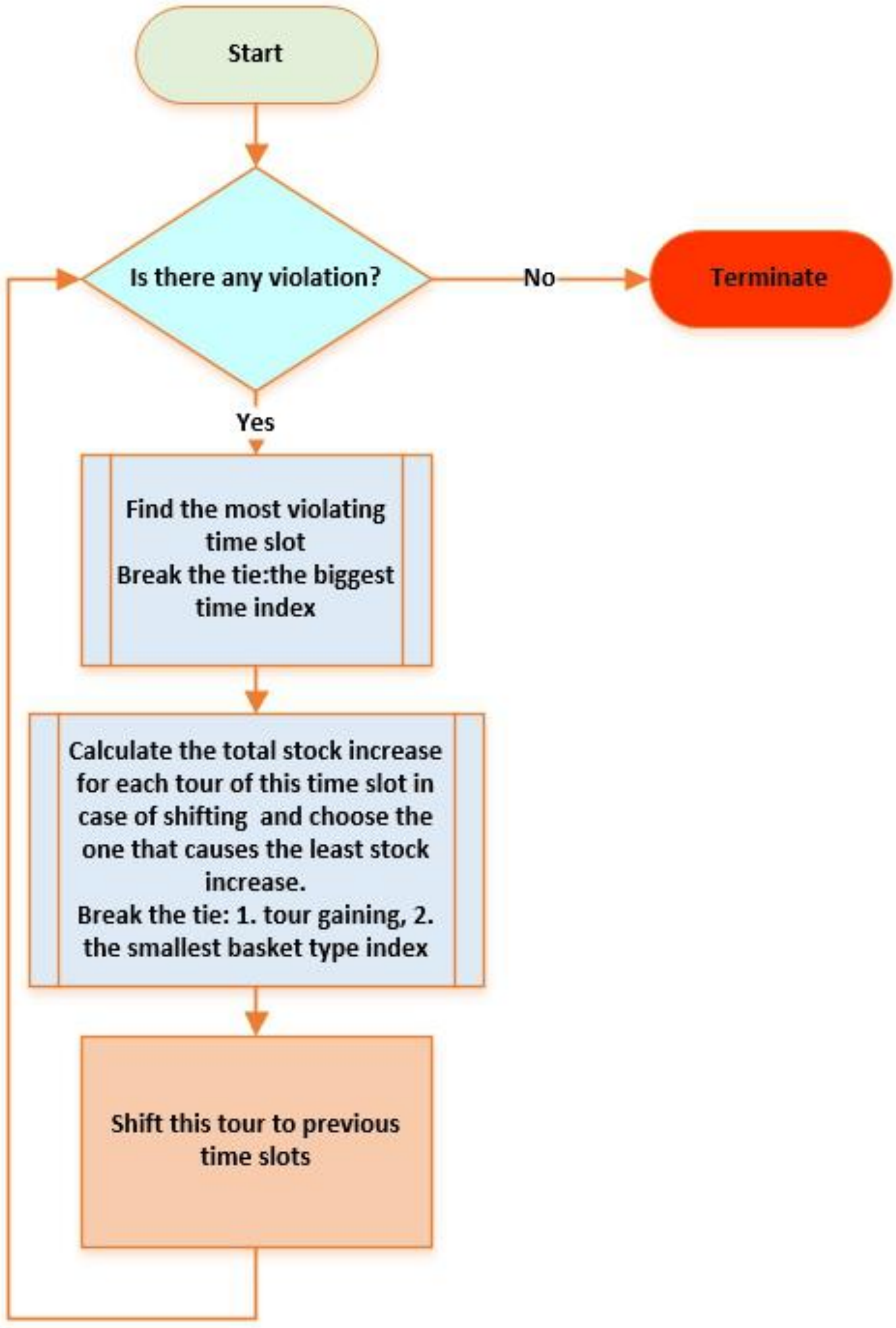

Figure 2. Details of the process that shift one tour of the most violating time slot

Figure 3 shows how we merge two tours to gain one tour and obtain all remaining solutions by repeating this step.

