EXACT APPROACHES FOR THE NO WAIT FLOWSHOP PROBLEM

A Thesis

by

Ahmet Emir TUZCU

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

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
June 2019

EXACT APPROACHES FOR THE NO WAIT FLOWSHOP PROBLEM

Approved by:

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

Associate Professor Okan Örsan Özener
Department of Industrial Engineering
Özyeğin University

Professor Tonguç Ünlüyurt
Department of Industrial Engineering
Sabancı University

Date Approved: June 11, 2019

To My Parents

ABSTRACT

In this study, no wait flow shop problem, which is a variant of permutation flow shop, is investigated. In a no wait flow shop, after processing of a job is started, it must be processed completely without any delay or cut-off. This scheduling model is generally used where operations are compulsory to follow one right way after the other due. No wait flow shop problem with objective of minimizing makespan is NP-hard, therefore researchers mostly study heuristic approaches, which give near optimal solutions, because of their ease of implementation. Proposed solution generates exact solution for the n jobs and m machines no wait flow shop systems with objective of minimizing makespan in competitive times. It uses adding lazy constraints technique. In additionally, a new heuristic is proposed. This heuristic find near optimal solution and uses chain injection method.

Keywords: scheduling; no wait flowshop; makespan; exact solutions; lazy constraints; chain injection;

ÖZETÇE

Bu çalışmada, permütasyon akış tipi üretimin bir çeşidi olan beklemesiz akış tipi üretim incelenmiştir. Beklemesiz akış tipi üretimlerde, bir işin işlenmesi başladıysa, o ürün gecikmeye ve kesintiye uğrayamaz. Bu çizelgeleme modeli genelde bir biri ardına gelen prosеslerin görüldüğü yerlerde kullanılır. Ürün üretim süresinin en aza indirilmesi amaçlanan beklemesiz akış tipi üretim problemi NP-hard'dır. Bu sebeple, birçok araştırmacı optimal çözüm bulmak yerine, daha makul zamanlarda optimal çözüme yakın çözümler üreten heuristic çözümlere yönelmişlerdir. Sunulan yöntem ise, n tane iş ve m tane makinenin olduğu beklemesiz akış tıpı üretimlerin ürün üretim süresini en aza indirecek olan kesin çözümü vermektedir. Bu yöntem tembel kısıtlama tekniklerini kullanmaktadır. Ayrıca optimale yakın bir çözüm üreten bir sezgisel yöntem sunulmuştur. Bu sezgisel yöntem beklemesiz akış tipi probleminin asimetrik gezgin satıcı problemine dönüştürelerek, çözüm esnasında oluşan döngüleri zincir kırma yöntemiyle yok etmeye dayanmaktadır.

Anahtar Kelimeler: çizelgeleme; beklemesiz akış tipi üretim; ürün üretim süresi; kesin çözüm

ACKNOWLEDGEMENTS

I would like to offer my sincere gratitute to my advisor Erhun Kundakçıoğlu for his patience and constant support during my thesis study. I would also like to thank Tonguç Yavuz, who helped me with his valuable guidance and suggestions.

I would like to thank my girlfriend Merve Özata for her endless patience and help.

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Manufacturing aims to transmute raw materials to valuable products with specialized machines, labor force, processes, and operations. A manufacturing system comprises of many variables in itself; hence it is open to many disturbances that may affect production. For the continuity of production, a manufacturing system must be able to resist these disturbances. Due to this need, different manufacturing systems are emerged for different product types. According to Pinedo (2005), most commonly implemented manufacturing systems in industries are flow shops and job shops.

Job shops are implemented generally for smaller lot size but high variety production. They aim to manufacture specialized products for a small number of customers. In addition, job shops contain different general purpose machines and it does not have a linear product flow, meaning when a job is completed or end items are obtained, it continues with a different job. Each product has different production flow and operating time. Along with these characteristics of job shops, they have many advantages and disadvantages. Advantages of job shops can be listed as below;

Flexibility: Machines are not specialized for product types and order size. It means much wider variety of jobs can be handle with job shops; hence it is easy to add and discard different processes.

Easy to launch: Job shop includes general purpose of machines rather than specialized machines; hence put them on the shop floor is enough to create a job shop.

High skilled workers: Job shops need high skilled workers; so supervisory level of workers at minimum level.

Robustness: Failure at one machine does not stop production flow.

Easy to boost capacity: Simply, set up new machines on the shop floor increases the capacity of a job shop.

Disadvantages of job shops are;

Scheduling issues: Because of non-linear production flow and non-standardization, it is hard to schedule of a job shop.

Non automation: Because of the variety of products, automated systems could not be implemented to job shops.

Low production size: Because of non-linear production flow, wait and transfer procedures of both workers and semi-finished goods is excessive.

Cost of workers: Labor costs are higher for high skilled workers.

On the contrary of job shops, flow shops are implemented generally for high lot size productions and it focuses on a certain product family. Therefore, it uses product-specific processes and technologies. Chemicals, electronics, metals, plastics, and food processing industries generally use flow shop systems.

Advantages of flow shops are;

High Production Rates: Because of the linear production flow, production flow can be divided stations which have fixed processes time. Hence, it increases speed of the flow and decreases wait and transfer times.

Specialized Workers: Each station have specialized workers, so workers productivity is higher than job shops but less skilled.

High utilization of materials and labor force: Because of its nature, flow shops use their equipment and labor force in high efficiency. Stations, machines and labors generally works at optimum level without interruption.

Easy scheduling: Because of the linear production flow, it is easy to scheduling. Generally, it is shaped and restricted with machines and labors production capabilities.

Automation: Because of the repetitive job, it is easy to implement automation systems.

Disadvantages of flow shops are;

Poorly skilled workers: Flow shops does not need high skilled workers. Because each station has simple and repetitive jobs.

Not robust to changes: A disturbance at the system might be affect all production, because all semi-finished goods must be processed at all machines.

Need maintenance: For preventing any disturbance at any time, maintenance programs must be arranged.

As mentioned above, flow shops are generally implemented for high speed production lines, therefore scheduling of production lines in flow shops are vital for competitive environments. Because of this, flow shop scheduling problem (FSP) is very popular among researchers over the past last five decades.

Flow shops contain at least two consecutive machines and each job must visit all machines with same route. In other words, flow pattern must be same for all jobs. For instance, if one job is at i-th position in the first machine, then it must be at i-th position in all machines. Furthermore, jobs cannot be processed in different machines at the same time which means each job can be processed in one machine at a time. Additionally, all machines can handle only one job at a time.

In this thesis, no-wait flow shop problem (NWFSP), which is a variant of permutation flow shop (PFSP), will be discussed. The most significant characteristic of NWFSP is having non-preemptive constraint. In other words, after processing of a job is started, it must be processed completely without any delay or cut-off. If a flow shop has this feature, it is called a no-wait flow shop. This scheduling model is usually used, where operations are compulsory to follow one right way after the other due. For illustration, agile production lines that contain 6-axis robots are designed with this system.

There are many performance criteria for scheduling a flow shop. For example, minimization of total flow time (TFT), minimization of makespan, minimization of total tardiness, minimization of weighted mean completion time and due date performance are few of them. The most widely used performance measure is minimizing the makespan. Ease of implementation of makespan criterion on different kind of problems increases its popularity. However, when customer requests on delivery dates become more important, due date performance criterion comes forward.

CHAPTER II

PROBLEM DEFINITION

This chapter covers detailed formal definition of the problem. We explain the mathematical optimization model in detail. Next, an illustrative example will be provided.

2.1 Problem Formulation

In this section $F_m|nwt|C_{max}$ and $F_m|nwt|\sum C_j$ problems are defined. In machine scheduling, $F_m|nwt|C_{max}$ indicates the problem that minimizes the makespan for no wait flow shop enviroment. F_m denotes the m-machine flow shop enviroment, nwt indicates no wait constraint. We define parameters and variables first. Next, we define objective function and constraints. The notation we use is as follows:

Parameters

i: job index

j: machine index

k: position index

n: number of jobs

m: number of machines

$P_{i,j}$: process time of job i at machine j

Decision Variables

π: feasible solution

π_k: job in k-th position in solution Π

d_{π_{k-1},π_k}: minimum delay on the first machine between start of job which is in position k_1 and job which is in position k, with no-wait constraint

C_i: completion time of job i

$C_{k,j}$: completion time of job which is in position k on machine j

$X_{i,k}$: if job i occupies position k then $x = 1$, otherwise $x = 0$

C_{max}: makespan

$\sum C_j$: total flow time

Next, we define how makespan and total flow time can be computed. $\sum C_j$ and C_{max} of a sequence of the n jobs in a flow shop with no wait constraint can be given by, respectively:

$$\sum C_j = \sum_{i=2}^{n}[(n+1-i)]d_{\pi_{k-1},\pi_k} + \sum_{i=1}^{n}\sum_{j=1}^{m} P_{i,j}$$

$$C_{max} = \sum_{k=2}^{n} d_{\pi_{k-1},\pi_k} + \sum_{j=1}^{m} P_{\pi_k,j}$$

where;

$$d_{\pi_{k-1},\pi_k} = max_{1\leq j\leq m}[\sum_{h=1}^{j} P_{i,h} - \sum_{h=2}^{j} P_{k,h-1}, 0]$$

for $1 \leq i \leq n$, $1 \leq k \leq n$, $i \neq k$.

The mixed integer programming model for $F_m|nwt|C_{max}$ can be given as follows:

$$
\begin{aligned}
\min \quad & C_{max} & & (1)\\
\text{s.t.} \quad & C_{max} \geq C_{k,m}, & \forall k, & \quad (2)\\
& C_{k,j} \geq 0, & \forall k, j, & \quad (3)\\
& C_{k,j} = C_{k,j-1} + \sum_{i=1}^{n} X_{i,k}.P_{i,j}, & \forall k, j, & \quad (4)\\
& C_{k,j} \geq \sum_{i=1}^{n} X_{i,k}.P_{i,j}, & \forall k, i = 1, & \quad (5)\\
& C_{k,j} \geq C_{k-1,j} + \sum_{i=1}^{n} X_{i,k}.P_{i,j}, & \forall k > 1, j, & \quad (6)\\
& \sum_{k=1}^{n} X_{i,k} = 1, & \forall i, & \quad (7)\\
& \sum_{i=1}^{n} X_{i,k} = 1, & \forall k, & \quad (8)\\
& X_{i,k} \in \{0, 1\}, & \forall k, i. & \quad (9)
\end{aligned}
$$

Equation (1) is the objective function of the problem. Equation (2) ensures that makespan or TFT of a schedule must be equal or greater than finishing time of the last job on the last machine. Equation (3) enforces the non-negativity of each job's completion time. Equation (4) provides the relation of completion time of each job on consecutive machines. Equation (5) ensures that completion time of a job is greater or equal to the process time of the job on the first machine. Equation (6) gives the relation between two consecutive jobs on same machine. Equation (7) guarantees that each job is assigned to a position. Equation (8) guarantees that every position has only one job. Equation (9) indicates the binary variables.

2.2 Numerical Example

Bertolissi (2000) states that NWFSP consists two jobs and two machines has two different Gantt chart pattern. This two patterns can be accepted as foundation of all no-wait flows shop patterns. Flow time sequences are shown in Figure 1 and Figure 2.

For Figure 1, if $P_{j2,1} > P_{j1,2}$, makespan is equal to $P_{j1,1} + P_{j2,1} + P_{j2,2}$.

For Figure 2, if $P_{j2,1} < P_{j1,2}$, makespan is equal to $P_{j1,1} + P_{j1,2} + P_{j2,2}$.

