

GREEN LOGISTICS APPLICATIONS
IN
TRANSPORTATION AND WAREHOUSING

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

BY

ARSHAM ATASHI KHOEI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

MAY 2021

Approval of the thesis:

**GREEN LOGISTICS APPLICATIONS
IN
TRANSPORTATION AND WAREHOUSING**

submitted by **ARSHAM ATASHI KHOEI** in partial fulfillment of the requirements
for the degree of **Doctor of Philosophy in Industrial Engineering Department,**
Middle East Technical University by,

Prof. Dr. Halil Kalıpçılar Dean, Graduate School of Natural and Applied Sciences	_____
Prof. Dr. Esra Karasakal Head of Department, Industrial Engineering	_____
Prof. Dr. Haldun Süral Supervisor, Industrial Engineering, METU	_____
Assoc. Prof. Dr. Mustafa Kemal Tural Co-supervisor, Industrial Engineering, METU	_____

Examining Committee Members:

Prof. Dr. Ömer Kırca Industrial Engineering, METU	_____
Prof. Dr. Haldun Süral Industrial Engineering, METU	_____
Prof. Dr. Hande Yaman Paternotte Economics and Business, KU Leuven	_____
Prof. Dr. Sinan Gürel Industrial Engineering, METU	_____
Prof. Dr. Ferda Can Çetinkaya Industrial Engineering, Çankaya University	_____

Date: ... / ... /2021



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, Surname: Arsham Atashi Khoei

Signature :

ABSTRACT

GREEN LOGISTICS APPLICATIONS IN TRANSPORTATION AND WAREHOUSING

Atashi Khoei, Arsham

Ph.D., Department of Industrial Engineering

Supervisor: Prof. Dr. Haldun Süral

Co-Supervisor: Assoc. Prof. Dr. Mustafa Kemal Tural

May 2021, 147 pages

Green logistics encompasses the efforts to observe and reduce the environmental impacts of logistics activities. This thesis studies different problems in two subdivisions of green logistics that are green transportation and green warehousing. To address green transportation, we introduce different problems in the context of the planar facility location problem. We first consider an extension of the classical Weber problem, named as the green Weber problem (GWP). The GWP decides on the location of a single facility in the plane and the speeds of the vehicles serving the customers from the facility within the customers' deadlines so as to minimize the total amount of carbon dioxide emitted in the whole distribution system. We also introduce time-dependent congestion on roads which limits the vehicle speeds in different time periods and call the resulting problem as the time-dependent green Weber problem (TD-GWP). We formulate the GWP and TD-GWP as second order cone programming problems both of which can be efficiently solved to optimality. Computational results compare the resulting carbon dioxide emissions of the classical Weber problem with those of the GWP and compare the GWP with the TD-GWP in terms of carbon dioxide emissions

in different traffic congestion patterns. Then, we study the multi-facility green Weber problem (MF-GWP), an extension of the classical multi-facility Weber problem, that determines the locations of p facilities on the plane, $p > 1$, allocations of customers to the facilities, and the speeds of the distribution vehicles so as to minimize the total amount of carbon dioxide emitted from the vehicles. We formulate this problem as a mixed-integer second order cone programming (MISOCP) problem. This formulation turns out to be weak and therefore only small size instances can be solved to optimality within four hours. For larger size instances, a local search heuristic is proposed and some well-known heuristics developed for the multi-facility Weber problem, namely “location-allocation”, “transfer follow-up”, and “decomposition” are adapted for the MF-GWP. The experimental results compare the proposed solution methods for the MF-GWP in terms of solution quality and time. We also investigate how the total amount of carbon dioxide emitted by distribution vehicles changes with respect to the number of facilities located. To address green warehousing, we concentrate on energy efficiency in the material handling systems of warehouses. The order picker forklifts, in recent applications of the material handling systems, provide efficient utilization of the limited storage space by their ability to move in narrow aisles and pick items from high level racks in warehouses. Routing the order picker forklifts to pick ordered items belongs to the operational decision making level and is done in high frequency. We introduce and study the energy minimizing order picker forklift routing problem (EMFRP) which aims to find an energy-efficient route for an order picker forklift to pick a given list of items. A mixed-integer programming formulation and a dynamic programming approach are developed to solve the EMFRP exactly. Since the exact solution approaches are able to solve only small size instances to optimality within a given time limit, we provide some tour construction and tour improvement heuristics for the problem and integrate them into a single solution approach. Computational results show that the proposed solution approach for the EMFRP finds high quality solutions. Moreover, it is observed that significant energy savings can be achieved by solving the EMFRP instead of the classical distance minimization problems.

Keywords: Green transportation, Green warehousing, Continuous location, Order picking problem, Second order cone programming, Time-dependent congestion

ÖZ

DEPOLAMA VE TAŞIMACILIKTA YEŞİL LOJİSTİK UYGULAMALARI

Atashi Khoei, Arsham

Doktora, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Haldun Süral

Ortak Tez Yöneticisi: Doç. Dr. Mustafa Kemal Tural

Mayıs 2021 , 147 sayfa

Yeşil lojistik, lojistik faaliyetlerin çevresel etkilerini gözlemleme ve azaltma çabalarını kapsar. Bu tez yeşil lojistiğin iki alt bölümü olan yeşil taşımacılık ve yeşil depolamada farklı problemleri incelemektedir. Yeşil taşımacılığı ele almak için, düzensel tesis yer seçimi problemi bağlamında farklı problemler ortaya koyuyoruz. İlk başta, yeşil Weber problemi (YWP) olarak adlandırılan klasik Weber probleminin bir uzantısını ele alıyoruz. YWP tüm dağıtım sisteminde salınan toplam karbondioksit miktarını enazlamak için düzlemde tek bir tesisin konumuna ve bu tesisten müşterilerin teslim süreleri içinde onlara hizmet veren araçların hızlarına karar verir. Ayrıca, farklı zaman dilimlerinde araç hızlarını sınırlandıran zamana bağlı trafik yoğunluğunu dikkate alan zamana-bağlı yeşil Weber problemini (ZB-YWP) ele alıyoruz. YWP ve ZB-YWP'nin ikisini de verimli şekilde optimal olarak çözülebilen ikinci dereceden konik programlama olarak formüle ediyoruz. Bilgisayarlı sonuçlar, klasik Weber probleminde ortaya çıkan karbondioksit emisyonlarını YWP'ninkilerle karşılaştırır ve YWP'yi karbondioksit emisyonları açısından farklı trafik sıklığı modeller dahilinde ZB-YWP ile karşılaştırır. Sonra, klasik çok tesisli Weber probleminin bir