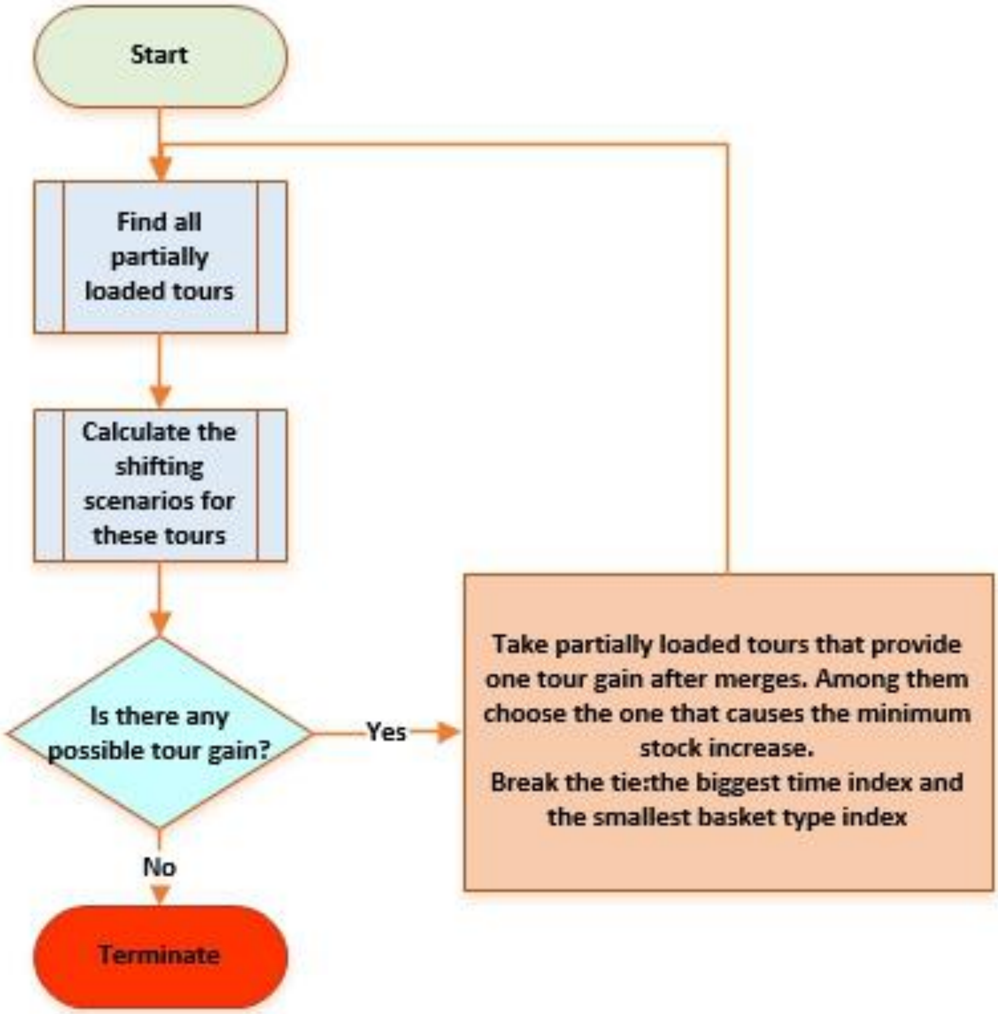

Figure 3. Details of the process that combines partially loaded tours

4.3. An Illustrative Example

In this section we explain the proposed heuristic algorithm on small-size illustrative example problem. Table 3 presents the demands of each station as a number of baskets. For simplicity we assume that the demand of the first time slot is delivered from the previous shift and we take the demand as zero for this time slot. Since step 1 and step 2 are independent we will first explain step 2 and then go to step 1. Step 2 is just a data normalization but step 1 checks the feasibility and the algorithm does not enter step 2 if it is infeasible. Here using net demands makes it easy to track so we will start with step 2. Because, by using net demand, we just work with integer values since the number of baskets will be integer.

Table 3. Demand of parts for each station (basket)

	Time Slot 1										Time Slot 2										Time Slot 3										Time Slot 4									
Parts/Stations	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0	1.7	0	0	0	0	0	0	1.2	0	0	0	0	0	0	0	1.6	0	0	1	0	1.4	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	1.5	0	0	1.2	0	0	0	0	0	0	1.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	0	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.8	0	1	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	2.7	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	1.7	0	0	0	0	0	0	0	0.1	0	0.7	0	0	0	0	0	0	0	1	0	0	0	1	0	0.8
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2.4	0	0	0	1.7
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0	0	0	0	0	0	0	0

Table 4. Net demand of parts for each station (basket)

	Time Slot 1										Time Slot 2										Time Slot 3										Time Slot 4									
Parts/Stations	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0	1.7	0	0	0	0	0	0	1.2	0	0	0	0	0	0	0	1.6	0	0	1	0	1.1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	1.5	0	0	1.2	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	0	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.8	0	1	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	2.6	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	1.7	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0.7	0	0	0	1	0	0.8
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2.4	0	0	0	1.7
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0	0	0	0	0	0	0	0

Step 2

Using the demand data, we calculated the net demand of each time slot. The demand for part 1 at the first station in the second time slot is 1.7. In order to satisfy it 2 baskets of part 1 must be delivered at the first time slot. After the consumption of the first time slot, the remaining part will be 30% of a basket. Since the demand for the same part at the same station in the last time slot is 1.4, after subtracting the remaining stock the net demand will be 1.1 in the last time slot. Another example is demand of part 6 at station 4 in the first time slot is 70% of a basket. In order to satisfy the demand, one basket must be delivered. Demand at the third time slot is 10% of a basket for this part. This means that we could delete the demand of the third time slot to calculate net demand and the remaining part stock will be 20% after this time slot and this remaining part will be used at the last time slot. Demand in the last time slot is one basket and the remaining part stock will decrease the demand to 80% of a basket. In this manner, we calculated the net demand of parts at each station for all time slots as in Table 4.

After calculating the net demand, we rounded up these data to obtain the number of baskets that should be delivered as in Table 5. Step 2 ends here, and we next show how step 1 is performed.

Table 5. Number of baskets should be delivered to satisfy the demand

Parts/Stations	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0
2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Parts/Stations	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	2	0	0	1	0	2	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0
4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1
7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0	0	0	2
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Step 1

The algorithm calculates the tour demand of each time slot by using basket types and capacity information as provided in Table 6.

Table 6. Basket type and capacity information for each part

Parts	Basket Type	Capacity of Forklift
1	1	6
2	2	4
3	1	6
4	3	3
5	4	2
6	2	4
7	3	3
8	4	2

In order to calculate the tour demand of any time slot, algorithm consolidates the same basket types' demand.

For example, for the second time slot: Parts 2 and 6 are in the same group (Basket type=2) and their total basket demand is 4 (2+2) and 3 (1+2) respectively and their total demand is 7. The forklift capacity for the second basket type is 4, and this implies that at least 2 tours required at the first time slot to provide demand of the second time slot for parts 2 and 6. Below we present all calculations for the first step.

$$\text{tour}_{21} = \frac{(2+2)^{*} + 0^{**}}{6} = 2/3 \qquad {}^{*}part\,1, {}^{**}part\,3$$

$$\text{tour}_{22} = \frac{(2+2)^{*} + (1+2)^{**}}{4} = 7/4 \qquad {}^{*}part\,2, {}^{**}part\,6$$

$$\text{tour}_{23} = \frac{0^{*} + 0^{**}}{3} = 0 \qquad {}^{*}part\,4, {}^{**}part\,7$$

$$\text{tour}_{24} = \frac{1^{*} + 1^{**}}{2} = 2/2 \qquad {}^{*}part\,5, {}^{**}part\,8$$

As seen from the calculations the total tour requirement is 4 and does not exceed the allowed tour capacity of 6.