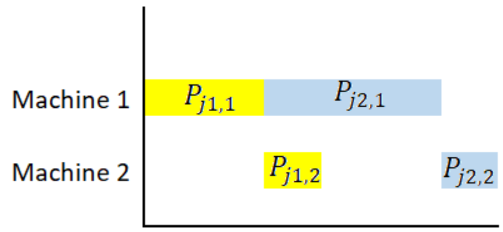

Figure 1: Gantt chart for $P_{j2,1} > P_{j1,2}$ pattern

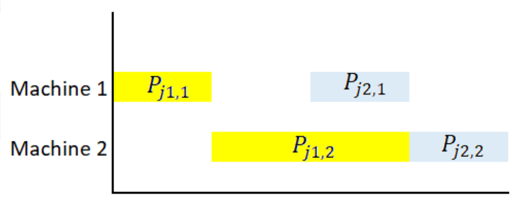

Figure 2: Gantt chart for $P_{j2,1} < P_{j1,2}$ pattern

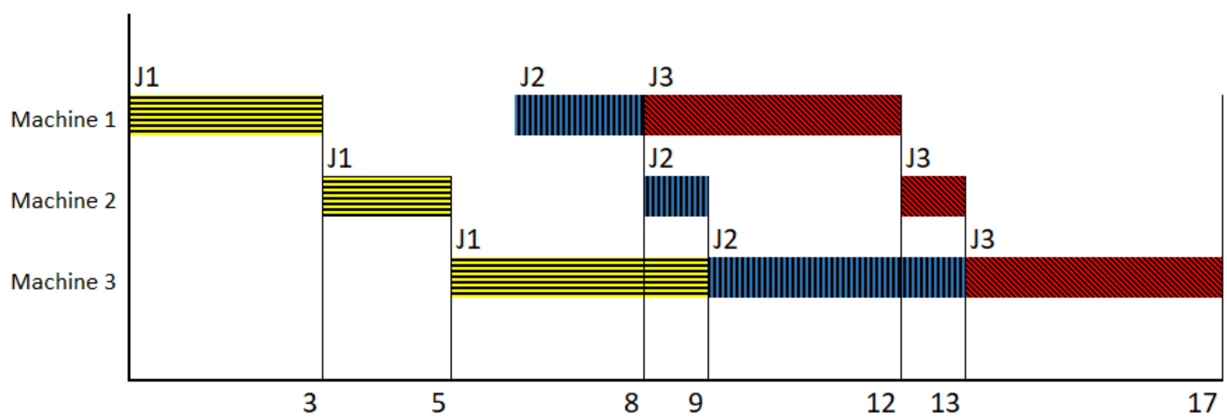

Figure 3: Gantt chart for numerical example

Table 1: Data table for numerical example

Jobs(j)	$P_{j,1}$	$P_{j,2}$	$P_{j,3}$
1	3	2	4
2	2	1	4
3	4	1	4

Process times are given in Table 1 for 3-job 3-machine problem. Job sequence is $\pi = 1, 2, 3$. Jobs visit machine 1, machine 2 and machine 3, respectively. According to the data, Gantt chart of the schedule is shown in Figure 3.

For Figure 3, j_1 starts at 0 and finishes at 3. After that, because of the no-wait constraint, it starts the process in machine 2, immediately. j_1 starts at 3, and finish at 5 in machine 2, because $P_{j1,2}$ equals to 2. After that, it starts the process in machine 3. $P_{j1,3}$ equals 4, therefore, j_1's process is completed at 9. For satisfying the no-wait constraint, j_2 starts at 6 and finishes at 8 because of its process time $P_{j2,1}$ which is 2. As seen in Figure 3, j_2 and j_3 also have delays for fulfilling the no-wait constraint.

CHAPTER III

LITERATURE REVIEW

There is a large number of NWFSP studies in the literature. The solutions of NWFSP can be considered under two categories. These are constructive heuristic solutions and metaheuristic solutions. Constructive heuristics are mostly greedy solutions. Calculation time of constructive heuristics is their advantage. On the other hand, metaheuristic solutions, which are generic solutions can be applied to many optimization problems. Their advantage is their high ability to find solutions in wide search regions. Hall and Sriskandarajah (1996) and Allahverdi (2016) present detailed survey, which covers studies in 1970s-1990s and 1993-2016, respectively. Nagano and Miyata (2016) also present a detailed survey on classification of constructive heuristics.

Notation for minimizing makespan can be shown as $F_m|nwt|C_{max}$ and notation for minimizing TFT can be shown as $F_m|nwt|\sum C_j$ using the 3- tuple standart notation of Graham et al. (1979). Wismer (1972) prove that NWFSP is equivalent to the cumulative Asymmetric Traveling Salesman Problem (ATSP). Sahni and Gonzalez (1976) prove that this problem is NP-hard. Röck (1984) prove that a NWFSP with three or more machines belong to the NP-hard problem type which means problem complexity increases with instance size. Because of the NP-hard nature of NWFSP, researchers mostly study heuristic solutions because of their ease of implementation. Although optimal solution is not obtained with heuristic solutions, relatively fast process time of these solutions satisfy researchers.

Heuristic of Bonney and Gundry (1976) works with slope match algorithm, which uses geometric relationship between adjacent jobs. King and Spachis (1980), one of the focus on minimum delay between adjacent jobs, which can be considered as one of the early works on this problem. Gangadharan and Rajendran (1993) and Rajendran (1994) improve those early studies with their heuristics, which are derived from Johnson (1954) rule. Besides that, Nawaz et al. (1983) give their well known heuristic NEH which inspires many researches. Laha and Chakraborty (2009) and Li and Wu (2008) use NEH algorithm in their heuristics and improve using Simulated Annealing Algorithm of (Osman and Potts, 1989) and RZ heuristics of (Rajendran and Ziegler, 1997). As well as these solutions, Fink and Voß (2003) propose metahuristics (Chin's heuristic) based on local search paradigm that focuses on minimum delays between adjacent jobs. Aldowaisan and Allahverdi (2003) propose metaheuristics such as Genetic Algorithm and Simulated Annealing. These metahuristics provide high quality solutions, but their solution times are not acceptable. Furthermore, Grabowski and Pempera (2005) present a tabu search algorithm that uses a dynamic tabu list for reducing error at near optimal solutions. Tseng and Lin (2010) improve the heuristic of Grabowski and Pempera (2005) with their empowered genetic algorithm with local novel search algorithm and achieve reducing error more effectively. Laha and Sapkal (2011) use delay matrix of Fink and Voß (2003) with shortest processing time technique and create their heuristic. Bertolissi (2000) transformes getting best sequence comparing the job pairs technique for NWFSP, which originally belongs to Chan and Bedworth (1990). In addition, Framinan et al. (2010) improve the heuristic of Bertolissi (2000) with a neighborhood search technique.

Recently, Lin and Ying (2016a) propose two matheuristics with three phases for minimizing makespan. First phase is applying modified NEH technique for obtaining initial sequence of jobs, second one is turning NWFSP to ATSP, and third is using the heuristic from (Lin and Ying, 2016a). Helsgaun (2000a) enhance the initial job sequence and third phase is achiving optimal solution by solving the corresponding binary integer problem. Computional results show matahuristics are very effective for big instances problems. Also Allahverdi and Aydilek (2015) investigate the problem for two different criteria which are makespan and total tardiness. This heuristic is a combination of simulated anneling and insertion algorithm and it is very effective for reducing error of the near optimal solution. In addition, Lin et al. (2018) provide a cloud theory-based iterative greedy algorithm for NWFSP, which is combined of modified iterated algorithm of Ruiz and Stützle (2007) and cloud theory mechanism of Torabzadeh and Zandieh (2010). This heuristic also investigate the NWFSP for different criteria which are makespan and total weighted tardiness.Engin and Güçlü (2018) propose a hybrid solution for NWFSP. This solution can be summaried as an ant colony algorithm which is based on crossover and mutation mechanism. Objective of this study is minimizing total flow time.

Related studies are shown in Table 2.

Table 2: A summary table for literature review

Author	Objective function
Bonney and Gundry (1976)	Min. Makespan
King and Spachis (1980)	Min. Total flow time
Rajendran and Chaudhuri (1990)	Min. Makespan
Gangadharan and Rajendran (1993)	Min. Makespan
Rajendran (1994)	Min. Makespan
Aldowaisan and Allahverdi (1998)	Min. Total flow time
Bianco et al. (1999)	Min. Makespan
Glass et al. (1999)	Min. Makespan
Espinouse et al. (1999)	Min. Makespan
Bertolissi (2000)	Min. Total flow time
Allahverdi and Aldowaisan (2000)	Min. Total flow time
Allahverdi and Aldowaisan (2001)	Min. Total flow time
Fink and Voß (2003)	Min. Total flow time
Grabowski and Pempera (2005)	Min. Makespan
Ruiz and Stützle (2007)	Min. Makespan
Li and Wu (2008)	Min. Makespan
Laha and Chakraborty (2009)	Min. Makespan
Ruiz and Allahverdi (2009)	Min. Makespan and Max. Lateness
Framinan et al. (2010)	Min. Total flow time
Laha and Sapkal (2011)	Min. Makespan
Aydilek and Allahverdi (2012)	Min. Makespan and Mean Completion Time
Gao et al. (2013)	Min. Makespan

CHAPTER IV

SOLUTION APPROACH

We propose mathematical optimization models to represent machine scheduling process in order to reach the exact solution. At first, problem is converted to asymmetrical travelling salesman problem.

4.1 Converting the problem to Asymmetrical Travelling Salesman Problem

Wismer (1972) points that flowshop sequencing problems can be converted to Asymmetrical Travelling Salesman Problem (ATSP). Let G represents complete digraph which is shown as $G = (V, A)$ where $V = 1, ..., n$ is vertex set and A the arc set denoted as $A = (i, j) : i, j \in V$. Cost of travelling between city/job i to city/job j represents as c_{ij} where $(i, j) \in A$ with $c_{ii} = 0$ for $i \in V$. The goal of TSP is the find Hamiltonian Cycle visit every vertex only once. Wismer (1972) and Van der Veen and van Dal (1991) shows that a feasible schedule of $F_m|nwt|C_{max}$ can be considered as a Hamiltonian tour. To expand the subject, minimal length of the road $d(\pi_a)$ which is a directed with Hamiltonian Tour ($\pi_a = 0, \pi_1, ..., \pi_n, 0$) can be considered as equal to $C_{max}(\pi_b)$ which is obtained by applying feasible schedule $\pi_a = \pi_1, ..., \pi_n$. Hence mathematical model of ATSP can be given as (Dantzig et al. (1954)):

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{10}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ij} = 1, \qquad \forall j, \tag{11}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad \forall i, \tag{12}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad S \subset V : S \neq \varnothing, \tag{13}$$

$$x_{i,j} \in \{0, 1\}, \qquad \forall i, j. \tag{14}$$

Equation (10) is the objective function which aims to minimize the total cost of the tour. Equation (11) ensures that only one arc can in to city/job j and Equation (12) ensures the only one arc can out from city/job i. Equation (13) is the subtour elimination constraint. Equation (14) represents binary decision variables.

NWFSP can be converted to ATSP where objective function maximizes the profit. Profit (p_{ij}) between two consecutive job can be defined as difference between sum of processing time of each job at each machine and possible minimum arc length (consideration under NWFSP constraints). To illustrate, profit (p_{ij}) of the system which is shown in Figure 4 and Figure 5 is equal to $P_{j2,1}$. Figure 4's C_{max} is enhanced edition of Figure 5's C_{max}, but delays between jobs are minimized.

Hence mathematical model of the system for objective of maximizing profit can be given as:

$$\max \quad \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} x_{ij} \tag{15}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ij} = 1, \qquad \forall j, \tag{16}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad \forall i, \tag{17}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad S \subset V : S \neq \varnothing, \tag{18}$$

$$x_{i,j} \in \{0, 1\}, \qquad \forall i, j. \tag{19}$$

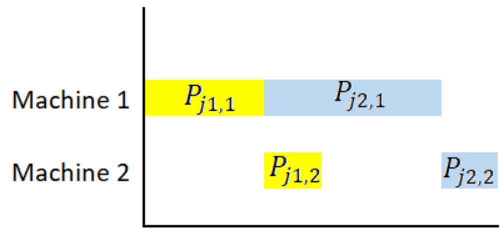

Figure 4: C_{max} equals to sum of $P_{j1,1}$, $P_{j2,1}$ and $P_{j2,2}$

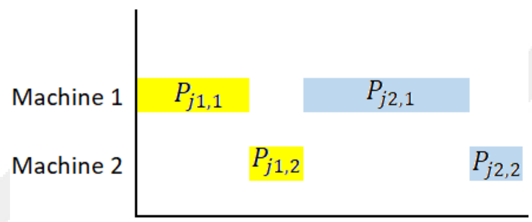

Figure 5: C_{max} equals to sum of $P_{j1,1}$, $P_{j2,1}$, $P_{j1,2}$ and $P_{j2,2}$

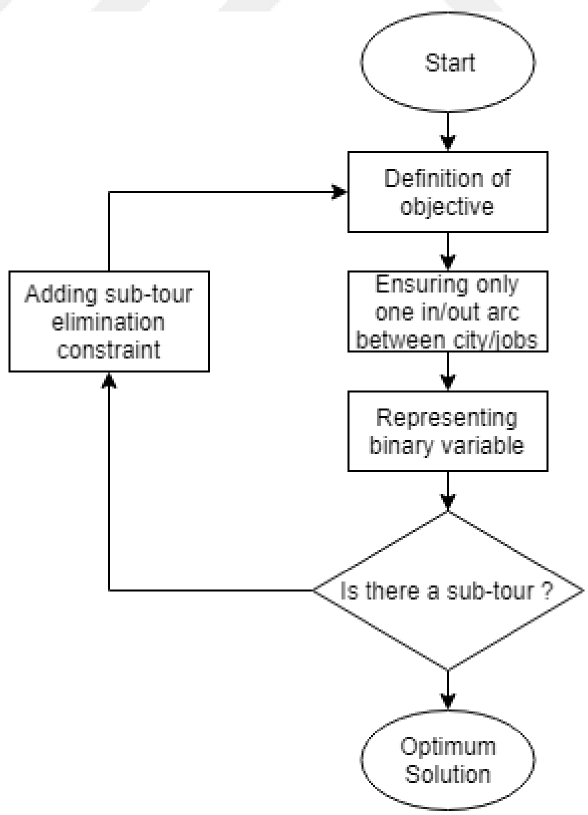

Figure 6: Adding lazy constraint procedure

4.2 Adding Lazy Constraints

NWFSP for small instances can be solved exactly in acceptable time, but for larger instances, more processing time is needed. To reduce processing times, only using needed constraints can be efficient. At this point, for increasing the performance of the model, adding lazy constraints may be conceivable.