uzantısı olan, düzlemde p sayıda, $p > 1$, tesisin konumlarını, müşterilerin atamalarını ve dağıtım araçlarının hızlarını araçlardan yayılan toplam karbondioksit miktarını enazlamak için belirleyen, çok-tesisli yeşil Weber problemini (ÇT-YWP) inceliyoruz. Bu problemi, karma tamsayılı ikinci dereceden konik programlama (KTİDKP) problemi olarak formüle ediyoruz. Bu formülasyonun zayıf olmasından dolayı sadece küçük boyutlu örnekler dört saat içinde optimal olarak çözülebilmektedir. Daha büyük boyutlu örnekler için, yerel arama sezgisel yöntemi geliştirilmiş ve çok tesisli Weber problemi için kullanılan bazı iyi bilinen sezgisel yöntemler, yani "yerseçimi-atama", "transfer takibi" ve "ayırıştırma" ÇT-YWP için uyarlanmıştır. Bilgisayarlı deneyler, ÇT-YWP için önerilen çözüm yöntemleri çözüm kalitesi ve zamanı açısından karşılaştırır. Ayrıca, dağıtım araçlarından kaynaklı toplam karbondioksit salınım miktarının bulunan tesis sayısına göre nasıl değiştiği gözlemlenir. Lojistikte yeşil depolamayı ele almak için depoların malzeme taşıma sistemlerinde enerji verimliliğine odaklanıyoruz. Sipariş toplayıcı forkliftler, dar koridorlarda hareket etme ve depolarda yüksek raflardan ürün alma kabiliyetleri ile, malzeme taşıma sistemlerinin yeni uygulamalarında sınırlı depolama alanından verimli bir şekilde yararlanılmasını sağlar. Sipariş toplayıcı forkliftlerin sipariş edilen ürünleri toplamak için yönlendirilmesi operasyonel karar düzeyine aittir ve yüksek sıklıkta yapılır. Sipariş toplayıcı forkliftin belirli bir ürün listesini toplaması için enerji açısından verimli bir yol bulmayı amaçlayan, enerjiyi enazlayan sipariş toplayıcı forklift rotalama problemini (EEFRP) tanıtlıyor ve inceliyoruz. EEFRP'yi optimal olarak çözmek için bir karma tamsayı programlama formülasyonu ve bir dinamik programlama yaklaşımı geliştirilmiştir. Kesin çözüm yaklaşımları yalnızca küçük boyutlu örnekleri (belirli bir süre içinde) optimal olarak çözebildiğinden dolayı bu problem için bazı tur oluşturma ve tur iyileştirme sezgiselleri öneriyoruz ve bunları tek bir çözüm yaklaşımına entegre ediyoruz. Bilgisayarlı sonuçlar, EEFRP için önerilen çözüm yaklaşımının yüksek kaliteli çözümler bulunduğunu göstermektedir. Ayrıca klasik mesafe enazlayan problemler yerine EEFRP'nin çözülmesinin önemli ölçüde enerji tasarrufu sağlayabileceğini ortaya koyuyoruz.

Anahtar Kelimeler: Yeşil taşımacılık, Yeşil depolama, Düzlemsel yerseçimi, Sipariş toplama problemi, İkinci dereceden konik programlama, Zamana bağlı trafik yoğunluğu



To my family.

ACKNOWLEDGMENTS

I wish to express my deepest appreciation to my supervisor Prof. Dr. Haldun Süral and co-supervisor Assist. Prof. Mustafa Kemal Tural for their guidance, advice, criticism, encouragements, and insight throughout this research. I would also like to extend my gratitude to Prof. Dr. Hande Yaman Paternotte and Prof. Dr. Sinan Gürel for their valuable suggestions and comments.

I would also like to express my special thanks to Dr. Mohammad Saleh Farham for his friendship and endless support.

The author gratefully acknowledges Dr. Francisco Saldanha-da-Gama, the editor-in-chief of *Computers & Operations Research*, who granted us permission to reproduce our published studies in this thesis.

This study was supported by TÜBİTAK (the Scientific and Technological Research Council of Turkey) [grant number 217M486].

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xiv
LIST OF FIGURES	xix
LIST OF ABBREVIATIONS	xxii
CHAPTERS	
1 INTRODUCTION AND LITERATURE REVIEW	1
1.1 Green Weber Problems	1
1.2 Energy Minimizing Order Picker Forklift Routing Problem	10
1.3 The Outline of the Thesis	17
2 TIME-DEPENDENT GREEN WEBER PROBLEM (TD-GWP)	19
2.1 The Green Weber Problem (GWP)	19
2.1.1 Second Order Cone Programming	22
2.1.2 An SOCP Formulation of the GWP	22
2.2 The Time-dependent Green Weber Problem (TD-GWP)	25
2.2.1 The TD-GWP with Non-decreasing Congestion	27

2.3	Illustrative Example	28
2.4	Computational Experiments	33
2.5	Concluding Remarks	38
3	MULTI-FACILITY GREEN WEBER PROBLEM (MF-GWP)	41
3.1	Problem Description and Notation	42
3.2	Solution Approaches for the MF-GWP	43
3.2.1	An MISOCP Formulation for the MF-GWP	43
3.2.2	Heuristics for the MF-GWP	45
3.2.2.1	Alternate Location-allocation Heuristic	46
3.2.2.2	Local Search	47
3.2.2.3	Transfer	49
3.2.2.4	Decomposition	54
3.3	Computational Results	57
3.3.1	Small Size Instances	57
3.3.2	Medium and Large Size Instances	68
3.4	Application Areas of the MF-GWP and an Illustrative Example . . .	80
3.5	Concluding Remarks	86
4	ENERGY MINIMIZING FORKLIFT ROUTING PROBLEM (EMFRP) . .	89
4.1	Notation and Problem Description	90
4.2	Exact Solution Approaches for the EMFRP	93
4.2.1	An MIP Formulation	94
4.2.2	A Dynamic Programming Solution Approach	96
4.3	Heuristic Solution Approaches for the EMFRP	97

4.3.1	Tour Construction	97
4.3.1.1	TSP-based Nearest Neighbor Algorithm	98
4.3.1.2	DP-Construction Algorithm	98
4.3.2	Tour Improvement	99
4.3.2.1	TSP-based Improvement Algorithms	99
4.3.2.2	DP-Improvement Matheuristic	103
4.4	Illustrative Example	105
4.5	Computational Experiments	107
4.5.1	Preliminary Experiments	108
4.5.1.1	Choosing among 2-opt, 3-opt, or 2&3-opt	108
4.5.1.2	Selection of the parameters of the DP-Const algorithm	109
4.5.1.3	Determining the implementation sequence of the two improvement heuristics	112
4.5.1.4	Selection of the parameters of the DP-Imp algorithm	113
4.5.2	Further Computational Experiments	117
4.6	Concluding Remarks	130
5	CONCLUSION	133
5.1	Major Findings	135
5.2	Future Research Directions	136
	REFERENCES	139