Below we also provided the calculation until the third time slot to show how the cumulative tour requirement is derived.

$$\text{tour}_{31} = \frac{(2+2)^{*} + (2+1)^{**}}{6} = 7/6 \qquad {}^{*}2^{nd}\ time\ slot,\ {}^{**}3^{rd}\ time\ slot$$

$$\text{tour}_{32} = \frac{(2+2+1+2)^{*} + (1+1)^{**}}{4} = 9/4$$

$$\text{tour}_{33} = \frac{(0+0)^{*} + (1+1)^{**}}{3} = 2/3$$

$$\text{tour}_{34} = \frac{(1+1)^{*} + (1+1)^{**}}{2} = 4/2$$

As seen from the calculations, the cumulative tour requirement is 8 for the consecutive two time slots and the cumulative allowed tour capacity is 12 which is the sum of 6 tours of each time slot.

We could calculate the tour requirement of each time slot in this fashion. We could check the feasibility of this problem by just also calculating the cumulative tour requirement of the fourth time slot. We have not presented this last calculation here, but the result does not violate the feasibility and this specific example has a feasible solution. If the problem is infeasible then the number of forklifts should be increased.

Step 3

After showing the feasibility and calculating the net demand, we apply step 3. Here we calculate the tour demand of each time slot separately based on the net demands. We already calculated the demand of the second time slot at step 1 of the algorithm. We present below the calculations for time slots 3 and 4.

$$\text{tour}_{31} = \frac{(2+1)^{*} + 0^{**}}{6} = 1/2 \qquad {}^{*}part\,1,\ {}^{**}part\,3$$

$$\text{tour}_{32} = \frac{1^{*} + 1^{**}}{4} = 1/2 \qquad {}^{*}part\,2,{}^{**}\ part\,6$$

$$\text{tour}_{33} = \frac{1^{*} + 1^{**}}{3} = 2/3 \qquad {}^{*}part\,4,{}^{**}\ part\,7$$

$$\text{tour}_{34} = \frac{1^{*} + 1^{**}}{2} = 1 \qquad {}^{*}part\,5,{}^{**}\ part\,8$$

$$\text{tour}_{41} = \frac{2^{*} + (2+1)^{**}}{6} = 5/6 \qquad {}^{*}part\,1,\ {}^{**}part\,3$$

$$\text{tour}_{42} = \frac{(1+1)^{*} + (1+1+1)^{**}}{4} = 5/4 \qquad {}^{*}part\,2,{}^{**}\ part\,6$$

$$\text{tour}_{43} = \frac{1^{*} + (3+2)^{**}}{3} = 2 \qquad {}^{*}part\,4,{}^{**}\ part\,7$$

$$\text{tour}_{44} = \frac{3^{*} + 0^{**}}{2} = 3/2 \qquad {}^{*}part\,5,{}^{**}\ part\,8$$

Based on these calculations, we derived the final demand of each time slot as in the Table 7.

Table 7. Final demand of each time slot

Basket Type/Time Slot	1	2	3	4
1	0	1	1	1
2	0	2	1	2
3	0	0	1	2
4	0	1	1	2
Total Demand	0	4	4	7

We assume that the maximum tour number allowed is 6. Under these conditions only the last time slot with 7 tours requirement violates the maximum tour limit. Here, the algorithm eliminates the violation by trying to move tours of last time slot. In this case, one tour shift satisfies the requirements. If there were another time slot with the same tour number, then the algorithm still would move tours of the last time slot first. This is the result of the tie breaking rule that mandates to choose the largest time slot index. Since we have two extra tour capacity at time slot 3, we could move any of the tours in time slot 4. There is only one tour of the basket type 1 which is partially loaded and contains 5 baskets. This means that if we move this tour for one time slot then the stock number will increase by 5 units. There are 2 tours for the second basket type and it is obvious that moving the partially loaded one will bring one unit extra stock. There are 2 fully loaded tours of the third basket type and moving one of them will bring 3 units extra stock. The fourth basket type has 2 tours and again the partially loaded one is the proper candidate which brings one unit extra stock. The minimum extra stock is one unit and two basket types provide this minimum extra stock level. Our tie breaking rule leads us to choose the one that provides tour gain. In this case we choose the second basket type as the candidate to move.

This shift decreases the total tour requirement of the fourth time slot from 7 to 6 which is within the limit. On the other hand, an extra basket at time slot 3 does not increase the total tour requirements. There is a partially loaded tour of second basket type at time slot 3 and this tour has 2 basket spaces. Below, the updated tour calculation is given for the second basket type at time slot 3.

$$\text{tour}_{32} = \frac{1^{*} + 1^{**} + 1^{***}}{4} = 3/4 \qquad {}^{*}\text{part 2,}\ {}^{**}\text{part 6,}\ {}^{***}\text{part 6 shifted from time slot 4}$$

Under these conditions, the incumbent situation gives the first feasible solution which is also the solution that provides the minimum stock level among all other solutions provided by the algorithm.

It is difficult to calculate the stock level of this solution manually, so we do not derive the total stock level here.

Step 4

The algorithm tries to combine tours and gain one tour to improve the total number of the tour objective.

We start the calculation from time slot 3. There is a partially loaded tour of basket type 1 with 3 baskets. At time slot 2 there is a partially loaded tour of basket type 1 with 4 baskets. The vehicle capacity is 6 for basket type 1 and we still need 2 tours for 7 baskets. This means that moving first basket type does not provide any tour improvement. So, we directly eliminate this shift option.