Adding lazy constraints is combinatorial method which aims to solve integer linear problems. Adding lazy constraints is a branch and bound method which also uses cutting plane method.

The method uses the simplex algorithm of Dantzig et al. (1954) without using integer constraints. After getting an optimal solution which is suppose to be an integer value but it is not,then cutting plane algorithm is activated and new linear constraints are added. These constraints are satisfied by all feasible integer solutions, but current fractional solution is not included to feasible area. As a consequence of this method, less fractional solution is expected. After that, branch and bound algorithm is activated and non-integer solutions, which are used for LP relaxations, are considered as upper bounds of the model and integer solutions are accepted as lower bounds. If an existing upper bound lower than an existing lower bound then nodes can be cut.

In other words, if ATSP model is considered which is given above, (18) is not used when mathematical model starts. A solution is obtained for the model without using (18) and investigated it is optimal or not. If it is optimal mathematical model is solved again without any constraint of (18), but when solution is not optimal a cut is added to (18). This process continues until problem is fully solved and all cuts are added to (18). As a result, we do not know all predefined constraints of (18) is necessary, hence with adding lazy constraints we only use convenient constraints of (18) for the solution. Therefore, solution time of the model is greatly decreased.

4.3 A New Heuristic

In this section a new heuristic is proposed which is based on ATSP chain ejection/break based algorithm.

TSP problem is NP-Hard; hence, there is no polynomial time algorithm that is able to solve all instances of problem. Because of this reason, there are many heuristic models literature.

Lin and Kernighan (1973) propose stem-and-cycle algorithms which provide a basis for heuristic solutions for TSP problems. Most significant difference between these heuristics are their reference structures: Lin and Kernighan (1973)'s reference structure is based on a Hamiltonian cycle, which is constituted by dropping an edge of TSP tour. On the other hand, stem-and-cycle structure consists of one path and one cycle which are connected to each other with a root node. In addition, steam-and-cycle procedure is a specialized approach that generates dynamic alternating paths. On the other hand, Lin and Kernighan (1973) generates static alternating paths.

The most significant difference between typical TSP and ATSP is that the distance function may not be symmetrical in ATSP. Hence for two location such as u and v, it is possible that $d(u, v) \neq d(v, u)$. $d(u, v) \geq 0$ is assumed for all pairs. so triangle inequality holds $d(u, w) \leq d(u, v) + d(v, w)$. If triangle inequality does not hold $d(u.v)$ can be excepted as length of the shortest path between u and v.

Hamiltonian cycle preferred structure is discussed before, when an element such as a node, edge or path unsettles a graph's preferred structure then ”ejection” terminology comes forward. According to Glover (1996), for achieving the preferred structure of a graph, corresponding element is ejected from graph in a way that restores critical area of the graph. A chain of ejection steps are applied until preferred graph is fully retrieved.

Kanellakis and Papadimitriou (1980)'s heuristic is based on Lin and Kernighan (1973)'s procedure. To the best of our knowledge this is the only ejection chain algorithm for ATSP in literature.

In our proposed heuristic, NWFSP is first converted to ATSP problem. Next, this problem is solved without subtour elimination constraint. After detection of subtours, subtours are transformed to nodes. At this moment, we take advantage of ”directed edge” feature of ATSP. Contrary to the TSP, edges between nodes are supposed to be directed in an ATSP which means; for Figure 10, there is a path between X_1 to X_2, but there is no path between X_2 to X_1. Because of this reason, stem-and-cycle form can not be seen in an ATSP; hence, this technique can not be applied to ATSP. If we expand this example to the NWSFP. Figure 7 and Figure 8 shows that if we change sequence between j_1 and j_2. costs and profits do not stay same; although they do not change for TSP model. Because of this reason a chain injected subtour can be accepted as a node. because their input nodes and output nodes are determined. After that profits of each combination of paths are calculated and $n \times n$ gain matrix is achieved. With this matrix, ATSP solution is updated and it solved again until there is only one subtour.

For illustration , if we get a solution like Figure 9 when the transformed NWFSP without (18) is calculated, firstly we determine the most expensive/worst cost path for these subtours. In this example, X_2X_3, Y_4Y_1, and Z_2Z_1 are most expensive paths. After that, these paths are ejected from subtours and new paths are obtained which are X_3X_2, Y_1Y_4, and Z_1Z_2. As we mentioned above, we can consider this obtained path as a nodes, because they carry the node's one input-one output feature. After obtaining these paths, $n \times n$ gain matrix is achieved with calculating the alternative paths between newly occured input/output nodes. For example, in Figure 10, X_2Y_1, Y_4Z_2, and Z_1X_3 alternative paths are created. After these paths are created only one cycle is remained which consist all nodes.

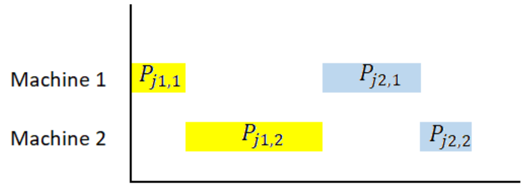

Figure 7: Costs/profits of directed paths/sequences are different for same nodes/jobs

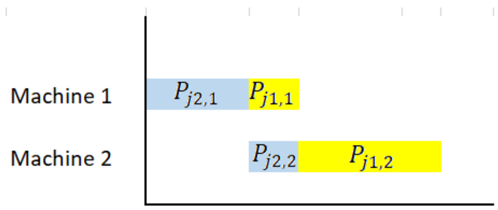

Figure 8: Costs/profits of directed paths/sequences are different for same nodes/jobs

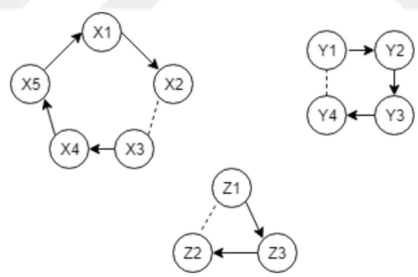

Figure 9: An example of a output for ATSP without subtour elimination

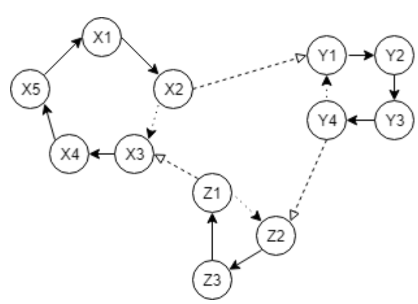

Figure 10: Breaking worst cost edges and creating alternative paths

Algorithm 1: A New Cycle Break Based Heuristic

Result: A near optimal solution for NWFSP

$readProblemData()$;

$convertNWFSPtoATSP()$;

while *number of subtour* $\neq$ *1* **do**

  $solveATSPwithoutSubtourElimination()$;

  $detectSubtours()$;

  if *numberofSubtour* $>$ *1* **then**

    $transformSubtoursToNodes()$;

    $calculateGainMatrix()$;

    $updateATSP()$;

  end

end

In Algorithm 1, our main program is illustrated. Firstly problem datas are read from source and NWSFP is converted to the ATSP. After that, without subtour elimination constraint ATSP is solved. After detection of subtours, subtours are transformed to nodes, in like Figure 9. After that, alternative path matrix is created, in like Figure 10. With this newly created nodes and gain matrix ATSP is solved again, until there is only one subtour.

Algorithm 2: transformingSubtourstoNodes()

Result: Transformed subtours

$findWorstCostForEachSubtour((subtourCostArray))$;

$findIndexofStartingNodes(worstCostArray())$;

$breakWorstCostPath(startingNodesArray())$;

In Algorithm 2, subtours to node transformation is explained. For transforming to subtours to nodes, firstly we need to find most expensive/worst; hence we need all subtours and subtours' cost array. After finding all worst costs for each subtour, index of starting nodes are determined. After that, starting with that subtour node, directed path is followed until the last node.

Algorithm 3: findWorstCostForEachSubtour()

Result: An array of worst cost of subtours

for $i = 0;\ i < subtourCostArray.length;\ i++$ **do**

  $int\ worst = subTourCostArray[i][0];$

  for $j = 0;\ j < subtourCostArray[i].length;\ i++$ **do**

    if *subtourCostArray[i][j]* $>$ *worst* **then**

      $worst = subtourCostArray[i][j];$

    end

    $worstCostArray[i] = worst;$

  end

end

return $worstCostArray;$

In Algorithm 3, obtaining of worst cost of each subtour is explained. After getting $subTourCostArray[][]$, every cost of each subtour is investigated and added to $worstCostArray[]$, respectively.

Algorithm 4: findIndexofStartingNodes()

Result: An array of indexes of starting nodes of breaked subtours(paths)

for $i = 0;\ i < worstCostArray.length;\ i++$ **do**

  $int\ worst = subTourCostArray[i][0];$

  $int\ node;$

  for $j = 0;\ j < subtourCostArray[i].length;\ i++$ **do**

    if *subtourCostArray[i][j]* $=$ *worst* **then**

      $k = j;$

      $startingNodesArray[i] = k + 1;$

    end

  end

end

return $startingNodesArray;$

In Algorithm 4, with $worstCostArray[]$ every node of is before worst cost path is determined and added $startingNodesArray[]$.

Algorithm 5: breakWorstCostPath()

Result: An array of paths

for $i = 0;\ i < startingNodesArray.length;\ i++$ **do**

  $int\ counter = 0$;

  $int\ counter2 = startingNodesArray[i]$;

  if *counter < PathArray[i].length* **then**

    for $j = 0;\ j < PathArray[i].length;\ j++$ **do**

      if *counter2 < PathArray[i].length* **then**

        $PathArray[i][j] = subTourNodeArray[i][counter2]$;

      end

      if *counter2 ≥ PathArray[i].length;* **then**

        $PathArray[i][j] =$

        $subTourNodeArray[i][counter2 - PathArray[i].length]$;

      end

      $counter++$;

      $counter2++$;

    end

  end

end

return $PathArray$;

In Algorithm 5, starting node of breaked subtours is matched with newly created $PathArray[][]$, for Figure 10 , in the first cycle for loop $PathArray[0][0]$ and $subTourNodeArray[0][2]$ is matched and it goes until all nodes are filled to $PathArray[][]$. $counter$ and $counter2$ is there for true matching.

CHAPTER V

RESULTS

In this section, three benchmark test instances are applied to the ATSP model with lazy constraint (TLC) and two benchmark test instances are applied to the proposed heuristic. We present how random instances are generated and use two well-known data sets from the literature. All computations are performed using Java codes, calling Gurobi 8.0 to solve optimization problems, on a 3.5 GHz Intel Xeon (E5-1650 v2) computer with 16 GB DDR3 ECC (1866 MHz) RAM and the macOS HighSierra operating system.

5.1 Instance Generation

Three sets of test benchmark is used to investigate the efficiency of proposed algorithms.

First sets of test benchmark instances generated randomly. Processing times for each job is integer and follows a uniform distribution between 1 and 99. The numbers of jobs are $n = 1000, 1500, 2000$ and the numbers of machines are $m = 5, 10, 15, 20$. Thus, there are 12 combinations. Every combination has 5 test instances; therefore there are 60 test instances.

Second test benchmark is proposed by Vallada et al. (2015). It includes 240 small-scale instances and 240 large-scale instances. For small-scale instances, the number of jobs are $n = 10, 20, 30, 40, 50, 60$ and the numbers of machines are $m = 5, 10, 15, 20$. Thus, there are 24 combinations. Every combination has 10 test instances. For large-scale instances, the number of jobs are $n = 100, 200, 300, 400, 500, 600, 700, 800$ and the numbers of machines are $m = 20, 40, 60$. Thus, there are 24 combinations. Every combination has 10 test instances.

Third test benchmark is proposed by Reeves (1994). It includes 21 test instances.The number of jobs are $n = 20, 30, 50, 75$ and the numbers of machines are $m = 5, 10, 15, 20$.

Seven combinations is used and these combinations have 3 test instances.

5.2 Exact Solution Performance

In this section, three benchmark test instances are applied to the ATSP model with lazy constraint (TLC). Tables include instances' names. number of jobs n. number of machines m. optimal solutions for C_{max} and. solution times. respectively.