LIST OF TABLES

TABLES

Table 1.1	Summary of the reviewed studies	8
Table 2.1	Values of the parameters used in the fuel consumption calculations .	21
Table 2.2	Comparison of the solutions of the WP and the GWP in the illustrative example	31
Table 2.3	Comparisons of the solutions of the GWP and TD-GWP-2; and TD-GWP-2 and TD-GWP-3 in the illustrative example	33
Table 2.4	Comparison of 4 problems with respect to the average total amounts of CO ₂ emission over 5 instances in the 500 customer case	35
Table 2.5	Comparison of 4 problems with respect to the average of the average speeds of the vehicles over 5 instances in different time periods in the 500 customer case	35
Table 2.6	Comparison of 3 problems with respect to the average amounts of CO ₂ emission over 10 instances in the 1000 customer case	36
Table 2.7	Comparison of 3 problems with respect to the average of the average speeds of the vehicles over 10 instances in different time periods in the 1000 customer case under the time limit type TL1	37
Table 2.8	Comparison of 3 problems with respect to the average of the average speeds of the vehicles over 10 instances in different time periods in the 1000 customer case under the time limit type TL2	37

Table 3.1	Notation used in the transfer heuristic	53
Table 3.2	Notation used in the decomposition heuristic	55
Table 3.3	Performance of the MISOCP formulation in solving small size instances with and without the SB constraints	59
Table 3.4	Performance of the MISOCP formulation in solving small size instances under different time limit settings	60
Table 3.5	Performances of the MISOCP formulation, the ALA heuristic, and the LS heuristic with different alternatives	62
Table 3.6	Comparison of the ALA heuristic and alternatives of the LS heuristic in terms of the number of times the best solution is found	63
Table 3.7	Performance of the LS heuristic with different β values for the problem instance with $n = 49$ and $p = 4$	63
Table 3.8	Average computational time spent in different stages over 10 replications for the problem instances with $n = 49$	65
Table 3.9	Objective function values of the best solutions in different stages and average percent improvements obtained with respect to the solution of the previous stage over 10 replications for the problem instances with $n = 49$	66
Table 3.10	Average computational time spent in different stages over 10 replications for the problem instances with $n = 75$	70
Table 3.11	Objective function values of the best solutions in different stages and average percent improvements obtained with respect to the solution of the previous stage over 10 replications for the problem instances with $n = 75$	71
Table 3.12	Average computational time spent in different stages over 10 replications for the problem instances with $n = 287$	75

Table 3.13 Objective function values of the best solutions in different stages and average percent improvements obtained with respect to the solution of the previous stage over 10 replications for the problem instances with $n = 287$	76
Table 3.14 Best Objective function value found by different heuristics for the problem with $n = 75$ and fixed 10 minutes solution time	77
Table 3.15 Best objective function value found by different heuristics for the problem with $n = 287$ and fixed 60 minutes solution time	78
Table 3.16 Coordinates and deadlines of the stations in the illustrative example .	87
Table 3.17 Coordinates of the corners of the depots area in the illustrative example	87
Table 4.1 Performances of the 2-opt, 2&3-opt, and 3-opt algorithms on 5-aisle instances with different n values	109
Table 4.2 Performances of the 2-opt, 2&3-opt, and 3-opt algorithms on 10-aisle instances with different n values	110
Table 4.3 Comparison of quality of solutions generated by DP-Const algorithm with $S = 4$ and $S = 6$ with that of the nearest neighbor algorithm . .	111
Table 4.4 Average percent deviations of the objective function values of the solutions obtained by the first sequence alternative from those of the second one with different DP-Imp parameter settings and n values	113
Table 4.5 Comparisons of the solutions obtained by the 2&3-opt algorithm with those that are obtained after further improvement by the DP-Imp algorithm with $K' - S' : 16 - 4$ parameter setting	114
Table 4.6 Comparisons of the solutions obtained by the 2&3-opt algorithm with those that are obtained after further improvement by the DP-Imp algorithm with $K' - S' : 16 - 6$ parameter setting	115

Table 4.7 Comparisons of the solutions obtained by the 2&3-opt algorithm with those that are obtained after further improvement by the DP-Imp algorithm with $K' - S' : 18 - 4$ parameter setting	115
Table 4.8 Comparisons of the solutions obtained by the 2&3-opt algorithm with those that are obtained after further improvement by the DP-Imp algorithm with $K' - S' : 18 - 6$ parameter setting	116
Table 4.9 Average solution times and relative MIP gaps when the 5-aisle instances with different number of items (n) and height-mass settings are solved using the MIP formulation	118
Table 4.10 Average solution times and relative MIP gaps when the 10-aisle instances with different number of items (n) and height-mass settings are solved using the MIP formulation	119
Table 4.11 Solution time of the DP, its percent deviation from the CPLEX time, and the percent deviation of the objective function values of the DP solutions from those of the CPLEX solution all on average for the 5- and 10-aisle instances with different n values	120
Table 4.12 Solution quality of the proposed solution approach compared to exact approaches in 5-aisle instances with different height-mass settings and different n	121
Table 4.13 Solution quality of the proposed solution approach compared to exact approaches in 10-aisle instances with different height-mass settings and different n	122
Table 4.14 Average proportion of the times the initial solution constructed results in the best solution and average solution times when different construction algorithms are utilized for 5-aisle instances with different n	124
Table 4.15 Average proportion of the times the initial solution constructed results in the best solution and average solution times when different construction algorithms are utilized for 10-aisle instances with different n	125

Table 4.16 Average percent energy savings when EMFRP is used instead of
HVMFRP on 5-aisle instances with different n and height-mass settings . 126

Table 4.17 Average percent energy savings when EMFRP is used instead of
HVMFRP on 10-aisle instances with different n and height-mass settings . 126



LIST OF FIGURES

FIGURES

Figure 1.1	An operator riding the order picker forklift when its fork is lowered (a) or lifted (c) and picking items (b,d)	16
Figure 2.1	Fuel consumption in liter per 100 km as a function of the vehicle speed	21
Figure 2.2	An instance of the Weber problem with four customers	29
Figure 2.3	Locations of the facilities obtained by the GWP for different ℓ_4 values in the illustrative example	30
Figure 2.4	Total amounts of CO ₂ emission of the solutions of the WP and the GWP together with the percent deviations as ℓ_4 changes in the illustrative example	32
Figure 2.5	Locations of the customers in the 500 customer case	34
Figure 3.1	Illustration of the steps of the ALA heuristic followed by the LS heuristic	48
Figure 3.2	Amount of CO ₂ emission of the distribution system with respect to the number of facilities for the problem instances with $n = 49$	67
Figure 3.3	Amount of CO ₂ emission of the distribution system with respect to the number of facilities for the problem instances with $n = 75$	73