When we check the second basket type, the same thing happens. There is a partially loaded tour at time slot 3 with 3 baskets (one comes from time slot 4 after step 3 of the algorithm) and one partially loaded tour at time slot 2 with 3 baskets. With a vehicle capacity of 4 for basket type 2, we still need 2 tours to deliver these 6 baskets. We also eliminate this option.

When we try to move the partially loaded tour of basket type 3 same thing happens again. The total number of baskets is 4 and the vehicle capacity is 3 then we need 2 tours. So, we also eliminate this option.

There is no partially loaded tour of basket type 4 at time slot 3 so there is no shift option.

Since the algorithm assesses all shift options before making any shift at step 4, we next consider shift options for time slot 4. To clarify, below we present the number of baskets for each basket type on the partially loaded tours in Table 8. Table 8 ignores the fully loaded tours. As seen from the table, the only tour gaining move is to distribute baskets of the last tour of type 1 to the second and third time slots. Since there is only one option for shifts the algorithm performs directly this shift and terminates. If there were alternative moves, then the algorithm would choose the one that brings the minimum increase in stock.

Table 8. Number of baskets on partially loaded tours

Basket Type / Time Slots	1	2	3	4
1	0	4	3	5
2	0	3	3	0
3	0	0	2	0
4	0	0	0	1

Three baskets of the 4th time slot are shifted to the 3rd time slot and two baskets are shifted to the 2nd time slot. The new tour schedule is presented in Table 9. Since all baskets of the 4th time slot for type 1 are merged with partially loaded tours of previous time slots, the total number of tours is decreased by one.

Table 9. Number of baskets on partially loaded tours after the first shift

Basket Type / Time Slots	1	2	3	4
1	0	6	6	0
2	0	3	3	0
3	0	0	2	0
4	0	0	0	1

CHAPTER 5

COMPUTATIONAL STUDY

All datasets (demands) are generated randomly, and each set size is tested for five randomly generated problems. These datasets are not included in appendices since they might take hundreds of pages.

In the first problem set, the number of parts is 8, the number of stations is 6 and demand is generated with 5% probability. We use uniform distribution (U (0,1)) for each demand cell to generate the demand. We also used uniform distribution for the probability. The mathematical model generates all nondominated points. The results show that the heuristic algorithm is able to generate a corresponding point with same tour number for all the nondominated points. The heuristic algorithm can generate the solution that minimizes the stock and some nondominated points close to this solution for tested datasets. 60% of the nondominated points are generated by heuristic algorithm. On the other hand, as we get closer to other extreme solution, the minimum stock objective starts to deviate from corresponding nondominated points. But the difference is still very small. The average difference is about 4% and standard deviation of difference is 6.9%. The average difference is obtained by dividing sum of percentage differences to number of nondominated points. The maximum difference is 23% which is occurred in the fifth run.

In the second problem set, the number of parts is 8, the number of stations is 6 and the probability is 10%. The results show that the heuristic algorithm again achieves to generate a corresponding point with the same tour number for all of the nondominated points. 60% of the nondominated points are generated by heuristic algorithm. The average difference is 1.9%, standard deviation of difference is 3.4% and maximum difference is 11%. The heuristic algorithm is again able to generate the solution that minimizes the stock for all runs.

In the third problem set, the number of parts is 8, the number of stations is 6 and the probability is 15%. Under this problem set, required number of tours is increased to 8. Changing only the required number of tours makes it difficult for the mathematical model to solve the problem. The mathematical model failed to generate some of the solutions. Heuristic algorithm generates all nondominated points that are generated by mathematical model and some other solution points.

In the fourth problem set, the number of parts is 8, the number of stations is 52 and the probability is 5%. The number of tours required is between 15 and 20 for different randomly generated problem sets. Still mathematical model only generates one nondominated point over all five runs. Heuristic algorithm easily generates all of its solutions in less than 10 seconds for this problem set.

In the fifth problem set, the number of parts is 8, the number of stations is 52 and the probability is 10%. We realized that in this problem set, parts delivered in the first time slot could also satisfy the demand in the remaining time slots significantly. So, we multiply the demand of the third time slot with 3 and demand of the fourth time slot with 4. We use the same logic also for the remaining time slots and multiply the demand with 5, 6 and 7, respectively. This ensures that the algorithm makes many shifts from the higher time slots to lower time slots. This approach improves the quality of comparison between mathematical model and heuristic approach. But the problem size is still too large for the mathematical model to generate the results. Heuristic algorithm easily generates all solutions in less than 10 seconds for this problem set, too.

In the sixth problem set, the mathematical model and heuristic algorithm are compared for a large size problem. In the sixth set, we assume there are 52 stations and 152 parts. We also assume that each basket type has 38 parts. The probability to generate demand for each time slot, station and part combination is 1%. We did this to prevent a problem set where all parts have demand in all time slots and at all stations. After putting this probability constraint, we observe that still the number of the tours required is around 30 for one time slot which is larger than the original case. This problem set shows the time efficiency of the heuristic algorithm. But the problem size is very big for the mathematical model to solve. On the other hand, the heuristic algorithm generates all solutions at once in less than 10 seconds.

All solutions are given in Table 10. Based on the solutions that mathematical model finds, 63% of all nondominated points are generated by heuristic algorithm. We do not count partially generated solutions. Since, heuristic algorithm is capable to generate the nondominated solutions that are close to extreme solution that minimize the stock level and mathematical model fails to generate solutions that are close to other extreme solution that minimizes the number of tours taking partially generated solutions into account will yield biased results. Under these conditions, if we take those points into our calculation the result mislead us to the conclusion that heuristic algorithm performs better than calculated 63%. Average difference of the stock objective between solutions generated by exact and heuristic methods is 3% and standard deviation of difference is 5.6%.