First test benchmark is randomly generated test instances. Table 3 shows results of TLC.

Table 3: Results of ATSP Model with lazy constraints for randomly generated benchmark test instances

Inst.	n	m	Opt. Sol.(C_{max})	Sol. T.(s)	Inst.	n	m	Opt. Sol.(C_{max})	Sol. T.(s)
Rndm1	1000	5	56548	79.905	Rndm31	1500	15	119523	363.993
Rndm2	1000	5	56436	35.715	Rndm32	1500	15	118633	1073.308
Rndm3	1000	5	56741	34.892	Rndm33	1500	15	118714	157.548
Rndm4	1000	5	56822	51.344	Rndm34	1500	15	118809	1142.287
Rndm5	1000	5	56419	57.728	Rndm35	1500	15	119691	652.175
Rndm6	1000	10	69280	59.719	Rndm36	1500	20	132450	194.958
Rndm7	1000	10	70129	381.036	Rndm37	1500	20	131975	1067.024
Rndm8	1000	10	70015	31.063	Rndm38	1500	20	132166	844.198
Rndm9	1000	10	69940	84.109	Rndm39	1500	20	132269	198.711
Rndm10	1000	10	70042	81.650	Rndm40	1500	20	131943	1190.503
Rndm11	1000	15	80753	47.398	Rndm41	2000	5	110687	422.459
Rndm12	1000	15	80554	56.138	Rndm42	2000	5	112177	1086.118
Rndm13	1000	15	80911	164.657	Rndm43	2000	5	112074	651.915
Rndm14	1000	15	80604	171.165	Rndm44	2000	5	111320	222.792
Rndm15	1000	15	80209	34.274	Rndm45	2000	5	110764	460.627
Rndm16	1000	20	89961	69.157	Rndm46	2000	10	136223	1328.609
Rndm17	1000	20	89535	174.387	Rndm47	2000	10	135050	1687.448
Rndm18	1000	20	89548	214.396	Rndm48	2000	10	135651	2718.401
Rndm19	1000	20	89674	122.775	Rndm49	2000	10	135383	1087.824
Rndm20	1000	20	90068	136.536	Rndm50	2000	10	135381	320.291
Rndm21	1500	5	84607	98.151	Rndm51	2000	15	155378	496.143
Rndm22	1500	5	84037	102.290	Rndm52	2000	15	156201	397.610
Rndm23	1500	5	83930	164.964	Rndm53	2000	15	157006	658.622
Rndm24	1500	5	84559	91.892	Rndm54	2000	15	156503	4388.642
Rndm25	1500	5	84300	137.695	Rndm55	2000	15	156604	329.398
Rndm26	1500	10	103605	174.502	Rndm56	2000	20	173232	540.121
Rndm27	1500	10	102935	239.305	Rndm57	2000	20	174383	1626.233
Rndm28	1500	10	103290	90.092	Rndm58	2000	20	173615	1536.101
Rndm29	1500	10	102629	129.907	Rndm59	2000	20	173917	1109.602
Rndm30	1500	10	102690	452.137	Rndm60	2000	20	174178	1909.635

This table shows the success of lazy constraints. It can be seen that even the largest instances can be solved in less than an hour.

Second test benchmark is Vallada et al. (2015)'s test benchmark. Tables 4-6 show results of TLC.

Table 4: Results of ATSP Model with lazy constraints for Vallada et al. (2015), Part I

Inst.	n	m	Opt. Sol.(C_{max})	Sol. T.(s)	Inst.	n	m	Opt. Sol.(C_{max})	Sol. T.(s)
10_10_1	10	10	1253	0.0138	20_10_10	20	10	1963	0.0130
10_10_10	10	10	1317	0.0040	20_10_2	20	10	1998	0.0096
10_10_2	10	10	1278	0.0183	20_10_3	20	10	2036	0.0180
10_10_3	10	10	1171	0.0161	20_10_4	20	10	1932	0.0158
10_10_4	10	10	1181	0.0034	20_10_5	20	10	2032	0.0165
10_10_5	10	10	1294	0.0140	20_10_6	20	10	2059	0.0136
10_10_6	10	10	1198	0.0030	20_10_7	20	10	2051	0.0596
10_10_7	10	10	1256	0.0145	20_10_8	20	10	2018	0.0225
10_10_8	10	10	1220	0.0131	20_10_9	20	10	1979	0.0134
10_10_9	10	10	1243	0.0215	20_15_1	20	15	2663	0.0079
10_15_1	10	15	1516	0.0151	20_15_10	20	15	2519	0.0300
10_15_10	10	15	1687	0.0075	20_15_2	20	15	2523	0.0284
10_15_2	10	15	1596	0.0032	20_15_3	20	15	2392	0.0143
10_15_3	10	15	1611	0.0051	20_15_4	20	15	2392	0.0309
10_15_4	10	15	1649	0.0108	20_15_5	20	15	2502	0.0117
10_15_5	10	15	1602	0.0297	20_15_6	20	15	2634	0.0786
10_15_6	10	15	1529	0.0211	20_15_7	20	15	2580	0.0064
10_15_7	10	15	1702	0.0075	20_15_8	20	15	2521	0.0133
10_15_8	10	15	1720	0.0048	20_15_9	20	15	2511	0.0467
10_15_9	10	15	1683	0.0192	20_20_1	20	20	3082	0.0084
10_20_1	10	20	1913	0.0102	20_20_10	20	20	2884	0.0349
10_20_10	10	20	1876	0.0075	20_20_2	20	20	2872	0.0071
10_20_2	10	20	1973	0.0091	20_20_3	20	20	2935	0.0781
10_20_3	10	20	1989	0.0074	20_20_4	20	20	2828	0.0061
10_20_4	10	20	1971	0.0101	20_20_5	20	20	3078	0.0261
10_20_5	10	20	1979	0.0091	20_20_6	20	20	3172	0.0185
10_20_6	10	20	2152	0.0103	20_20_7	20	20	2999	0.0144
10_20_7	10	20	1893	0.0040	20_20_8	20	20	2837	0.0416
10_20_8	10	20	1933	0.0231	20_20_9	20	20	3094	0.0145
10_20_9	10	20	1941	0.0030	20_5_1	20	5	1414	0.0170
10_5_1	10	5	760	0.0048	20_5_10	20	5	1546	0.0112
10_5_10	10	5	719	0.0028	20_5_2	20	5	1481	0.0079
10_5_2	10	5	759	0.0327	20_5_3	20	5	1588	0.0798
10_5_3	10	5	823	0.0112	20_5_4	20	5	1355	0.0130
10_5_4	10	5	776	0.0042	20_5_5	20	5	1520	0.0116
10_5_5	10	5	798	0.0049	20_5_6	20	5	1333	0.0184
10_5_6	10	5	849	0.0177	20_5_7	20	5	1388	0.0465
10_5_7	10	5	843	0.0029	20_5_8	20	5	1340	0.0054
10_5_8	10	5	768	0.0089	20_5_9	20	5	1499	0.0122
10_5_9	10	5	841	0.0038	30_10_1	30	10	2653	0.0423
30_10_10	30	10	2647	0.0496	40_10_10	40	10	3447	0.0187
30_10_2	30	10	2861	0.0925	40_10_2	40	10	3416	0.0160
30_10_3	30	10	2796	0.0151	40_10_3	40	10	3408	0.0318
30_10_4	30	10	2762	0.0130	40_10_4	40	10	3622	0.0351
30_10_5	30	10	2773	0.0188	40_10_5	40	10	3488	0.0301
30_10_6	30	10	2808	0.0251	40_10_6	40	10	3565	0.0867
30_10_7	30	10	2683	0.0279	40_10_7	40	10	3496	0.0335
30_10_8	30	10	2532	0.0105	40_10_8	40	10	3427	0.0234
30_10_9	30	10	2693	0.0175	40_10_9	40	10	3501	0.0530
30_15_1	30	15	3347	0.0422	40_15_1	40	15	4370	0.1046
30_15_10	30	15	3390	0.1466	40_15_10	40	15	4301	0.0362
30_15_2	30	15	3243	0.0164	40_15_2	40	15	4214	0.0253
30_15_3	30	15	3301	0.0403	40_15_3	40	15	4251	0.0159
30_15_4	30	15	3406	0.0103	40_15_4	40	15	4249	0.0144
30_15_5	30	15	3463	0.0542	40_15_5	40	15	4353	0.1380
30_15_6	30	15	3478	0.0652	40_15_6	40	15	4120	0.0852
30_15_7	30	15	3416	0.0324	40_15_7	40	15	4299	0.1322
30_15_8	30	15	3444	0.0600	40_15_8	40	15	4279	0.0321
30_15_9	30	15	3314	0.0261	40_15_9	40	15	4116	0.1449
30_20_1	30	20	3894	0.0282	40_20_1	40	20	4935	0.1667
30_20_10	30	20	4113	0.1272	40_20_10	40	20	4726	0.0247
30_20_2	30	20	4017	0.0557	40_20_2	40	20	4854	0.0744
30_20_3	30	20	4022	0.0101	40_20_3	40	20	5103	0.1033
30_20_4	30	20	3786	0.0795	40_20_4	40	20	4838	0.0801
30_20_5	30	20	3781	0.0449	40_20_5	40	20	4712	0.0535
30_20_6	30	20	3971	0.0801	40_20_6	40	20	4936	0.1273
30_20_7	30	20	3999	0.0377	40_20_7	40	20	5092	0.1289
30_20_8	30	20	4016	0.0239	40_20_8	40	20	4999	0.0971
30_20_9	30	20	4019	0.0537	40_20_9	40	20	5041	0.0728
30_5_1	30	5	2072	0.0114	40_5_1	40	5	2842	0.0173
30_5_10	30	5	2040	0.0399	40_5_10	40	5	2797	0.0164
30_5_2	30	5	1960	0.0409	40_5_2	40	5	2875	0.0145
30_5_3	30	5	2029	0.0162	40_5_3	40	5	2592	0.0520
30_5_4	30	5	2111	0.0103	40_5_4	40	5	2637	0.0170
30_5_5	30	5	1967	0.0152	40_5_5	40	5	2738	0.0155
30_5_6	30	5	2127	0.0519	40_5_6	40	5	2598	0.0147
30_5_7	30	5	2036	0.0798	40_5_7	40	5	2649	0.0525
30_5_8	30	5	2051	0.0101	40_5_8	40	5	2829	0.0155
30_5_9	30	5	2046	0.0106	40_5_9	40	5	2753	0.0410
40_10_1	40	10	3550	0.0752	50_10_1	50	10	4121	0.0570

Table 5: Results of ATSP Model with lazy constraints for Vallada et al. (2015), Part II