Figure 3.4	Histograms of the objective function values of the ALA, transfer, and decomposition solutions for 500 replications for the instance with $p = 10$ of the ruspini data set.	74
Figure 3.5	The layout of the assembly line system	84
Figure 3.6	Solution of the uncapacitated problem	84
Figure 3.7	Solution of the capacitated problem with intersecting rails	85
Figure 3.8	Solution of the capacitated problem with no intersecting rails . .	86
Figure 4.1	The steps of the DP-Construction algorithm, (a) in the 1st step, K closest items to the depot location are taken (b) after the DP algorithm is run, C items are detached from the middle of the incomplete tour, (c) in the second step, detached C items together with new $K - C$ items form the new subset, (d) after the DP algorithm is run, C items are detached from the middle of the incomplete tour, (e) in the third step, detached C items together with new $K - C$ items form the new subset, (f) tour construction is completed.	100
Figure 4.2	The initial tour (a) and a neighboring tour (b) obtained by removing the dashed arcs from (a) in the 2-opt algorithm	101
Figure 4.3	The initial tour (a) and the neighboring tours (b)-(h) obtained by removing the dashed arcs from (a) in the 3-opt algorithm	102
Figure 4.4	The steps of the DP-Improvement algorithm, (a) a given initial tour, (b) first subset is formed by selecting the first K' picked items, (c) the DP is run to optimally reinsert the selected items, (d) the selected subset is rotated by $K' - C'$ items to form the second subset, (e) the DP is run to optimally reinsert the selected items in the second subset, (f) the selected subset is rotated by $K' - C'$ items to form the third subset, (g) fifth subset is formed, (h) last subset is formed, (i) the solution obtained at the end of the first iteration	104

Figure 4.5	(a) an instance with 14 items, (b) solution of the horizontal distance minimization problem, (c) solution of the horizontal + vertical distance minimization problem, (d) solution of the EMFRP	106
Figure 4.6	Average energy consumption in kilo joule (in vertical axes) when the HVMFRP and EMFRP are used to solve the 5-aisle instances with different height-mass settings and different number of items (in horizontal axes)	128
Figure 4.7	Average percent change (in vertical axes) in total horizontal and vertical distances (D), total travel times (T), and total energy consumptions (E) of the forklift when 5-aisle instances with different height-mass settings and different number of items (in horizontal axes) are solved with the EMFRP instead of the HVMFRP	129
Figure 4.8	Average percent change (in vertical axes) in horizontal distances (H), vertical distances (V), and total horizontal and vertical distances (D) of the forklift when the instances with different height-mass settings and different number of items (in horizontal axes) are solved with the EMFRP instead of the HVMFRP	131

LIST OF ABBREVIATIONS

ABBREVIATIONS

ACO	Ant-colony optimization
ALA	Alternate location-allocation
ALNS	Adaptive large neighborhood search
CMEM	Comprehensive modal emission model
CO ₂	Carbon dioxide
CVRP	Capacitated vehicle routing problem
Disc.	Discretized
EMFRP	Energy minimizing order picker forklift routing problem
EMVRP	Energy minimizing vehicle routing problem
FCR	Fuel consumption rate
FW	Floyed-Warshall
G	General
GHG	Greenhouse gas
GW	Green warehousing
GWP	Green Weber problem
H	High
HVDMFRP	Horizontal+vertical distance minimization forklift routing problem
LP	Linear programming
LPG	Liquid petroleum gas
LS	Local search

L	Low
M	Medium
Meta	Metaheuristic
MF-GWP	Multi-facility green Weber problem
MILP	Mixed-integer linear programming
MINLP	Mixed-integer nonlinear programming
MIP	Mixed-integer programming
MISOCP	Mixed-integer second order cone programming
OFV	Objective function value
Param	Parameter
PRP	Pollution routing problem
RGV	Rail-guided vehicle
SA	Simulated annealing
SB	Symmetry breaking
SCN	Supply chain network
SO	Speed optimization
SOC	Second order cone programming
TD-GWP	Time-dependent green Weber problem
TS	Tabu search
TSP	Traveling salesman problem
Var	Variable
VRPTW	Vehicle routing problem with time windows
WP	Weber problem



CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

1.1 Green Weber Problems

Distribution and transportation related activities are growing all over the world, especially in urban areas, due to a continuous increase in the demand of goods and services (The Population Division of the Department of Economic and Social Affairs of the United Nations 2014). However, sending vehicles to customers and delivering/picking goods in distribution and transportation logistics result in a significant amount of CO₂ emissions affecting citizens' quality of life and the climate. Thus, the importance of green freight transportation in city logistics has been growing to reduce the harmful effects of CO₂ emission (Demir et al. 2014b).

Logistics activities conducted all over the world account for 5.5% – 13% of overall greenhouse gas (GHG) emissions (World Economic Forum (2009)) (90% of this is due to transportation activities). These emissions result from energy consumption due to inter-facility distribution activities as well as intra-facility activities such as material handling and storage activities. Transportation is a large contributor to global emissions of CO₂ and accounts for 23% of all CO₂ emissions from fossil fuel combustion. Besides, 40% of road transportation emissions is due to freight transportation (OECD (2010)).

Green logistics encompasses the efforts to observe and reduce the environmental impacts of logistics activities (Piecyk et al. 2015). Because of the global air pollution and climate change, interest in sustainability and green logistics has been growing and studies on the reduction of GHG emissions resulting from logistics activities in distribution systems have become of great value. In particular, the importance of

taking environmental considerations into account when deciding on facilities' locations and designing distribution systems has been increasing (Dukkanci et al. 2019). In this thesis, we introduce extensions of the Weber problem so that CO₂ emission amounts are managed by setting the vehicle speeds during the delivery operations of a distribution system.

The Weber problem corresponds to locating a single facility or a number of facilities on the plane so as to minimize the sum of the weighted Euclidean distances between the facilities and the customers (Drezner & Hamacher 2002). It is assumed that it is always possible to go directly from the facilities to any customer. The applications of the Weber problems can be exemplified by locating warehouses or facilities for a distribution system in which the demands are delivered directly to the customers. The delivery activities in such distribution systems result in a large amount of energy and fuel consumption and increase the emissions of greenhouse gases (GHG).

A standard approach for solving the single- facility Weber problem is the *Weiszfeld method* (Weiszfeld & Plastria 2009) which is a simple closed form iterative formula. Several studies investigate the convergence of the Weiszfeld method, (see, for example, Chandrasekaran & Tamir 1989, Katz 1974, Kuhn 1973), and a number of modifications have been developed, (see, for example, Vardi & Zhang 2001), since its first introduction by Weiszfeld in 1937.