Based on the results, we could say that the heuristic algorithm solves the problem in a very short time even for large size problems for tested datasets. The heuristic algorithm successfully generates the solution that minimizes stock level. The algorithm could also generate nondominated solutions close to this solution. As we get closer to other solutions, the stock level could slightly differ from nondominated points for some problems. The overall average of difference is 3%. The overall average is obtained by dividing sum of percentage difference to number of nondominated points. But for all cases, the heuristic algorithm manages to produce a corresponding point with the same number of tours. On the other hand, the number of stocks for these corresponding points is sometimes different. The overall deviation occurs at 37% of all solutions generated both methods.

Table 10. Comparison of results for the mathematical model and heuristic algorithm

Problem Set	Problem	Mathematical Model Stock	Mathematical Model Tour	Mathematical Model Time(s)	Heuristic Algorithm Stock	Heuristic Algorithm Tour	Heuristic Algorithm Time(s)	Difference (%)	Average Difference (%)	Number of Parts	Number of Stations	Max Tour Limit
1	1	40	18	4	40	18	10	0%	4,6%	8	6	6
		41	17	4	41	17		0%				
		42	16	4	42	16		0%				
		43	15	4	43	15		0%				
		46	14	4	47	14		2%				
		55	13	12	63	13		15%				
		65	12	7	75	12		15%				
	2	46	18	5	46	18	10	0%	1,1%	8	6	6
		47	17	5	47	17		0%				
		49	16	9	49	16		0%				
		51	15	8	52	15		2%				
		54	14	7	56	14		4%				
	3	44	19	4	44	19	10	0%	0,0%	8	6	6
		45	18	4	45	18		0%				
		48	17	6	48	17		0%				
		51	16	8	51	16		0%				
		57	15	7	57	15		0%				
	4	50	19	4	50	19	10	0%	3,9%	8	6	6
		51	18	5	51	18		0%				
		54	17	6	56	17		4%				
		59	16	29	66	16		12%				
	5	29	12	4	29	12	10	0%	11,4%	8	6	6
		31	11	4	32	11		3%				
		35	10	6	43	10		23%				
		41	9	6	49	9		20%				
2	1	63	22	8	63	22	10	0%	4,4%	8	6	6
		65	21	15	65	21		0%				
		67	20	58	69	20		3%				
		71	19	60	71	19		0%				
		75	18	16	80	18		7%				
		81	17	31	90	17		11%				
		91	16	931	100	16		10%				
	2	54	20	4	54	20	10	0%	0,4%	8	6	6
		56	19	35	56	19		0%				
		59	18	12	59	18		0%				
		62	17	25	63	17		2%				
	3	81	25	5	81	25	10	0%	1%	8	6	6
		82	24	6	82	24		0%				
		83	23	6	83	23		0%				
		85	22	25	85	22		0%				
		88	21	10	89	21		1%				
		91	20	9	92	20		1%				
		95	19	25	98	19		3%				
	4	87	26	4	87	26	10	0%	2%	8	6	6
		88	25	5	88	25		0%				
		89	24	11	89	24		0%				
		92	23	404	93	23		1%				
		96	22	526	104	22		8%				
	5	106	29	323	106	29	10	0%	NA	8	6	6
		107	28	4410	107	28		0%				
		NA			108	27		NA				
					114	26						

3	1	80	28	8	80	28	10	0%	NA	8	6	8
		81	27	41	81	27		0%				
		NA			83	26		NA				
					85	25						
					90	24						
					95	23						
					107	22						
	2	104	31	69	104	31	10	0%	NA	8	6	8
		106	30	556	106	30		0%				
		NA			108	29		NA				
					111	28						
					118	27						
	3	102	33	72	102	33	10	0%	NA	8	6	8
		103	32	189	103	32		0%				
		104	31	257	104	31		0%				
		NA			106	30		NA				
					109	29						
					112	28						
					121	27						
	4	112	33	6	112	33	10	0%	NA	8	6	8
		113	32	47	113	32		0%				
		NA			115	31		NA				
					120	30						
					128	29						
	5	NA			119	36	10	NA	NA	8	6	8
					120	35						
					121	34						
					124	33						
					128	32						
					130	31						
4	1	NA			286	73	10	NA	NA	8	52	15
					299	72						
	2	NA			345	95	10	NA	NA	8	52	20
					349	94						
					352	93						
	3	266	77	59	266	77	10	0%	NA	8	52	15
		NA			267	76		NA				
					268	75						
					269	74						
					270	73						
					272	72						
					287	71						
	4	NA			472	90	10	NA	NA	8	52	20
	5	NA			329	95	10	NA	NA	8	52	20
					330	94						
					331	93						
					334	92						
					336	91						
					354	90						
5	1	738	164	10	738	164	10	0%	NA	8	52	30
		NA			743	163		NA				
					765	162						
	2	NA			577	150	10	NA	NA	8	52	30
					578	149						
					579	148						
					580	147						
					582	146						
					584	145						
					615	144						
	3	NA			856	173	10	NA	NA	8	52	30
	4	NA			843	177	10	NA	NA	8	52	30
					846	176						
	5	NA			638	153	10	NA	NA	8	52	30
					639	152						
					640	151						
					643	150						
					646	149						

6	**1**	NA	496	157	10	NA	NA	152	52	30
			497	156						
			499	155						
	2	NA	518	157	10	NA	NA	152	52	30
			519	156						
			520	155						
			521	154						
			522	153						
			524	152						
			526	151						
			547	150						
	3	NA	506	159	10	NA	NA	152	52	30
			507	158						
			508	157						
			509	156						
			511	155						
			513	154						
	4	NA	527	162	10	NA	NA	152	52	30
			528	161						
			529	160						
			532	159						
			536	158						
			539	157						
			553	156						
			565	155						
	5	NA	490	156	10	NA	NA	152	52	30
			491	155						
			492	154						
			493	153						
			494	152						
			495	151						
			497	150						
			501	149						

CHAPTER 6

CONCLUSION

In this thesis, we address traffic and assembly line side stock problems of manufacturing companies. Most of the studies regarding the assembly line focus on line balancing, facility layout and lot sizing problems. Part feeding at assembly line has been addressed by relatively fewer studies. Especially, there are very few multi-objective studies. Contemporary assembly plants have high product diversity and high model variability. This makes the design of the material delivery policy a more complex task. Increasing the number of vehicles could be a solution to control stock level but this leads to the in-plant traffic problem. Therefore, a balance should be maintained between stock and traffic level to overcome this challenge.