Inst.	n	m	Opt. Sol.(C_{max})	Sol. T.(s)	Inst.	n	m	Opt. Sol.(C_{max})	Sol. T.(s)
50_10_10_	50	10	4268	0.0363	60_10_10	60	10	5040	0.1167
50_10_2_	50	10	4261	0.0559	60_10_2	60	10	5185	0.1935
50_10_3_	50	10	4227	0.0770	60_10_3	60	10	4953	0.0284
50_10_4_	50	10	4320	0.0608	60_10_4	60	10	5006	0.0778
50_10_5_	50	10	4356	0.0601	60_10_5	60	10	5140	0.0972
50_10_6_	50	10	4205	0.0259	60_10_6	60	10	5146	0.0642
50_10_7_	50	10	4096	0.0575	60_10_7	60	10	5130	0.0341
50_10_8_	50	10	4322	0.0929	60_10_8	60	10	4976	0.0295
50_10_9_	50	10	4289	0.0230	60_10_9	60	10	5001	0.1078
50_15_1_	50	15	4972	0.1221	60_15_1	60	15	5972	0.1201
50_15_10_	50	15	5173	0.0910	60_15_10	60	15	6092	0.1767
50_15_2_	50	15	5079	0.1481	60_15_2	60	15	5965	0.1288
50_15_3_	50	15	5136	0.0945	60_15_3	60	15	6070	0.1822
50_15_4_	50	15	5248	0.0983	60_15_4	60	15	5974	0.1249
50_15_5_	50	15	5092	0.1584	60_15_5	60	15	6004	0.1000
50_15_6_	50	15	5194	0.1603	60_15_6	60	15	6149	0.0567
50_15_7_	50	15	5297	0.1591	60_15_7	60	15	6059	0.2027
50_15_8_	50	15	5174	0.3361	60_15_8	60	15	5974	0.1971
50_15_9_	50	15	5096	0.0587	60_15_9	60	15	5760	0.0655
50_20_1_	50	20	5854	0.1319	60_20_1	60	20	6925	0.1755
50_20_10_	50	20	5926	0.1756	60_20_10	60	20	6724	0.3514
50_20_2_	50	20	5825	0.4417	60_20_2	60	20	6928	0.1642
50_20_3_	50	20	5952	0.0881	60_20_3	60	20	7151	0.3448
50_20_4_	50	20	5960	0.0645	60_20_4	60	20	7077	0.1199
50_20_5_	50	20	5893	0.1850	60_20_5	60	20	6699	0.0283
50_20_6_	50	20	6042	0.0245	60_20_6	60	20	6781	0.0868
50_20_7_	50	20	5984	0.1002	60_20_7	60	20	6909	0.0756
50_20_8_	50	20	5906	0.1847	60_20_8	60	20	6871	0.0771
50_20_9_	50	20	5977	0.0780	60_20_9	60	20	6833	0.0323
50_5_1_	50	5	3577	0.0272	60_5_1	60	5	3906	0.0306
50_5_10_	50	5	3372	0.0222	60_5_10	60	5	3980	0.0303
50_5_2_	50	5	3303	0.0386	60_5_2	60	5	3779	0.0294
50_5_3_	50	5	3289	0.0236	60_5_3	60	5	3858	0.0290
50_5_4_	50	5	3391	0.0475	60_5_4	60	5	3900	0.0298
50_5_5_	50	5	3405	0.1161	60_5_5	60	5	3941	0.0422
50_5_6_	50	5	3302	0.0884	60_5_6	60	5	3758	0.0316
50_5_7_	50	5	3088	0.0285	60_5_7	60	5	4001	0.0659
50_5_8_	50	5	3238	0.0470	60_5_8	60	5	4138	0.0774
50_5_9_	50	5	3117	0.0206	60_5_9	60	5	3784	0.0340
60_10_1_	60	10	5067	0.0996	100_20_1	100	20	10441	0.4922
100_20_10	100	20	10495	0.5231	200_40_10	200	40	26723	4.4871
100_20_2	100	20	10617	0.4061	200_40_2	200	40	26434	5.4186
100_20_3	100	20	10693	0.5918	200_40_3	200	40	26320	5.9120
100_20_4	100	20	10622	0.4459	200_40_4	200	40	26576	1.8700
100_20_5	100	20	10762	0.5164	200_40_5	200	40	27038	5.7529
100_20_6	100	20	10544	0.5596	200_40_6	200	40	26586	4.9933
100_20_7	100	20	10875	0.2384	200_40_7	200	40	26555	2.6358
100_20_8	100	20	10640	0.3216	200_40_8	200	40	26844	3.8999
100_20_9	100	20	10549	0.1432	200_40_9	200	40	26487	4.5533
100_40_1	100	40	14968	1.0921	200_60_1	200	60	32175	28.7862
100_40_10	100	40	14490	0.2828	200_60_10	200	60	32134	5.7394
100_40_2	100	40	14761	0.6939	200_60_2	200	60	32140	3.0592
100_40_3	100	40	14599	0.1906	200_60_3	200	60	32091	6.7659
100_40_4	100	40	14651	0.4148	200_60_4	200	60	31886	5.4655
100_40_5	100	40	14737	0.9762	200_60_5	200	60	32242	3.0156
100_40_6	100	40	14470	0.5847	200_60_6	200	60	31902	2.6404
100_40_7	100	40	14894	1.8937	200_60_7	200	60	31793	9.0042
100_40_8	100	40	14807	0.5718	200_60_8	200	60	31745	1.0891
100_40_9	100	40	14778	0.3320	200_60_9	200	60	32162	3.6932
100_60_1	100	60	17851	0.2511	300_20_1	300	20	28476	3.9984
100_60_10	100	60	17831	1.0584	300_20_10	300	20	29154	9.2049
100_60_2	100	60	17887	0.1850	300_20_2	300	20	28583	7.0133
100_60_3	100	60	17786	0.8045	300_20_3	300	20	28623	5.0317
100_60_4	100	60	18030	0.3413	300_20_4	300	20	28742	4.2919
100_60_5	100	60	18123	0.9239	300_20_5	300	20	28749	11.4948
100_60_6	100	60	18167	0.7555	300_20_6	300	20	28811	4.4233
100_60_7	100	60	17984	0.6871	300_20_7	300	20	28574	6.4210
100_60_8	100	60	18191	0.5258	300_20_8	300	20	28734	12.3017
100_60_9	100	60	17810	0.5108	300_20_9	300	20	28591	7.3738
200_20_1	200	20	19731	4.8943	300_40_1	300	40	38247	20.6495
200_20_10	200	20	19798	2.8069	300_40_10	300	40	38250	6.8309
200_20_2	200	20	19768	1.7726	300_40_2	300	40	38450	13.5474
200_20_3	200	20	19895	2.5278	300_40_3	300	40	38028	17.3547
200_20_4	200	20	19624	2.9073	300_40_4	300	40	38270	18.3147
200_20_5	200	20	19500	1.5854	300_40_5	300	40	38511	5.2853
200_20_6	200	20	19878	1.2219	300_40_6	300	40	38477	6.6370
200_20_7	200	20	19619	1.2570	300_40_7	300	40	38274	23.5590
200_20_8	200	20	19850	5.0940	300_40_8	300	40	38196	9.2343
200_20_9	200	20	19551	3.2322	300_40_9	300	40	38026	5.4556
200_40_1	200	40	26652	5.1787	300_60_1	300	60	45767	12.7447

Table 6: Results of ATSP Model with lazy constraints for Vallada et al. (2015), Part III

Inst.	n	m	Opt. Sol.(C_{max})	Sol. T.(s)	Inst.	n	m	Opt. Sol.(C_{max})	Sol. T.(s)
300_60_10	300	60	46245	36.5312	500_20_10	500	20	45754	10.7345
300_60_2	300	60	45455	6.6578	500_20_2	500	20	46646	13.9323
300_60_3	300	60	45622	17.3802	500_20_3	500	20	46489	33.7420
300_60_4	300	60	46023	11.8742	500_20_4	500	20	46187	20.1735
300_60_5	300	60	45763	13.6073	500_20_5	500	20	46517	43.2145
300_60_6	300	60	45936	13.9095	500_20_6	500	20	46171	12.0497
300_60_7	300	60	46563	18.0987	500_20_7	500	20	46503	33.8578
300_60_8	300	60	45932	16.0489	500_20_8	500	20	46377	10.7782
300_60_9	300	60	46112	13.7744	500_20_9	500	20	46323	27.8183
400_20_1	400	20	37222	15.9056	500_40_1	500	40	60765	61.9090
400_20_10	400	20	37735	5.5872	500_40_10	500	40	61274	82.6358
400_20_2	400	20	37693	24.3026	500_40_2	500	40	61655	382.5617
400_20_3	400	20	37482	6.5273	500_40_3	500	40	61557	77.9267
400_20_4	400	20	37329	23.1255	500_40_4	500	40	61180	49.1731
400_20_5	400	20	37520	7.6820	500_40_5	500	40	61746	104.5844
400_20_6	400	20	37433	7.0760	500_40_6	500	40	61060	41.1385
400_20_7	400	20	37748	22.3799	500_40_7	500	40	60982	96.1946
400_20_8	400	20	37657	6.0833	500_40_8	500	40	61772	96.9937
400_20_9	400	20	37452	24.7066	500_40_9	500	40	61725	80.0445
400_40_1	400	40	49529	40.4211	500_60_1	500	60	73039	343.4833
400_40_10	400	40	49789	13.6406	500_60_10	500	60	72458	68.6062
400_40_2	400	40	49565	32.9379	500_60_2	500	60	72660	54.6318
400_40_3	400	40	49555	28.7149	500_60_3	500	60	73038	64.1307
400_40_4	400	40	50155	42.4492	500_60_4	500	60	73211	52.6849
400_40_5	400	40	49884	27.5286	500_60_5	500	60	72498	122.5714
400_40_6	400	40	49759	36.6250	500_60_6	500	60	73448	77.1217
400_40_7	400	40	49989	57.1179	500_60_7	500	60	72735	90.5624
400_40_8	400	40	49747	53.8057	500_60_8	500	60	73479	339.3028
400_40_9	400	40	49875	30.4419	500_60_9	500	60	72443	75.9865
400_60_1	400	60	59650	47.0387	600_20_1	600	20	55209	25.0584
400_60_10	400	60	59537	51.0543	600_20_10	600	20	54530	60.8713
400_60_2	400	60	59530	28.3684	600_20_2	600	20	54776	52.4846
400_60_3	400	60	59583	17.6233	600_20_3	600	20	55247	41.5353
400_60_4	400	60	60001	97.0199	600_20_4	600	20	54825	36.3042
400_60_5	400	60	58865	33.4369	600_20_5	600	20	54911	27.4104
400_60_6	400	60	59605	34.1920	600_20_6	600	20	55181	24.8204
400_60_7	400	60	59235	37.5048	600_20_7	600	20	54747	79.4676
400_60_8	400	60	59245	38.5791	600_20_8	600	20	54868	50.4709
400_60_9	400	60	59784	23.8204	600_20_9	600	20	55177	24.6872
500_20_1	500	20	46305	24.9537	600_40_1	600	40	72374	34.9744
600_40_10	600	40	72324	112.6033	700_60_10	700	60	100481	239.8693
600_40_2	600	40	72497	398.3712	700_60_2	700	60	99288	354.8277
600_40_3	600	40	72353	135.2196	700_60_3	700	60	98604	310.7968
600_40_4	600	40	72648	87.5277	700_60_4	700	60	99206	197.9582
600_40_5	600	40	72471	151.6198	700_60_5	700	60	99327	210.2762
600_40_6	600	40	72535	132.1696	700_60_6	700	60	99394	1010.8812
600_40_7	600	40	72533	161.0076	700_60_7	700	60	98785	124.1283
600_40_8	600	40	72426	163.8628	700_60_8	700	60	99317	417.7565
600_40_9	600	40	73289	101.8947	700_60_9	700	60	99617	790.2098
600_60_1	600	60	86234	144.7652	800_20_1	800	20	72360	83.6110
600_60_10	600	60	86200	238.1480	800_20_10	800	20	71859	73.3596
600_60_2	600	60	86026	162.1529	800_20_2	800	20	72008	138.8617
600_60_3	600	60	86187	507.0948	800_20_3	800	20	72097	107.2073
600_60_4	600	60	86477	441.4304	800_20_4	800	20	71910	240.7324
600_60_5	600	60	86109	277.3638	800_20_5	800	20	72427	119.5517
600_60_6	600	60	86122	117.0849	800_20_6	800	20	72344	32.8788
600_60_7	600	60	85911	182.2481	800_20_7	800	20	71870	114.1674
600_60_8	600	60	85978	238.2083	800_20_8	800	20	71986	223.7060
600_60_9	600	60	87162	98.2268	800_20_9	800	20	71761	54.8931
700_20_1	700	20	63478	76.4080	800_40_1	800	40	94679	298.6301
700_20_10	700	20	63166	122.5675	800_40_10	800	40	94725	286.2419
700_20_2	700	20	63252	68.6318	800_40_2	800	40	94360	668.8346
700_20_3	700	20	63354	18.1820	800_40_3	800	40	94358	412.6403
700_20_4	700	20	63390	23.9787	800_40_4	800	40	94936	439.4173
700_20_5	700	20	63484	66.7537	800_40_5	800	40	95372	242.7374
700_20_6	700	20	63589	73.0495	800_40_6	800	40	94806	339.1765
700_20_7	700	20	63751	37.7304	800_40_7	800	40	94295	1064.2792
700_20_8	700	20	63685	117.7486	800_40_8	800	40	94883	260.7170
700_20_9	700	20	63459	28.7232	800_40_9	800	40	95475	285.5402
700_40_1	700	40	83864	207.2749	800_60_1	800	60	112635	1187.3067
700_40_10	700	40	83550	240.1623	800_60_10	800	60	111427	577.5486
700_40_2	700	40	83773	158.5755	800_60_2	800	60	112306	1319.8452
700_40_3	700	40	83657	857.6574	800_60_3	800	60	111782	289.6952
700_40_4	700	40	84147	623.5697	800_60_4	800	60	112154	300.6308
700_40_5	700	40	83641	276.1906	800_60_5	800	60	112351	382.5399
700_40_6	700	40	83650	220.0862	800_60_6	800	60	112377	353.5371
700_40_7	700	40	83580	259.7814	800_60_7	800	60	112640	1487.4829
700_40_8	700	40	84074	179.3191	800_60_8	800	60	112589	488.8110
700_40_9	700	40	84266	454.2688	800_60_9	800	60	112950	1089.7995

Third test benchmark is Reeves (1994)'s test benchmark. It should be notes that our solution times are shorter than those reported in Lin and Ying (2016a)'s matheuristics' solution times. However, this can be due to computing power employed as well.