The multi-facility Weber problem, also referred as the planar p -median problem (Brimberg & Drezner 2013) and the multi-source Weber problem (Brimberg et al. 2000), is a classical facility location problem in the literature. It aims to locate a fixed number of facilities on the plane, allocate customers to the facilities so as to minimize a weighted sum of Euclidean distances between the facilities and the allocated customers. Several solution approaches are proposed for this problem (see, Brimberg et al. 2008, for a review). In the literature, high quality solutions are obtained by different approaches such as the ones proposed by Brimberg & Drezner (2013), Cooper (1964), Drezner et al. (2015), Drezner et al. (2016), and Drezner & Salhi (2017). The "location-allocation" heuristic proposed by Cooper (1964) is a main ingredient in several solution approaches proposed for the multi-facility Weber problem and several other location-allocation problems. Brimberg & Drezner (2013) propose the so

called “transfer follow-up” heuristic for the multi-facility Weber problem. In (Drezner et al. 2015), the solution approaches are based on the heuristics previously proposed in the literature as well as the newly developed variable neighborhood search and genetic metaheuristics. In (Drezner et al. 2016) propose a constructive heuristic to find good starting solutions, a “decomposition” approach, and a neighborhood search algorithm based on limited distance median problem. In (Drezner & Salhi 2017), efficient neighborhood reduction schemes are proposed to be used within the heuristics and metaheuristics proposed in the literature.

In this thesis, we consider several extensions of the classical Weber problems that aim to minimize the total amount of CO₂ emission originating from the distribution activities. First, we consider an extension of the single-facility Weber problem, called as the green Weber problem (GWP), which in addition to the location of the single facility also determines the speeds of the vehicles serving the customers from the facility so as to minimize the total amount of CO₂ emitted in the whole distribution system. It is assumed that the customers have deadlines and the vehicles serving a customer must arrive at the location of the customer no later than its deadline. We also consider a time-dependent version of the problem, named as the time-dependent green Weber problem (TD-GWP), in which the vehicle speeds are limited due to congestion. Moreover, we consider an extension of the classical multi-facility Weber problem, the multi-facility green Weber problem (MF-GWP), that determines the locations of a fixed number of facilities on the plane and the speeds of the vehicles serving the customers within their deadlines so as to minimize the total amount of CO₂ emission originating from the distribution vehicles. The solutions of the GWP, TD-GWP and MF-GWP can be used for the strategic level decision of locating facilities for a distribution system prior to making tactical and operational level decisions such as how to route the vehicles. To the best of our knowledge, such extensions of the Weber problem have not been considered before in the literature. Most of the studies in the literature related with carbon (or GHG) emission issues in distribution systems assume that the facility locations are fixed and mainly deal with tactical and operational level decisions.

Related studies in the literature can be broadly classified into two groups in terms of the objective function used. In the first group, the objective (or a part of the objective)

is the minimization of the amount of energy consumed by the distribution vehicles, where energy consumption of a vehicle is computed as the load of the vehicle multiplied with the distance traveled. In the second group, the objective function to be minimized includes the amount of CO₂ emitted by the vehicles. The CO₂ emission in several studies is calculated by one of the available vehicle emission models (see e.g., Demir et al. 2011) directly or after some simplification, e.g., approximation or discretization. Another classification of the studies in this area can be done based on the decision-making level, where some studies include strategic level decisions such as the determination of the facility locations and some others include tactical and operational level decisions such as fleet sizing and routing decisions.

In one of the early studies on vehicle routing in the context of green logistics, Kara et al. (2007) introduce the energy minimizing vehicle routing problem (EMVRP), where the objective is to minimize the total amount of energy consumption of the capacitated vehicles. They propose a mixed integer programming formulation for the EMVRP. Xiao et al. (2012) develop a fuel consumption rate (FCR) formulation to be used in the objective function of the capacitated vehicle routing problem (CVRP). Their fuel consumption formulation, similar to the one in (Kara et al. 2007), is dependent on the distance traveled and the vehicle load. They propose a simulated annealing metaheuristic with problem-based modifications which turns out to be efficient on CVRP benchmark instances.

Bektaş & Laporte (2011) introduce an extension of the vehicle routing problem with time windows (VRPTW) named as the pollution routing problem (PRP) in which a part of the objective function is the minimization of the total amount of fuel emission (multiplied with some constant) of the vehicles. They use the comprehensive modal emission model (CMEM) provided by Barth et al. (2005) in emission calculations according to which the fuel consumption of a distribution vehicle depends on a number of factors such as speed, distance traveled, curb weight, load, and vehicle and road characteristics. The authors ignore a part of the fuel consumption formula for simplification purposes which may result in inaccurate emission values for low speed levels. They provide a mixed integer linear programming formulation based on the discretization of the speeds and solve instances with up to 20 customers within three hours. Demir et al. (2012) also consider the PRP and propose an adaptive large

neighborhood search (ALNS) algorithm for its solution along with a speed optimization algorithm. In another study (Demir et al. 2014a), the same authors consider a bi-objective version of the PRP, where the first objective is the minimization of the total amount of fuel emission and the second one is the minimization of the drivers' wages and propose an ALNS algorithm as a solution approach. Saka (2013) provides a mixed integer second order cone programming formulation for the PRP for the first time with continuous speed variables. This formulation also, as reported, can only solve small size instances to optimality in the given time limits. A matheuristic approach is proposed by Kramer et al. (2015) for the PRP in which a local search heuristic, speed optimization algorithm, and optimization over a set partitioning formulation are integrated.

In the literature, only a few studies involving strategic level decisions (e.g., location decisions) exist that take the emission of the distribution vehicles into consideration. As the most relevant studies in the literature, we can refer to the ones in the contexts of location routing problem and facility location problem. Chen et al. (2018) consider two objectives in the location-routing problem with full vehicle loads. The first one sums up the total fixed vehicle and facility opening costs and variable operational costs. The second one encompasses the total CO₂ emission associated with transportation activities, opening facilities, and demand processing activities in facilities. The bi-objective problem is solved by the well-known NSGA-II evolutionary algorithm enhanced with the tabu search heuristic. Speed decisions are ignored and no explicit emission model from the literature is used in this study.

Koç et al. (2016) investigate the impact of depot location, fleet composition, and routing on emissions in city logistics. They use the CMEM in emission calculations and consider different zones with different speed limits for the vehicles. The objective function contains the fixed facility opening cost, the variable traveling cost, and the fuel emission cost which depends on different speed zones the vehicles travel. In the provided formulation for the exact solution approach, the emission of a vehicle traveling between two nodes are implicitly given to the model and no zone information is explicitly stated. An ALNS solution approach combined with the simulated annealing metaheuristic and a cheapest path (in terms of cost) calculation heuristic is developed. The cheapest path heuristic is proposed to overcome the difficulty of finding the path

with the lowest cost between two nodes which depends on the vehicle type, its load, and the distance traveled within each speed zone.