In our study, a bi-objective mathematical model has been developed to create a loading and transportation schedule for each vehicle driver. We show that the mathematical model fails to solve large size problems. Then, we develop a heuristic algorithm to overcome the drawbacks of the mathematical model.

Based on the tested datasets, we observed that our heuristic algorithm can solve the problem size with 152 parts and 52 stations in less than 10 seconds. The heuristic algorithm not only achieves to solve problems in a very reasonable time limit but also generates a point with the same number of tours for all nondominated points of tested datasets. The failure of the mathematical model to solve the big size problems makes it impossible to compare the results of these problems.

To the best of our knowledge, our study is the first study that attempts to generate all nondominated solutions. With its time efficiency, good results and easy to apply structure, the heuristic algorithm could improve the part feeding policy of manufacturing plants.

Further studies could incorporate different objectives, which are mostly desired to be optimized, such as minimizing maximum stock, minimizing number of forklifts. Also, the dynamic part feeding approach could be used to dynamically determine the demand under predictable cases which enables to determine demand without use of time slots. Results of heuristic method could be good starting points for evolutionary heuristic methods.

In our model, we try to minimize total number of baskets at stations. We are attaching same importance to each basket type. In other words, each basket type has a weight value of 1. On the other hand, the area occupied by each basket type could be different from each other and this could be a critical concern. In that case, the objective function used in this thesis could be converted to total space occupied by baskets at stations by using different weights for each basket type. In that case, the current version of the heuristic algorithm should also be modified accordingly. This can be a further research topic.

REFERENCES

Alnahhal, M. and Noche, B. 2013. Efficient material flow in mixed model assembly lines, SpringerPlus http://www.springerplus.com/content/2/1/415 last access:20.05.2018 /23:38

Battini, D., Faccio, M., Persona, A. and Sgarbossa, F. 2010. "Supermarket warehouses": stocking policies optimization in an assembly-to-order environment, The International Journal of Advanced Manufacturing Technology, 50. 775-788.

Battini, D., Gamberi, M., Persona, A. and Sgarbossa, F. 2015. Part-feeding with supermarket in assembly systems: transportation mode selection model and multi-scenario analysis, Assembly Automation, 35(1), 149-159.

Bortolini, M., Ferrari, E., Gamberi, M., Manzini, R. and Regattieri, A. 2015. New Kanban model for tow-train feeding system design, Assembly Automation, 35(1).128-136.

Boysen, N. and Bock, S. 2011. Scheduling just-in-time part supply for mixed-model assembly lines, European Journal of Operational Research, 211. 15-25.

Bozer, Y. A. and McGinnis, L. F. 1992. Kitting versus line stocking: A conceptual framework and a descriptive model, International Journal of Production Economics, 28. 1-19.

Carlsson, O. and Hensvold, B. 2008. Kitting in a high variation assembly line – A case study at Caterpillar BCP-E, Master's Thesis, Luleå University of Technology, Sweden.

Caputo, A. C. and Pelagagge, P. M. 2011. A methodology for selecting assembly systems feeding policy, Industrial Management & Data Systems, 111(1). 84-112.

Caputo, A.C., Pelagagge, P.M. and Salini, P. 2013. Analysis and optimization of assembly lines feeding policies, Proceedings of the Summer School Francesco Turco, 73-79.

Caputo, A. C., Pelagagge, P. M. and Salini, P. 2015a. A model for kitting operations planning, Assembly Automation, 35(1). 69-80.

Caputo, A.C., Pelagagge, P. M. and Salini, P. 2015b. A decision model for selecting parts feeding policies in assembly lines, Industrial Management & Data Systems, 115(6). 974-1003.

Caputo, A. C., Pelagagge, P. M. and Salini, P. 2015c. Planning models for continuous supply of parts in assembly systems, Assembly Automation, 35(1), 35-46.

Caputo, A. C., Pelagagge, P. M. and Salini, P. 2016. Selection of assembly lines feeding policies based on parts features, IFAC-PapersOnLine, 49(12). 185–190.

Caputo, A. C., Pelagagge, P. M. and Salini, P. 2018. Selection of assembly lines feeding policies based on parts features and scenario conditions, International Journal of Production Research, 56(3). 1208-1232.

Chen, J. F. and Wilhelm, W.E. 1993. An evaluation of heuristics for allocating components to kits in small-lot, multi-echelon assembly systems, International Journal of Production Research, 31(12). 2835-2856.

Chen, J. F. and Wilhelm, W.E. 1997. Kitting in multi-echelon, multi-product assembly systems with parts substitutable, International Journal of Production Research, 35(10). 2871-2897.

Choi, W. and Lee, Y. 2002. A dynamic part-feeding system for an automotive assembly line, Computers & Operations Research, 43(1). 123–134.

Corakci, M. A. 2008. An evaluation of kitting systems in lean production, Master Thesis, University of Boras, Sweden.

Cunha, A. S. and Souza, M. C. 2008. Stronger upper and lower bounds for a hard -batching problem to feed assembly lines, Electronic Notes in Discrete Mathematics, 30. 159-164.