Table 7: Solution times for TLC on (Reeves, 1994) benchmark data

Inst. Name	n	m	Optimal Solution (C_{max})	TLC Solution Time(s)
reC01	20	5	1526	**0.0057**
reC03	20	5	1361	**0.0057**
reC05	20	5	1511	**0.0183**
reC07	20	10	2042	**0.0973**
reC09	20	10	2042	**0.0092**
reC11	20	10	1881	**0.0054**
reC13	20	15	2545	**0.0194**
reC15	20	15	2529	**0.0205**
reC17	20	15	2587	**0.0235**
reC19	30	10	2850	**0.0118**
reC21	30	10	2821	**0.0286**
reC23	30	10	2700	**0.0113**
reC25	30	15	3593	**0.0290**
reC27	30	15	3431	**0.0223**
reC29	30	15	3291	**0.0109**
reC31	50	10	4307	**0.0402**
reC33	50	10	4424	**0.0243**
reC35	50	10	4397	**0.0504**
reC37	75	20	8008	**0.1479**
reC39	75	20	8419	**0.1725**

5.3 Comparison of the TLC and the Proposed Heuristic Approach

Our proposed heuristic gives fast near optimal solutions whereas TLC gives exact solutions for the problem. First, we show results of our heuristic on the data from Vallada et al. (2015) in Tables 8-10.

Table 8: Results of Proposed Heuristic Model for Vallada et al. (2015), Part I

Inst.	n	m	C_{max}	Sol. T.(s)	Inst.	n	m	C_{max}	Sol. T.(s)
10_10_1	10	10	1496	0.0594	20_10_10	20	10	2159	0.0084
10_10_10	10	10	1607	0.0061	20_10_2	20	10	2311	0.0082
10_10_2	10	10	1553	0.0057	20_10_3	20	10	2246	0.0094
10_10_3	10	10	1262	0.0054	20_10_4	20	10	2051	0.0100
10_10_4	10	10	1368	0.0057	20_10_5	20	10	2268	0.0080
10_10_5	10	10	1603	0.0053	20_10_6	20	10	2361	0.0090
10_10_6	10	10	1385	0.0044	20_10_7	20	10	2425	0.0091
10_10_7	10	10	1503	0.0067	20_10_8	20	10	2296	0.0099
10_10_8	10	10	1297	0.0043	20_10_9	20	10	2223	0.0091
10_10_9	10	10	1366	0.0044	20_15_1	20	15	3039	0.0068
10_15_1	10	15	1637	0.0048	20_15_10	20	15	2709	0.0087
10_15_10	10	15	1989	0.0044	20_15_2	20	15	2799	0.0086
10_15_2	10	15	2132	0.0045	20_15_3	20	15	2772	0.0083
10_15_3	10	15	1952	0.0049	20_15_4	20	15	2619	0.0086
10_15_4	10	15	1794	0.0058	20_15_5	20	15	2705	0.0070
10_15_5	10	15	1916	0.0059	20_15_6	20	15	2855	0.0098
10_15_6	10	15	1665	0.0044	20_15_7	20	15	2772	0.0071
10_15_7	10	15	1809	0.0045	20_15_8	20	15	2749	0.0077
10_15_8	10	15	1933	0.0037	20_15_9	20	15	2648	0.0084
10_15_9	10	15	1833	0.0056	20_20_1	20	20	3443	0.0058
10_20_1	10	20	2153	0.0042	20_20_10	20	20	3017	0.0085
10_20_10	10	20	1945	0.0047	20_20_2	20	20	3420	0.0081
10_20_2	10	20	2209	0.0062	20_20_3	20	20	2992	0.0071
10_20_3	10	20	2053	0.0053	20_20_4	20	20	3231	0.0089
10_20_4	10	20	2332	0.0059	20_20_5	20	20	3557	0.0083
10_20_5	10	20	2126	0.0044	20_20_6	20	20	3465	0.0095
10_20_6	10	20	2486	0.0039	20_20_7	20	20	3578	0.0070
10_20_7	10	20	2426	0.0041	20_20_8	20	20	3246	0.0085
10_20_8	10	20	2146	0.0042	20_20_9	20	20	3426	0.0085
10_20_9	10	20	2218	0.0048	20_5_1	20	5	1651	0.0074
10_5_1	10	5	831	0.0047	20_5_10	20	5	1602	0.0067
10_5_10	10	5	764	0.0042	20_5_2	20	5	1564	0.0081
10_5_2	10	5	828	0.0056	20_5_3	20	5	1768	0.0083
10_5_3	10	5	1087	0.0040	20_5_4	20	5	1428	0.0086
10_5_4	10	5	802	0.0047	20_5_5	20	5	1675	0.0068
10_5_5	10	5	954	0.0043	20_5_6	20	5	1462	0.0111
10_5_6	10	5	880	0.0040	20_5_7	20	5	1531	0.0071
10_5_7	10	5	1000	0.0038	20_5_8	20	5	1620	0.0068
10_5_8	10	5	875	0.0039	20_5_9	20	5	1652	0.0068
10_5_9	10	5	971	0.0038	30_10_1	30	10	2883	0.0147
30_10_10	30	10	2824	0.0148	40_10_10	40	10	3481	0.0181
30_10_2	30	10	3101	0.0145	40_10_2	40	10	3611	0.0200
30_10_3	30	10	3168	0.0149	40_10_3	40	10	3522	0.0203
30_10_4	30	10	2966	0.0150	40_10_4	40	10	3818	0.0204
30_10_5	30	10	3088	0.0134	40_10_5	40	10	3957	0.0216
30_10_6	30	10	3125	0.0152	40_10_6	40	10	3765	0.0210
30_10_7	30	10	2850	0.0150	40_10_7	40	10	3735	0.0206
30_10_8	30	10	2756	0.0145	40_10_8	40	10	3592	0.0199
30_10_9	30	10	2952	0.0147	40_10_9	40	10	3670	0.0200
30_15_1	30	15	3682	0.0147	40_15_1	40	15	4895	0.0231
30_15_10	30	15	3675	0.0151	40_15_10	40	15	4630	0.0205
30_15_2	30	15	3450	0.0134	40_15_2	40	15	4796	0.0204
30_15_3	30	15	3772	0.0154	40_15_3	40	15	4591	0.0216
30_15_4	30	15	3625	0.0137	40_15_4	40	15	4656	0.0206
30_15_5	30	15	3707	0.0153	40_15_5	40	15	4664	0.0214
30_15_6	30	15	3890	0.0160	40_15_6	40	15	4432	0.0215
30_15_7	30	15	3731	0.0155	40_15_7	40	15	4538	0.0207
30_15_8	30	15	3566	0.0153	40_15_8	40	15	4472	0.0200
30_15_9	30	15	3719	0.0148	40_15_9	40	15	4402	0.0208
30_20_1	30	20	4374	0.0153	40_20_1	40	20	5471	0.0199
30_20_10	30	20	4450	0.0150	40_20_10	40	20	5059	0.0181
30_20_2	30	20	4408	0.0141	40_20_2	40	20	5499	0.0193
30_20_3	30	20	4254	0.0130	40_20_3	40	20	5550	0.0212
30_20_4	30	20	4227	0.0151	40_20_4	40	20	5377	0.0213
30_20_5	30	20	4184	0.0171	40_20_5	40	20	4963	0.0207
30_20_6	30	20	4141	0.0154	40_20_6	40	20	5243	0.0202
30_20_7	30	20	4107	0.0134	40_20_7	40	20	5668	0.0212
30_20_8	30	20	4298	0.0158	40_20_8	40	20	5429	0.0208
30_20_9	30	20	4458	0.0161	40_20_9	40	20	5388	0.0210
30_5_1	30	5	2260	0.0151	40_5_1	40	5	3096	0.0220
30_5_10	30	5	2183	0.0134	40_5_10	40	5	2869	0.0190
30_5_2	30	5	2031	0.0151	40_5_2	40	5	2886	0.0184
30_5_3	30	5	2174	0.0137	40_5_3	40	5	2766	0.0186
30_5_4	30	5	2177	0.0126	40_5_4	40	5	2739	0.0186
30_5_5	30	5	2034	0.0126	40_5_5	40	5	2881	0.0188
30_5_6	30	5	2233	0.0141	40_5_6	40	5	2642	0.0184
30_5_7	30	5	2240	0.0147	40_5_7	40	5	2748	0.0190
30_5_8	30	5	2089	0.0129	40_5_8	40	5	3065	0.0187
30_5_9	30	5	2059	0.0123	40_5_9	40	5	2912	0.0191
40_10_1	40	10	3891	0.0212	50_10_1	50	10	4400	0.0286

Table 9: Results of Proposed Heuristic Model for Vallada et al. (2015), Part II

Inst.	n	m	C_{max}	Sol. T.(s)	Inst.	n	m	C_{max}	Sol. T.(s)
50_10_10	50	10	4633	0.0280	60_10_10	60	10	5258	0.0353
50_10_2	50	10	4610	0.0301	60_10_2	60	10	5649	0.0354
50_10_3	50	10	4413	0.0295	60_10_3	60	10	5268	0.0332
50_10_4	50	10	4774	0.0288	60_10_4	60	10	5238	0.0343
50_10_5	50	10	4663	0.0264	60_10_5	60	10	5524	0.0369
50_10_6	50	10	4375	0.0245	60_10_6	60	10	5458	0.0334
50_10_7	50	10	4435	0.0269	60_10_7	60	10	5603	0.0361
50_10_8	50	10	4546	0.0264	60_10_8	60	10	5148	0.0371
50_10_9	50	10	4695	0.0268	60_10_9	60	10	5120	0.0345
50_15_1	50	15	5395	0.0274	60_15_1	60	15	6291	0.0351
50_15_10	50	15	5299	0.0262	60_15_10	60	15	6641	0.0344
50_15_2	50	15	5410	0.0260	60_15_2	60	15	6325	0.0364
50_15_3	50	15	5368	0.0274	60_15_3	60	15	6393	0.0348
50_15_4	50	15	5546	0.0272	60_15_4	60	15	6344	0.0357
50_15_5	50	15	5366	0.0274	60_15_5	60	15	6344	0.0358
50_15_6	50	15	5526	0.0274	60_15_6	60	15	6599	0.0361
50_15_7	50	15	5518	0.0287	60_15_7	60	15	6576	0.0347
50_15_8	50	15	5787	0.0271	60_15_8	60	15	6466	0.0367
50_15_9	50	15	5311	0.0260	60_15_9	60	15	6193	0.0477
50_20_1	50	20	6276	0.0268	60_20_1	60	20	7344	0.0357
50_20_10	50	20	6321	0.0270	60_20_10	60	20	7470	0.0382
50_20_2	50	20	6223	0.0276	60_20_2	60	20	7280	0.0354
50_20_3	50	20	6550	0.0382	60_20_3	60	20	7907	0.0368
50_20_4	50	20	6387	0.0267	60_20_4	60	20	7796	0.0371
50_20_5	50	20	6332	0.0273	60_20_5	60	20	6954	0.0355
50_20_6	50	20	6583	0.0285	60_20_6	60	20	7042	0.0373
50_20_7	50	20	6670	0.0278	60_20_7	60	20	7231	0.0358
50_20_8	50	20	6365	0.0271	60_20_8	60	20	7173	0.0357
50_20_9	50	20	6266	0.0263	60_20_9	60	20	7264	0.0347
50_5_1	50	5	3833	0.0270	60_5_1	60	5	4118	0.0342
50_5_10	50	5	3625	0.0265	60_5_10	60	5	4194	0.0335
50_5_2	50	5	3436	0.0243	60_5_2	60	5	3863	0.0344
50_5_3	50	5	3428	0.0256	60_5_3	60	5	4003	0.0353
50_5_4	50	5	3511	0.0260	60_5_4	60	5	3985	0.0343
50_5_5	50	5	3679	0.0269	60_5_5	60	5	4076	0.0362
50_5_6	50	5	3514	0.0281	60_5_6	60	5	3860	0.0331
50_5_7	50	5	3116	0.0252	60_5_7	60	5	4084	0.0363
50_5_8	50	5	3455	0.0258	60_5_8	60	5	4287	0.0347
50_5_9	50	5	3391	0.0246	60_5_9	60	5	3810	0.0333
60_10_1	60	10	5322	0.0331	100_20_1	100	20	11057	0.1028
100_20_10	100	20	11024	0.1062	200_40_10	200	40	28109	0.4526
100_20_2	100	20	11140	0.1019	200_40_2	200	40	27710	0.4568
100_20_3	100	20	11175	0.1007	200_40_3	200	40	27283	0.4527
100_20_4	100	20	11066	0.1008	200_40_4	200	40	27689	0.4448
100_20_5	100	20	11439	0.0997	200_40_5	200	40	28057	0.4433
100_20_6	100	20	11286	0.1024	200_40_6	200	40	27641	0.4675
100_20_7	100	20	11566	0.1028	200_40_7	200	40	27979	0.4540
100_20_8	100	20	11044	0.1033	200_40_8	200	40	28343	0.4675
100_20_9	100	20	10990	0.0970	200_40_9	200	40	28121	0.4745
100_40_1	100	40	15690	0.1066	200_60_1	200	60	33802	0.4856
100_40_10	100	40	15520	0.1005	200_60_10	200	60	33194	0.4572
100_40_2	100	40	15597	0.1069	200_60_2	200	60	33433	0.4839
100_40_3	100	40	15207	0.1101	200_60_3	200	60	33592	0.4532
100_40_4	100	40	15908	0.1095	200_60_4	200	60	33204	0.4820
100_40_5	100	40	15438	0.1079	200_60_5	200	60	33887	0.4870
100_40_6	100	40	15151	0.1006	200_60_6	200	60	33328	0.4844
100_40_7	100	40	15807	0.1070	200_60_7	200	60	32998	0.4809
100_40_8	100	40	15498	0.1082	200_60_8	200	60	32705	0.4607
100_40_9	100	40	15370	0.1037	200_60_9	200	60	33165	0.4638
100_60_1	100	60	18848	0.1133	300_20_1	300	20	29172	1.1083
100_60_10	100	60	18850	0.1064	300_20_10	300	20	29878	1.1809
100_60_2	100	60	18718	0.1049	300_20_2	300	20	29072	1.0773
100_60_3	100	60	18613	0.1074	300_20_3	300	20	29409	1.0771
100_60_4	100	60	19224	0.1035	300_20_4	300	20	29496	1.1311
100_60_5	100	60	18878	0.1052	300_20_5	300	20	29344	1.0361
100_60_6	100	60	18940	0.1131	300_20_6	300	20	29451	1.0832
100_60_7	100	60	19561	0.1075	300_20_7	300	20	29413	1.0223
100_60_8	100	60	19503	0.1030	300_20_8	300	20	29484	1.1543
100_60_9	100	60	18617	0.1075	300_20_9	300	20	29347	1.0584
200_20_1	200	20	20242	0.4506	300_40_1	300	40	39882	1.0947
200_20_10	200	20	20435	0.4803	300_40_10	300	40	39520	1.1117
200_20_2	200	20	20688	0.4722	300_40_2	300	40	39492	1.0836
200_20_3	200	20	20493	0.4329	300_40_3	300	40	39605	1.0991
200_20_4	200	20	20226	0.4427	300_40_4	300	40	39070	1.1392
200_20_5	200	20	20443	0.4668	300_40_5	300	40	39738	1.1176
200_20_6	200	20	20637	0.4490	300_40_6	300	40	39803	1.1363
200_20_7	200	20	20028	0.4259	300_40_7	300	40	39060	1.0998
200_20_8	200	20	20812	0.4656	300_40_8	300	40	39415	1.1329
200_20_9	200	20	20287	0.4670	300_40_9	300	40	39224	1.1450
200_40_1	200	40	27431	0.4395	300_60_1	300	60	47157	1.1613