The green capacitated location routing problem is considered with two objectives by Toro et al. (2017). In this study, one of the objective functions minimizes the economic aspects in the problem including fixed facility costs and variable traveling costs. The other one minimizes the emissions of the vehicles. In the emission calculations, the speeds of the vehicles are assumed to be constant and consequently the corresponding objective function depends on the vehicles' traveled distances and loads. The epsilon constraint method is used to solve the bi-objective problem where the economic objective is posed as a constraint.

A capacitated facility location problem with two objectives is studied by Harris et al. (2014). The objectives minimize the total facility fixed costs and variable allocation costs, and the total CO₂ emissions due to running facilities and transportation activities. The amount of CO₂ emitted due to any potential facility and any allocation of customers to facilities are predefined as parameters. The authors use a hybrid evolutionary multi-objective algorithm as the solution approach. Xifeng et al. (2013) study a multi-objective uncapacitated facility location problem with fixed costs for the facilities on a network. The objectives in this study are: 1) minimizing total cost, i.e., the sum of fixed facility costs and variable allocation costs, 2) maximizing the minimum customer service reliability which is dependent on the customer's deadline, its distance from the facility, and the speed of the vehicle, and 3) minimizing the total CO₂ emission. The effect of vehicle's speed on the CO₂ emission is ignored and the emission of a vehicle is computed by considering the emission for empty and fully loaded vehicles (both of which do not depend on the speed of the vehicle) and using the load over capacity fraction and the distance. After transforming multiple objectives into a single one by taking all but the emission objectives into the constraints, the authors use a greedy heuristic as the solution approach. The authors provide a set of solutions depicting the trade-offs between the total cost of the distribution system and the other objectives.

We provide a summary of the studies with emission or energy considerations reviewed in this section of the thesis in Table 1.1. In this table, the column titled as "Ref." lists

the studies' references. The columns titled as "Obj. Fun." and "Emis. Model" explain the type of the objective function used and the emission model considered in the study, respectively. The next two columns show whether or not the location and routing decisions are made in the problem, respectively. The column titled as "Traffic Cong." indicate whether traffic congestion limiting the speed of vehicles is considered in the study or not. The next three columns specify whether the load, speed, and distance are taken as parameters or variables, respectively. The columns "Formul." and "Sol. App." detail the type of the mathematical programming formulation and type of the solution approach provided for the considered problem in the study. The last column of the table gives the number of facilities to be located in the considered problem.

There are a few studies involving only location (but no routing) decisions in the literature dealing with emission minimization. These studies, however, do not use any comprehensive emission model from the literature and do not take the effects of the vehicles' speed on emissions into account. The routing problems involving emission considerations studied in the literature, on the other hand, are already hard to solve with the fixed facility locations. Therefore, as one of the first green logistics studies that makes facility location decisions, this thesis does not consider routing decisions. Instead, we assume that the vehicles are sent directly to the customers as in the classical Weber problems.

For the GWP and the TD-GWP, we propose second order cone programming formulations for their solution and show that both problems are polynomial-time solvable. The MF-GWP, on the other hand, is an NP-hard problem and is much more difficult to solve. Due to the Euclidean distances and the emission model used, the MF-GWP is a nonlinear optimization problem. In this thesis, we first propose a mixed integer second order cone programming (MISOCP) formulation for the MF-GWP. As this formulation is weak, we try to strengthen it by adding some symmetry breaking constraints. Still, only small size instances are solved to optimality by the strengthened MISOCP formulation within four hours. To be able to solve larger-size instances, we propose a local search heuristic, and adapt "location-allocation", "transfer follow-up", and "decomposition" heuristics from the literature to the studied problem. In the heuristics, the SOCP formulation of the single facility case and the MISOCP formulation of the multiple facility case are used as subproblems.

Table 1.1: Summary of the reviewed studies

Ref.	Obj. Fun.	Emis. Model	Loc.	R.	Traffic Cong.	Load	Speed	Dist.	Formul.	Sol. App.	# Fac.
Kara et al. (2007)	Load $\times$ Distance			✓		Var	Param	Param	MILP	Exact (up to 16 nodes)	-
Xiao et al. (2012)	Vehicle Cost + Fuel Cost	FCR		✓		Var	Param	Param	MILP	Meta (SA)	-
Bektaş & Laporte (2011)	Fuel and Emission Cost + Driver Cost	CMEM		✓		Var	Var (Disc.)	Param	MILP	Exact for Discretized Formulation (up to 20 nodes)	-
Demir et al. (2012)	Fuel and Emission Cost + Driver Cost	CMEM		✓		Var	Var (Disc.)	Param	MILP	Meta (ALNS) for Discretized Formulation	-
Demir et al. (2014a)	Fuel and Emission Cost & Driver Cost	CMEM		✓		Var	Var (Disc.)	Param	MILP	Meta (ALNS) for Bi-objective and SO	-
Kramer et al. (2015)	Fuel and Emission Cost & Driver Cost	CMEM		✓		Var	Var	Param	MILP	Matheuristic (LS & SO & MILP)	-
Chen et al. (2018)	Total SCN Cost & Emission		✓	✓		Var	Param	Param	MILP	NSGA-II & TS	Multi
Koç et al. (2016)	Depot and Vehicle Cost + Fuel and Emission Cost	CMEM	✓	✓	✓	Var	Var	Param	MINLP	ALNS	Multi
Toro et al. (2017)	Total SCN Cost & Load $\times$ Distance		✓	✓		Var	Param	Param	MILP	Epsilon Constraint for Bi-objective (up to 30 nodes)	Multi
Harris et al. (2014)	Location and Allocation Cost & Location and Allocation Emission		✓				Param	Param	MILP	Hybrid Evolutionary for Bi-objective	Multi
Xifeng et al. (2013)	Cost & Customer Service & Load $\times$ Distance		✓			Param	Param	Param	MILP	Epsilon Constraint for Multi-objective	Multi
Atashi Khoei et al. (2017)	Emission	CMEM	✓		✓	Param	Var	Var	SOCP	Exact (1000 nodes in a couple of seconds)	Single
Atashi Khoei et al. (2020)	Emission	CMEM	✓			Param	Var	Var	MISOCP	Exact & Hybrid Heuristics	Multi

*SCN : Supply Chain Network, Var : Variable, Param : Parameter, Disc. : Discretized, MILP: Mixed-integer Linear Program, MINLP: Mixed-integer Nonlinear Program, SOCP: Second Order Cone Program, MISOCP: Mixed-integer Second Order Cone Program, Meta : Metaheuristic, SA : Simulated Annealing, SO : Speed Optimization, TS : Tabu Search