Emde, S., Fliedner, M. and Boysen, N. 2012a. Optimally loading tow trains for just-in-time supply of mixed-model assembly lines, IIE Transaction, 44. 121-135.

Emde, S. and Boysen, N. 2012b. Optimally locating in-house logistics areas to facilitate JIT-supply of fixed-model assembly lines, International Journal of Production Economics, 135(1). 393-402.

Emde, S. and Boysen, N. 2012c. Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines, European Journal of Operational Research, 217. 287-299.

Faccio, M., Gamberi, M., Persona, A., Regattieri, A., Sgarbossa, F. 2013a. Design and simulation of assembly line feeding systems in the automotive sector using supermarket, kanbans and tow trains: a general framework, Journal of Management Control, 24. 187-208.

Faccio, M., Gamberi, M. and Persona, A. 2013b. Kanban number optimisation in a supermarket warehouse feeding a mixed-model assembly system, International Journal of Production Research, 51(10). 2997–3017.

Faccio, M. 2014. The impact of production mix variations and models varieties on the parts-feeding policy selection in a JIT assembly system, International Journal of Advanced Manufacturing Technology, 72. 543–560.

Fathi, M., Alvarez M. J., Mehraban, F. H., and Rodríguez V. 2014. A Multiobjective Optimization Algorithm to Solve the Part Feeding Problem in Mixed-model Assembly Lines, Mathematical Problems in Engineering, 2014. 1–12.

Golz, J., Gujjula, R., Günther, H.O., Rinderer, S. and Ziegler, M. 2011. Part feeding at high-variant mixed-model assembly lines, Flexible Services and Manufacturing, 24. 119-141.

Günther, H.O., Gronalt, M. and Piller, F. 1996. Component kitting in semi-automated printed circuit board assembly, International Journal of Production Economics, 43. 213-226.

Hanson, R., Medbo, L. and Medbo, P. 2012. Assembly station design: a quantitative comparison of the effects of kitting and continuous supply, Journal of Manufacturing Technology Management, 23(3). 315-327.

Hanson, R. and Brolin, A. 2013. A comparison of kitting and continuous supply in in-plant materials supply, International Journal of Production Research, 51(4). 979–992.

Jerald, J., Asokan, P., Saravanan, R. and Delphin Carolina Rani, A. 2006. Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm, The International Journal of Advanced Manufacturing Technology, 29. 584-589.

Karlsson, E. and Thoresson, T. 2011. A comparative study of the material feeding principles kitting and sequencing at Saab Automobile, Trollhättan: creation of guiding principles of which articles to be supplied with kitting, Master of Science Thesis, Chalmers University of Technology, Sweden

Kilic, H. S. and Durmusoglu, M. B. 2012. Design of kitting system in lean-based assembly lines, Assembly Automation, 32(3).226-234.

Kilic, H. S. and Durmusoglu M. B. 2014. Advances in assembly line parts feeding policy: a literature review, Assembly Automation, 35(1). 57-68.

Limere, V., Landeghem, H. V., Goetschalckx, M., Aghezzaf, E. H. and Mcginnis, L. F. 2012. Optimising part feeding in the automotive assembly industry: deciding between kitting and line stocking, International Journal of Production Research, 50(15). 4046-4060.

Limère, V., Van Landeghem, H. and Goetschalckx M. 2015. A decision model for kitting and line stocking with variable operator walking distances, Assembly Automation, 35(1). 47-56.

Lolli, F., Gamberini, R., Giberti, C., Rimini, B. and Bondi F. 2016. A simulative approach for evaluating alternative feeding scenarios in a kanban system, International Journal of Production Research, 54(14). 4228–4239.

Rao, Y.Q., Wang, M.C., Wang, K.P. and Wu T.M. 2013. Scheduling a single vehicle in the just-in-time part supply for a mixed-model assembly line, Computers & Operations Research, 40. 2599-2610.

Salameh, M.K. and Ghattas, R.E. 2001. Optimal just-in-time buffer inventory for regular preventive maintenance, Int. J. Production Economics, 74. 157-161.

Sali M., Sahin E. and Patchong A. 2015. An empirical assessment of the performances of three line feeding modes used in the automotive sector: line stocking vs. kitting vs. sequencing, International Journal of Production Research, 53(5).1439-1459

Sali, M. and Sahin, E. 2016. Line feeding optimization for Just in Time assembly lines: An application to the automotive industry, Int. J. Production Economics, 174. 54–67.

Shahabudeen, P. and Sivakumar, G.D. 2008. Algorithm for the design of single-stage adaptive kanban system, Computers & Industrial Engineering, 54. 800-820.

Shahabudeen, P., Gopinath, R. and Krishnaiah, K. 2002. Design of bi-criteria kanban system using simulated annealing technique, Computers & Industrial Engineering, 41. 355-370.

Sivasankaran, P. and Shahabudeen, P. 2014. Literature review of assembly line balancing problems International Journal of Advanced Manufacturing Technology, 73. 1665-1694.

Sendil Kumar, C. and Panneerselvam, R. 2007. Literature review of JIT-KANBAN system International Journal of Advanced Manufacturing Technology, 32. 393–408.

Souza, M. C., Carvalho, C. R. V. and Brizon, W. B. 2008. Packing items to feed assembly lines European Journal of Operational Research, 184. 480-489.

Usta, S. K., Oksuz, M. K. and Durmusoglu, M. B. 2017. Design methodology for a hybrid part feeding system in lean-based assembly lines, Assembly Automation, 37(1). 84-102

Zammori, F., Braglia, M. and Castellano, D. 2015. Just-in-time parts feeding policies for paced assembly lines:possible solutions for highly constrained layouts, International Transactions Inoperational Research, 23. 691-724.

Zhou, B. and Peng, T. 2017. Scheduling the in-house logistics distribution for automotive assembly lines with just-in-time principles, Assembly Automation, 37(1). 51-63.