Table 10: Results of Proposed Heuristic Model for Vallada et al. (2015), Part III

Inst.	n	m	C_{max}	Sol. T.(s)	Inst.	n	m	C_{max}	Sol. T.(s)
300_60_10	300	60	47737	1.0940	500_20_10	500	20	46324	6.9926
300_60_2	300	60	46842	1.1743	500_20_2	500	20	47509	6.9280
300_60_3	300	60	47123	1.1080	500_20_3	500	20	47221	6.7847
300_60_4	300	60	47425	1.1880	500_20_4	500	20	46925	6.4294
300_60_5	300	60	47520	1.1193	500_20_5	500	20	47214	6.6699
300_60_6	300	60	47838	1.1423	500_20_6	500	20	46966	6.3893
300_60_7	300	60	48477	1.1379	500_20_7	500	20	47044	6.6410
300_60_8	300	60	47740	1.1591	500_20_8	500	20	46951	6.5515
300_60_9	300	60	47224	1.1010	500_20_9	500	20	47123	6.4323
400_20_1	400	20	37975	4.3100	500_40_1	500	40	61654	6.7036
400_20_10	400	20	38237	4.1665	500_40_10	500	40	62502	6.7265
400_20_2	400	20	38509	4.0833	500_40_2	500	40	62619	6.7446
400_20_3	400	20	38523	4.4309	500_40_3	500	40	63247	6.5812
400_20_4	400	20	38089	4.1132	500_40_4	500	40	62285	6.4129
400_20_5	400	20	38396	4.2777	500_40_5	500	40	63701	6.6410
400_20_6	400	20	38213	4.1169	500_40_6	500	40	62184	6.6210
400_20_7	400	20	38476	4.2537	500_40_7	500	40	62616	6.7354
400_20_8	400	20	38347	4.2208	500_40_8	500	40	62911	6.6041
400_20_9	400	20	38024	4.3519	500_40_9	500	40	63496	6.3866
400_40_1	400	40	50987	4.1908	500_60_1	500	60	75235	6.4379
400_40_10	400	40	50990	4.2029	500_60_10	500	60	74115	6.2898
400_40_2	400	40	50652	4.3841	500_60_2	500	60	74532	6.5719
400_40_3	400	40	50859	4.1237	500_60_3	500	60	74881	6.8926
400_40_4	400	40	51515	4.1029	500_60_4	500	60	75229	6.8896
400_40_5	400	40	50886	4.0925	500_60_5	500	60	74729	6.3457
400_40_6	400	40	50762	4.1493	500_60_6	500	60	75325	7.0117
400_40_7	400	40	51730	4.1625	500_60_7	500	60	74745	6.7962
400_40_8	400	40	50832	4.2678	500_60_8	500	60	75140	6.3398
400_40_9	400	40	51350	4.3762	500_60_9	500	60	73698	6.6241
400_60_1	400	60	61222	4.2996	600_20_1	600	20	56165	10.1764
400_60_10	400	60	61054	4.1194	600_20_10	600	20	55371	10.5967
400_60_2	400	60	60891	4.1501	600_20_2	600	20	55394	9.7712
400_60_3	400	60	61134	4.0682	600_20_3	600	20	55960	10.2103
400_60_4	400	60	62157	4.2090	600_20_4	600	20	55542	9.8333
400_60_5	400	60	60292	4.2800	600_20_5	600	20	55641	10.7677
400_60_6	400	60	60978	4.2428	600_20_6	600	20	55947	9.9333
400_60_7	400	60	61039	4.0284	600_20_7	600	20	55610	9.9280
400_60_8	400	60	61312	4.0380	600_20_8	600	20	55529	9.8601
400_60_9	400	60	61794	4.2708	600_20_9	600	20	55853	10.5330
500_20_1	500	20	46896	6.9716	600_40_1	600	40	74114	10.0492
600_40_10	600	40	74007	9.7876	700_60_10	700	60	102330	13.4835
600_40_2	600	40	74441	10.0151	700_60_2	700	60	101816	13.6031
600_40_3	600	40	73846	10.5918	700_60_3	700	60	101174	15.8413
600_40_4	600	40	74294	10.7029	700_60_4	700	60	100989	14.5319
600_40_5	600	40	74057	9.9686	700_60_5	700	60	101556	14.3186
600_40_6	600	40	74154	10.2432	700_60_6	700	60	101058	14.6149
600_40_7	600	40	74149	10.0361	700_60_7	700	60	101454	16.1012
600_40_8	600	40	73619	10.4043	700_60_8	700	60	101370	14.5528
600_40_9	600	40	74608	10.5365	700_60_9	700	60	102892	14.4627
600_60_1	600	60	87852	10.0275	800_20_1	800	20	73296	18.2061
600_60_10	600	60	87842	10.2524	800_20_10	800	20	72565	16.6353
600_60_2	600	60	88018	9.6369	800_20_2	800	20	72656	19.8142
600_60_3	600	60	88580	10.4614	800_20_3	800	20	72922	18.2744
600_60_4	600	60	88502	9.8733	800_20_4	800	20	72781	17.3335
600_60_5	600	60	87923	9.5705	800_20_5	800	20	73101	16.7227
600_60_6	600	60	88013	10.4509	800_20_6	800	20	73016	16.7912
600_60_7	600	60	88034	10.2641	800_20_7	800	20	72730	18.6417
600_60_8	600	60	88314	9.4993	800_20_8	800	20	73509	18.3125
600_60_9	600	60	90038	9.4714	800_20_9	800	20	72617	17.3480
700_20_1	700	20	64591	13.8908	800_40_1	800	40	96200	16.5939
700_20_10	700	20	64150	14.2029	800_40_10	800	40	96641	20.0378
700_20_2	700	20	64289	15.0375	800_40_2	800	40	95873	16.3577
700_20_3	700	20	64134	14.7401	800_40_3	800	40	95855	17.2959
700_20_4	700	20	64377	15.4262	800_40_4	800	40	96505	18.3081
700_20_5	700	20	64147	13.2447	800_40_5	800	40	97016	17.3505
700_20_6	700	20	64265	16.1402	800_40_6	800	40	96238	17.0215
700_20_7	700	20	64619	15.7435	800_40_7	800	40	95423	17.6554
700_20_8	700	20	64534	15.6129	800_40_8	800	40	96322	17.5836
700_20_9	700	20	64221	15.2170	800_40_9	800	40	97317	19.7387
700_40_1	700	40	85547	14.0068	800_60_1	800	60	114405	17.1968
700_40_10	700	40	84943	13.6344	800_60_10	800	60	113550	17.4674
700_40_2	700	40	85449	13.8760	800_60_2	800	60	114455	17.1839
700_40_3	700	40	84763	14.8998	800_60_3	800	60	113742	16.8768
700_40_4	700	40	85554	14.0054	800_60_4	800	60	114248	18.1745
700_40_5	700	40	84902	14.5340	800_60_5	800	60	114549	19.1990
700_40_6	700	40	85331	14.3361	800_60_6	800	60	114393	19.3204
700_40_7	700	40	84890	14.7301	800_60_7	800	60	114831	18.8887
700_40_8	700	40	85953	13.5706	800_60_8	800	60	114305	16.4865
700_40_9	700	40	85469	14.8205	800_60_9	800	60	114745	20.1657

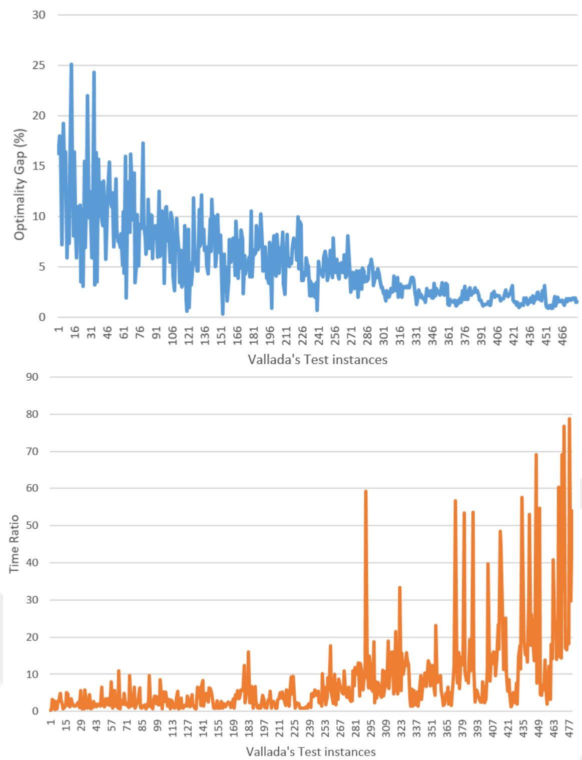

Figure 11: The optimality gap and the time ratio of proposed heuristic on test instances from (Vallada et al., 2015)

A summary of results on the data from (Vallada et al., 2015) can be seen above, where we compute the optimality gap of our proposed heuristic and the time factor. This factor is the ratio of time it takes TLC to find a exact solution to the time it takes to find the near optimal solution using proposed heuristic. It can be seen from Figure 11, the optimality gap changes between 25% and 0.38% and the average optimality gap is 5.33% . The time factor is changes between 78.75% and 0.23% and the average time factor value is 7.97%. These results clearly demonstrate the success of our proposed heuristic approach, especially for large instances.

Table 11 shows solution times of TLC is compared with proposed heuristic's solution times and C_{max} values of the models.