The contributions of this study are as follows. First, different versions of green Weber problems are considered which may find uses in areas where the classical Weber problems are applicable and emission or energy consumption considerations are prevalent. The multi-facility Weber problem is a well-known problem in the literature studied for many years without taking the CO₂ emission or energy consumption point of view. This study is one of the first studies dealing with emission minimization in the context of a facility location problem. In the following paragraph, several application areas of the MF-GWP are exemplified where emission or energy consumption considerations can be integrated into planar facility location problems. An application area is detailed in Section 3.4 as an illustrative example within the context of an assembly line system where the stations are fed by dedicated rail-guided vehicles. Second, the MF-GWP is formulated as an MISOCP problem without any approximation of the emission model and without any discretization of the speed values. Most of the studies in the literature that use the emission model considered in this study approximate the emission model by ignoring one of its terms and discretize the speed values to handle the nonlinearity, (see, for example, Bektaş & Laporte 2011, Demir et al. 2014a). Third, a local search method is proposed for the MF-GWP as a matheuristic which employs a strengthened and reduced-size MISOCP formulation by restricting the possible locations of the facilities, fixing some binary variables, and reducing the big-M values used in the MISOCP formulation. Fourth, some well-known heuristics developed for the multi-facility Weber problem are adapted for the MF-GWP. The adapted versions are all matheuristics and use SOCP techniques. However, the original versions of the heuristics employ different techniques and do not directly solve nonlinear optimization problems using off-the-shelf solvers. We give some examples of our contributions when adapting these well-known heuristics. In the transfer heuristic, when dealing with the empty facilities and selecting customers that are to be transferred, we take the emission amounts into account. In the decomposition heuristic, we apply the proposed local search method to solve the resulting smaller-size problems. Finally, we compare all solution methods proposed for the MF-GWP in terms of solution quality and computational time and show that within a fixed computational time, even though the location-allocation heuristic is able to make more replications, the improvement heuristics considered, i.e., transfer or transfer followed by decomposition, usually find better solutions even with less number of replications.

The applicability of this study is beyond this particular setting where delivery is carried out by trucks. For example, in aviation, emission models similar to the one used in this study, (see e.g., Aktürk et al. 2014, Senzig & Cumper 2013), exist and solution approaches proposed here or their extensions/modifications can be utilized in locating facilities to serve customers with helicopters through direct shipments or finding the locations of launch points where drones make direct deliveries to the customers. Moreover, energy consumption functions of different types of robots consist of terms that are similar to the terms of the fuel consumption formulation used in this study (Gürel et al. 2019, Tokekar et al. 2011). Therefore the solution methods proposed in this paper can be used, for example, to determine meeting location(s) for a group of autonomous mobile robots for battery replacement or recharging by tanker(s) (Zebrowski et al. 2007). In this case, all mobile robots have the same fixed or variable deadline and their speeds can be controlled to ensure arrival to the location on or before the deadline and to optimize energy consumptions. Moreover, in assembly-line design, where the feeding of stations are done directly via rail-guided vehicles (RGVs) from depots, one may minimize the energy consumption of RGVs by controlling their speed. For this problem, the feeding of a station is done periodically within a given deadline by a designated RGV. We will show an illustrative example about this problem in Section 3.4. Moreover, we will argue in the same section that several emission or energy consumption formulations employed in the literature are “similar” to the one used in this study. Therefore the solution approaches proposed herein can be employed directly or after some problem specific modifications in related planar facility location problems where deliveries are made by direct shipments.

1.2 Energy Minimizing Order Picker Forklift Routing Problem

Warehousing, as an important segment of logistics, incorporates several technology-driven operations such as material handling and packaging. In nowadays technology of material handling systems, picking the orders or materials in some warehouses are done by high-tech order picker forklifts. The order picker forklifts are used in picker-to-part material handling systems and in applications where pallet trucks, rolling ladders, and other piece-picking methods were traditionally used. The order

picker forklifts with their maneuverability in very narrow aisles and high level picking capabilities enable the warehousing systems to utilize the limited storage space more efficiently. Moreover, since an operator performs the picking tasks on the forklift, there is no need for any additional equipment or operator. These technology-driven advantages of the order picker forklifts make them widely used in industries that aim sustaining efficiency in warehouses.

Routing the order picker forklifts in warehouses to pick the listed orders is an operational decision which may be done several times a day. The high frequency of this job in warehouses results in high energy consumption. Performing the order picking tasks in an energy-efficient manner can yield a significant reduction in the energy consumption in warehouses. This will in turn reduce the CO₂ emission resulting from warehousing operations and hence promote sustainability in all dimensions, i.e., environmental, social, and economical. In this paper, given a list of items to be picked, we aim to find an energy-efficient route for an order picker forklift. To the best of our knowledge, there is no study in the literature with explicit evaluation of the energy consumption of the order picker forklifts.

The logistics industry accounts for 13% of all greenhouse gas emissions worldwide (World Economic Forum (2016)). Most of the emissions (and costs) for manufacturing companies are due to supply chain activities with transportation having the largest share (Waltho et al. 2019). In recent years, there has been a significant increase in the number of studies that aim to reduce the CO₂ emission originating from transportation activities in logistics (Bektaş et al. 2019). In addition to CO₂ emission resulting from transportation between facilities or between facilities and customers, the energy consumption inside the facilities also accounts for a huge amount of CO₂ emission (Rüdiger et al. 2016). Warehousing operations are among the main contributors to the intra-facility energy consumption. Of the greenhouse gas emissions produced by the logistics industry, warehousing activities has a share of 11% (Bartolini et al. 2019). In this paper, our focus is on green warehousing (GW) which refers to all efforts used to reduce the impacts of warehousing operations on the environment and the society. In recent studies on GW, energy saving methods, environmental impacts of warehouses, and green warehousing management are the most considered topics (Bartolini et al. 2019). The sources of energy consumption in warehouses are discussed by Fichtinger

et al. (2015). Material handling, as an intra-facility activity, accounts for a significant proportion of energy consumption in warehouses (Freis et al. 2016). Therefore, the environmental assessment of material handling systems along with other relevant processes can provide significant sustainability benefits.

Considering different material handling equipment used in warehouses, e.g., conveyors, automated guided vehicles, and forklifts, most of the energy is consumed by forklifts (Anand et al. 2014). Facchini et al. (2015) develop a decision support tool that takes the forklift type (internal combustion engine or electric motor) and the storage configuration (stackable units or with storage racks) of a warehouse into account to minimize the CO₂ emission resulting from order picking operations. The study measures the energy consumption due to lifting, lowering, and transporting movements of each forklift type to collect the listed orders in corresponding storage configuration. Note that no routing decision is made in this study. The authors only provide some calculations based on average values to suggest the forklift type for each storage configuration ignoring the effects of acceleration, deceleration, and friction forces.