Zhou, B. and Xu, J. 2018. A novel optimized cyclic part feeding system with line-integrated supermarkets, Journal of Engineering Manufacture, DOI: 10.1177/0954405417752510

Wänström, C. and Medbo, L. 2008. The impact of materials feeding design on assembly process performance, Journal of Manufacturing Technology Management, 20(1). 30-51.

APPENDIX A

Parameters for pseudo code

D_{cst} demand for basket type c at station $s \in S$ at time slot t

D_{mst} demand of part m at station $s \in S$ at time slot t

A_{mst} residual of part m at station $s \in S$ at time slot t

$ctour_{tc}$ cumulative number of tours required until time slot t for basket type c

$tour_{tc}$ number of tours required for basket type c to deliver the demand of the time slot t

v_c capacity of forklift for basket type c

r number of tours can be made in each time slo$t$

n number of time slots

Step 1

calculate all : $ctour_{lc} = \dfrac{\sum_{s \in S} \sum_{t=1}^{l} D_{cst}}{v_c} \qquad \forall c \in C, l = 2,\ldots,n$

```
forall(t in 2..n)
 {
   if (ctour_tc >= r*(t-1)) then
       {
          the problem is infeasible
       }
   else{
         go to step 2
       }
  }
```

Step 2

forall (t in 1..6, m in parts, s in stations)

{

$A_{mst} = RoundUp(D_{mst}) - D_{mst}$

{

forall (t in 1..6)

{

forall(l in 1..6)

{

If ($A_{mst} \geq D_{ms(l+1)}$) then

{

$D_{ms(l+1)} = 0$

$A_{mst} = A_{mst} - D_{ms(l+1)}$

{

Else

{

$D_{ms(l+1)} = D_{ms(l+1)} - A_{mst}$

}

}

}

go to step 3

Step 3

calculate all : $tour_{tc} = \dfrac{\sum_{s \in S} D_{cs(t)}}{v_c}$ $\quad\quad\quad \forall c \in C, s \in S, t = 2, ..., n$

If ($\sum_{t=1}^{n} \max\{ \sum_{c \in C} tour_{tc} - r, 0\} = 0$)

{

go to step 4

}

Else

{

Find the most violating time slot:break the tie by selecting the highest time slot

Find moving cost of all tours in this time slot:break the tie by selecting minumum basket type index

Move this tour and update the state

repeat the step3

}

Step 4

If(any tour gain is possible)

{

Find moving cost of all tours :break the tie by selecting highest time slot and minumum basket type index

Move this tour and update the state

repeat the step 4

}

Else

{

break

}

APPENDIX B

Problems

Problem Set 1 / Problem 1

	Time Slot 1						Time Slot 2					
Parts/Stations	1	2	3	4	5	6	1	2	3	4	5	6
1	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2
7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
8	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,8	0,0	0,0

Time Slot 3						Time Slot 4					
1	2	3	4	5	6	1	2	3	4	5	6
0,0	0,0	0,0	0,0	0,0	1,1	0,0	0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,1
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	2,1	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,9	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

Time Slot 5						Time Slot 6					
1	2	3	4	5	6	1	2	3	4	5	6
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	4,9	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	3,1
0,0	0,0	0,0	0,0	0,0	0,0	0,0	4,2	0,0	4,0	1,5	4,5
0,0	2,4	0,0	0,0	0,0	1,9	0,0	0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,9	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

Time Slot 7						Basket
1	2	3	4	5	6	Type
0,0	0,0	0,0	0,0	0,0	0,0	1
0,0	0,0	1,5	0,0	0,0	0,0	1
0,0	0,0	4,7	0,0	0,0	0,0	2
0,5	0,0	0,0	0,0	0,0	0,0	2
0,0	0,0	0,0	0,0	0,0	0,0	3
0,0	1,6	0,0	0,0	2,5	0,0	3
0,0	0,0	0,0	0,0	0,0	0,0	4
0,0	0,0	0,0	0,0	0,0	0,0	4

TEZ İZİN FORMU / THESIS PERMISSION FORM

ENSTİTÜ / INSTITUTE

Fen Bilimleri Enstitüsü / Graduate School of Natural and Applied Sciences ☒

Sosyal Bilimler Enstitüsü / Graduate School of Social Sciences ☐

Uygulamalı Matematik Enstitüsü / Graduate School of Applied Mathematics ☐

Enformatik Enstitüsü / Graduate School of Informatics ☐

Deniz Bilimleri Enstitüsü / Graduate School of Marine Sciences ☐

YAZARIN / AUTHOR

Soyadı / Surname : KIZILYILDIRIM
Adı / Name : RAMAZAN
Bölümü / Department : ENDÜSTRİ MÜHENDİSLİĞİ / INDUSTRIAL ENGINEERING

TEZİN ADI / TITLE OF THE THESIS (**İngilizce** / English) : A MULTIOBJECTIVE APPROACH TO ASSEMBLY LINE PART FEEDING PROBLEM

TEZİN TÜRÜ / DEGREE: **Yüksek Lisans** / Master ☒ **Doktora** / PhD ☐

1. **Tezin tamamı dünya çapında erişime açılacaktır. /** Release the entire work immediately for access worldwide. ☒
2. **Tez iki yıl süreyle erişime kapalı olacaktır.** / Secure the entire work for patent and/or proprietary purposes for a period of **two year**. * ☐
3. **Tez altı ay süreyle erişime kapalı olacaktır.** / Secure the entire work for period of **six months**. * ☐

* *Enstitü Yönetim Kurulu Kararının basılı kopyası tezle birlikte kütüphaneye teslim edilecektir.*
A copy of the Decision of the Institute Administrative Committee will be delivered to the library together with the printed thesis.

Yazarın imzası / Signature .... **Tarih** / Date 27.12.2018