Figure 12 illustrates the makespan and solution times of the TLC and the proposed heuristic together with the optimality gap for the proposed heuristic. Our heuristic clearly

Table 11: Comparison of TLC with the proposed heuristic on test instances from (Reeves, 1994)

Inst. Name	n	m	TLC's (C_{max})	TLC Sol. Time(s)	Proposed H.(C_{max})	Proposed H. Sol. Time(s)
reC01	20	5	1526	**0.0057**	1619	0.0075
reC03	20	5	1361	**0.0057**	1521	0.0081
reC05	20	5	1511	0.0183	1628	**0.0086**
reC07	20	10	2042	0.0973	2204	**0.0083**
reC09	20	10	2042	0.0092	2185	**0.0078**
reC11	20	10	1881	**0.0054**	2060	0.0116
reC13	20	15	2545	0.0194	2626	**0.0083**
reC15	20	15	2529	0.0205	2563	**0.0082**
reC17	20	15	2587	0.0235	2731	**0.0103**
reC19	30	10	2850	**0.0118**	3037	0.0171
reC21	30	10	2821	0.0286	2967	**0.0159**
reC23	30	10	2700	**0.0113**	2875	0.0159
reC25	30	15	3593	0.0290	3834	**0.0146**
reC27	30	15	3431	0.0223	3641	**0.0165**
reC29	30	15	3291	**0.0109**	3494	0.0159
reC31	50	10	4307	0.0402	4514	**0.0305**
reC33	50	10	4424	**0.0243**	4692	0.0300
reC35	50	10	4397	**0.0504**	4619	0.0700
reC37	75	20	8008	0.1479	8422	**0.0735**
reC39	75	20	8419	0.1725	9430	**0.0686**

gives fast solutions with an average optimality gap of 6.5%. Note that the optimality gap is computed using

$$OptGap = \frac{C_{max}^{Heur} - C_{max}^{OPT}}{C_{max}^{OPT}}.$$

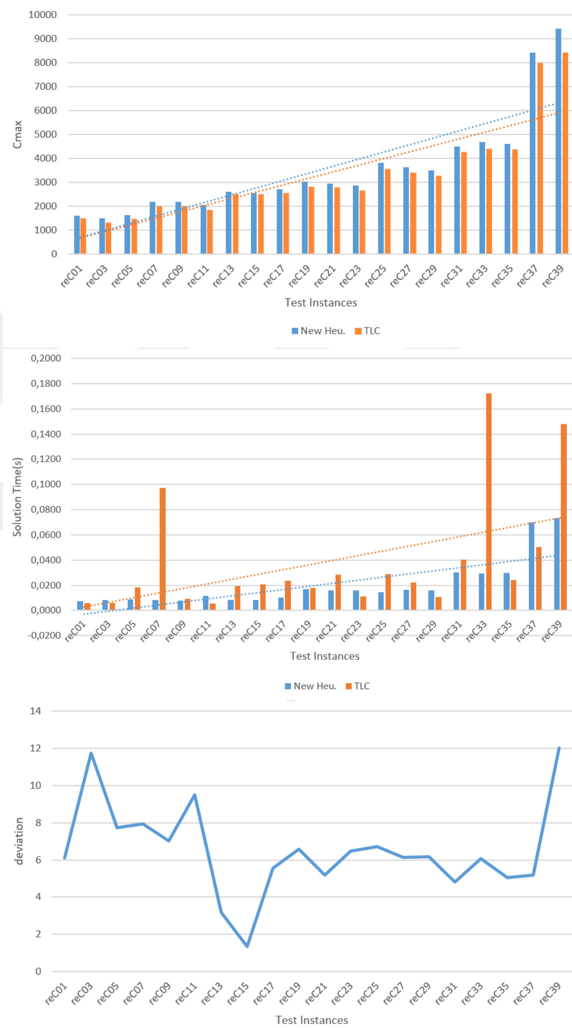

Figure 12: C_{max} and solution time comparison, and the optimality gap of proposed heuristic on test instances from (Reeves, 1994)

CHAPTER VI

CONCLUSION

In this study, we investigate no wait flow shop problem with objective of minimizing makespan. The $F_m|nwt|C_{max}$ problem is studied in many industries. Finding an optimal solution to the this problem is a challenging task. We propose two different solution techniques within the study: ATSP model with lazy constraints and a heuristic model. Their calculation times are competitive; moreover, solution of TLC generates exact solution for the problem. We compare these solutions according to their solution times and their C_{max}. Our TLC technique give exact solution; hence, we measure performance of the proposed solution with TLC's solution data. Optimality gap and time ratio between TLC and proposed heuristic show that proposed heuristic give near optimal solution efficiently and effectively. TLC also give solutions very fast although, it gives exact solutions. We also compare TLC technique results with results of Lin and Ying (2016a). Results show that, these solutions are valuable for the practical systems because of their efficiency and faster responses.

CHAPTER VII

FUTURE RESEARCH

We observe that our TLC solution give exact solution very effectively; however, when it comes larger test instances its solution time performance decreases significantly. Proposed heuristic give near optimal solutions very fast; although, its optimality gap for larger instances is effectively small. Optimality gap of proposed heuristic for small instances can be decreased.

In this thesis, NWFSP for objective minimizing C_{max} is investigated. Other performance criterias can be considered as future research.

Single-machine scheduling problem is investigated in this research. solution approaches can be upgraded for multi-machine systems.

Bibliography

Aldowaisan, T. and Allahverdi, A. (1998). Total flowtime in no-wait flowshops with separated setup times. *Computers & Operations Research*, 25(9):757–765.

Aldowaisan, T. and Allahverdi, A. (2003). New heuristics for no-wait flowshops to minimize makespan. *Computers & Operations Research*, 30(8):1219–1231.

Allahverdi, A. (2016). A survey of scheduling problems with no-wait in process. *European Journal of Operational Research*, 255(3):665–686.

Allahverdi, A. and Aldowaisan, T. (2000). No-wait and separate setup three-machine flowshop with total completion time criterion. *International Transactions in Operational Research*, 7(3):245–264.

Allahverdi, A. and Aldowaisan, T. (2001). Minimizing total completion time in a no-wait flowshop with sequence-dependent additive changeover times. *Journal of the Operational Research Society*, 52(4):449–462.

Allahverdi, A. and Aydilek, H. (2015). The two stage assembly flowshop scheduling problem to minimize total tardiness. *Journal of Intelligent Manufacturing*, 26(2):225–237.

Aydilek, H. and Allahverdi, A. (2012). Heuristics for no-wait flowshops with makespan subject to mean completion time. *Applied Mathematics and Computation*, 219(1):351–359.

Bertolissi, E. (2000). Heuristic algorithm for scheduling in the no-wait flow-shop. *Journal of Materials Processing Technology*, 107(1-3):459–465.

Bianco, L., DellOlmo, P., and Giordani, S. (1999). Flow shop no-wait scheduling with sequence dependent setup times and release dates. *INFOR: Information Systems and Operational Research*, 37(1):3–19.

Bonney, M. and Gundry, S. (1976). Solutions to the constrained flowshop sequencing problem. *Journal of the Operational Research Society*, 27(4):869–883.

Chan, D.-Y. and Bedworth, D. D. (1990). Design of a scheduling system for flexible manufacturing cells. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 28(11):2037–2049.

Dantzig, G., Fulkerson, R., and Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410.

Engin, O. and Güçlü, A. (2018). A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems. *Applied Soft Computing*, 72:166–176.

Espinouse, M.-L., Formanowicz, P., and Penz, B. (1999). Minimizing the makespan in the two-machine no-wait flow-shop with limited machine availability. *Computers & Industrial Engineering*, 37(1-2):497–500.

Fink, A. and Voß, S. (2003). Solving the continuous flow-shop scheduling problem by metaheuristics. *European Journal of Operational Research*, 151(2):400–414.

Ford Jr, L. R. and Fulkerson, D. R. (1958). A suggested computation for maximal multi-commodity network flows. *Management Science*, 5(1):97–101.

Framinan, J. M., Nagano, M. S., and Moccellin, J. V. (2010). An efficient heuristic for total flowtime minimisation in no-wait flowshops. *The International Journal of Advanced Manufacturing Technology*, 46(9-12):1049–1057.

Gangadharan, R. and Rajendran, C. (1993). Heuristic algorithms for scheduling in the no-wait flowshop. *International Journal of Production Economics*, 32(3):285–290.

Gao, K., Pan, Q., Suganthan, P., and Li, J. (2013). Effective heuristics for the no-wait flow shop scheduling problem with total flow time minimization. *The International Journal of Advanced Manufacturing Technology*, 66(9-12):1563–1572.

Glass, C. A., Gupta, J. N., and Potts, C. N. (1999). Two-machine no-wait flow shop scheduling with missing operations. *Mathematics of Operations Research*, 24(4):911–924.

Glover, F. (1996). Ejection chains, reference structures and alternating path methods for traveling salesman problems. *Discrete Applied Mathematics*, 65(1-3):223–253.

Grabowski, J. and Pempera, J. (2005). Some local search algorithms for no-wait flow-shop problem with makespan criterion. *Computers & Operations Research*, 32(8):2197–2212.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of discrete mathematics*, volume 5, pages 287–326. Elsevier.

Hall, N. G. and Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations research*, 44(3):510–525.

Helsgaun, K. (2000a). An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130.

Helsgaun, K. (2000b). An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130.

Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval research logistics quarterly*, 1(1):61–68.

Kanellakis, P.-C. and Papadimitriou, C. H. (1980). Local search for the asymmetric traveling salesman problem. *Operations Research*, 28(5):1086–1099.

King, J. and Spachis, A. (1980). Heuristics for flow-shop scheduling. *International Journal of Production Research*, 18(3):345–357.

Laha, D. and Chakraborty, U. K. (2009). A constructive heuristic for minimizing makespan in no-wait flow shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 41(1-2):97–109.

Laha, D. and Sapkal, S. U. (2011). An efficient heuristic algorithm for m-machine no-wait flow shops. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1. Citeseer.

Li, X. and Wu, C. (2008). Heuristic for no-wait flow shops with makespan minimization based on total idle-time increments. *Science in China Series F: Information Sciences*, 51(7):896.

Lin, S. and Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516.

Lin, S.-W., Lu, C.-C., and Ying, K.-C. (2018). Minimizing the sum of makespan and total weighted tardiness in a no-wait flowshop. *IEEE Access*, 6:78666–78677.

Lin, S.-W. and Ying, K.-C. (2016a). Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics. *Omega*, 64:115–125.

Lin, S.-W. and Ying, K.-C. (2016b). Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics. *Omega*, 64:115–125.

Nagano, M. S. and Miyata, H. H. (2016). Review and classification of constructive heuristics mechanisms for no-wait flow shop problem. *The International Journal of Advanced Manufacturing Technology*, 86(5-8):2161–2174.

Nawaz, M., Enscore Jr, E. E., and Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1):91–95.

Osman, I. H. and Potts, C. (1989). Simulated annealing for permutation flow-shop scheduling. *Omega*, 17(6):551–557.

Pinedo, M. L. (2005). *Planning and scheduling in manufacturing and services*. Springer.

Rajendran, C. (1994). A no-wait flowshop scheduling heuristic to minimize makespan. *Journal of the Operational Research Society*, 45(4):472–478.

Rajendran, C. and Chaudhuri, D. (1990). Heuristic algorithms for continuous flow-shop problem. *Naval Research Logistics (NRL)*, 37(5):695–705.

Rajendran, C. and Ziegler, H. (1997). An efficient heuristic for scheduling in a flowshop to minimize total weighted flowtime of jobs. *European Journal of Operational Research*, 103(1):129–138.

Reeves, C. R. (1994). Genetic algorithms and neighbourhood search. In *AISB Workshop on Evolutionary Computing*, pages 115–130. Springer.

Röck, H. (1984). The three-machine no-wait flow shop is np-complete. *Journal of the ACM (JACM)*, 31(2):336–345.

Ruiz, R. and Allahverdi, A. (2009). New heuristics for no-wait flow shops with a linear combination of makespan and maximum lateness. *International Journal of Production Research*, 47(20):5717–5738.

Ruiz, R. and Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–2049.

Sahni, S. and Gonzalez, T. (1976). P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565.

Torabzadeh, E. and Zandieh, M. (2010). Cloud theory-based simulated annealing approach for scheduling in the two-stage assembly flowshop. *Advances in Engineering Software*, 41(10-11):1238–1243.

Tseng, L.-Y. and Lin, Y.-T. (2010). A hybrid genetic algorithm for no-wait flowshop scheduling problem. *International journal of production economics*, 128(1):144–152.

Vallada, E., Ruiz, R., and Framinan, J. M. (2015). New hard benchmark for flowshop scheduling problems minimising makespan. *European Journal of Operational Research*, 240(3):666–677.

Van der Veen, J. A. and van Dal, R. (1991). Solvable cases of the no-wait flow-shop scheduling problem. *Journal of the Operational Research Society*, 42(11):971–980.

Wismer, D. (1972). Solution of the flowshop-scheduling problem with no intermediate queues. *Operations research*, 20(3):689–697.

VITA

Ahmet Emir Tuzcu received the bachelor's degree in Control And Automation Engineering Department of Istanbul Technical Technical University, Turkey in 2014. After graduation, he started to work in Vestel Electronics Company as a Automation Engineer and now he is working as Automation Specialist in Automation Department. He started to Master of Science Program in Industrial Engineering, Department of Özyeğin University in 2015. He conducted her M.Sc. study under supervision of Assoc. Prof. Erhun Kundakçıoğlu.