A simulation approach is developed by Facchini et al. (2016) to select the best forklift type among the liquid petroleum gas (LPG) and electric forklifts in terms of emissions. In this simulation model, greenhouse gas emissions are evaluated based on the energy consumption considering the technical characteristics of each forklift type. The energy consumption of the forklifts is measured, without any routing optimization, according to the average horizontal and vertical movements and the average time spent for material handling. Using simulation on a numerical example where the weights of the listed orders and the characteristics and capacities of the forklifts are known, the authors observe that electric forklifts should be preferred with low- and medium-weight orders for a lower carbon emission. Note that the forklifts used in the two studies discussed above are not the order picker forklifts that we consider herein.

The order picking problem deals with finding the best route of an order picker that starts from a depot location, picks the ordered items, and brings them to the depot location. In the order picking problem literature, the target of most of the models is not to improve the environmental performance of the system (e.g., carbon footprint or CO₂ emissions) but to reduce the travel distance and / or time (Facchini et al. 2015,

Cortés et al. 2017). In general, the order picking problem is formulated as the traveling salesman problem (TSP) in the literature. In some of the studies, the trade-off between the energy consumption and travel time is assessed. For example, Makris et al. (2006) develop a TSP-based routing algorithm to analyze the trade-off between energy consumption and travel time of the order pickers. They, however, ignore the effects of load, acceleration, and deceleration in the energy consumption evaluation. An order batching and picking optimization with energy consumption minimization is studied by Ene et al. (2016). The authors use genetic algorithm to solve their problem and obtain significant energy savings with efficient batching and routing solutions. However, the energy consumption evaluation is done based on constant energy consumption per unit time, instead of an exact calculation. Moreover, load, acceleration, deceleration, and friction forces are not taken into account in the energy consumption evaluation. Rojanapitoon & Teeravaraprug (2018) consider total distance minimization and energy consumption minimization as two separate objectives. The objectives are then weighted resulting in a single-objective problem. Up to 7.7% saving in energy consumption is reported in this study. However, in the computation of the energy consumption of the order picker, vertical movements, acceleration, and deceleration are not considered.

The classical order picking problem with travel distance or time minimization objective has been widely studied in the literature (Roodbergen 2001, De Koster et al. 2007, Masae et al. 2020). De Koster & Ven der Poort (1998) present a study on comparison of optimal and heuristic solution approaches for the order picking problem. Since the order picking problem can be represented as a TSP (Scholz et al. 2016), some solution approaches have been borrowed from the TSP literature (Charkhgard & Savelsbergh 2015, Makris & Giakoumakis 2003, Theys et al. 2010). Theys et al. (2010) consider the routing problem of order pickers in multi-parallel-aisle system and use a state-of-the-art heuristic for TSP, the Lin-Kernighan-Helsgaun algorithm, to get outperforming solutions. They obtain solutions that are up to 47% better than the ones provided by the existing order picking problem heuristics for the multiple-block warehouse test instances.

Çelik & Süral (2019) provide a complexity analysis of the order picker problem in parallel aisle warehouses with multiple blocks and propose a heuristic called merge-

and-reach. In this heuristic, the warehouse is divided into several single-block parts and each part is considered as a subproblem. The subproblems are solved using a dynamic programming approach that was proposed earlier by Ratliff & Rosenthal (1983). After merging the solutions of the subproblems, the 3-opt local search is applied for possible improvements. The authors, argue that their solution approach outperforms the state-of-the-art heuristics in the literature in terms of both solution quality and robustness.

For the order picking problem, a metaheuristic is proposed by Santis et al. (2018). The proposed approach, namely FW-ACO, integrates the Floyed-Warshall (FW) algorithm and ant-colony optimization (ACO) metaheuristic. In the first stage of this solution approach, the shortest path connecting each pair of item locations is identified using the FW algorithm, while the shortest route for an order picker is found by the ACO in the second stage using the distance matrix provided by the FW algorithm. It is observed that the performance of the proposed approach is effective for small size instances (e.g., with 20 orders) and can be made effective for larger size instances (e.g., with 50 orders) when the parameters for the ACO are well tuned.

In this thesis, we study the energy minimizing order picker forklift routing problem (EMFRP), where an order picker forklift starts its tour at the depot and is to pick all of the ordered items one by one and bring them all at once to the depot in the most energy-efficient manner. To the knowledge of the authors, this is the first study that considers specifically the high-tech order picker forklifts in the context of the order picking problem. Moreover, in the energy consumption calculations, the effects of load, acceleration, deceleration, and friction forces are all taken into account in addition to the effects of horizontal and vertical moves.

The order picker forklifts are able to carry up to a couple of tonnes of load and reach to higher than 10 meters. When a number of orders have to be picked from different locations in a warehouse, an operator rides the forklift to the locations of the items and pick them by hand after adjusting the height of the fork to the height of the item. Note that the order picker forklifts are able to move when their fork are lifted (no need to lower the fork to move). This provides significant energy savings when going from one location to another one that have similar (possibly the same) heights. Figure 1.1

shows how an operator performs the order picking tasks using an order picker forklift. In Figure 1.1a, the operator has already picked some items and is riding the forklift (while its fork is lowered) to the location of the next item. In Figure 1.1b, the operator has adjusted the fork's height to the rack of the item, picked the item and is locating it on the fork of the order picker. In Figure 1.1c, the operator is riding the order picker forklift to the location of the next item while its fork is lifted (preventing undesirable energy consumption due to lowering and lifting the fork). In Figure 1.1d, the operator is locating the picked item on the fork. The energy consumption of an order picker forklift is due to its horizontal and vertical moves while carrying variable amounts of load on its tour. When going from the location of one item to the next one, an amount of energy is needed to move the forklift horizontally to the location of the next item. Also, an amount of energy is needed to lift or lower the fork to the level of the next item to be picked, while the operator is on the fork. In this study, the total energy consumption of the order picker forklift through its complete tour is aimed to be minimized. To this end, the energy consumption of horizontal and vertical moves of the forklift during the tour is obtained by means of work and force calculations.

We provide a mixed-integer (linear) programming (MIP) formulation to solve the EMFRP, which has some characteristics of the classical TSP formulations and is not able to solve large size problem instances in reasonable time. We adapt a dynamic programming algorithm for the EMFRP as well, which can solve instances with up to 25 items within twenty minutes. To be able to solve larger size instances of the EMFRP, TSP-based construction and improvement heuristics, such as nearest neighbor, 2-opt, and 3-opt algorithms, and newly developed construction and improvement matheuristics are used as heuristic approaches. These heuristics are integrated into a single solution approach for the EMFRP. Several sets of computational experiments, on instances generated based on warehouse schemes found in the literature, are performed using the MIP formulation, the dynamic programming approach, and the proposed solution approach. The experimental results show that the dynamic programming approach beats the MIP formulation for small size instances. According to the results, savings in both energy and time are achieved with the solutions of the EMFRP, when compared with the solutions of the classical order picking problem with distance minimization objective.

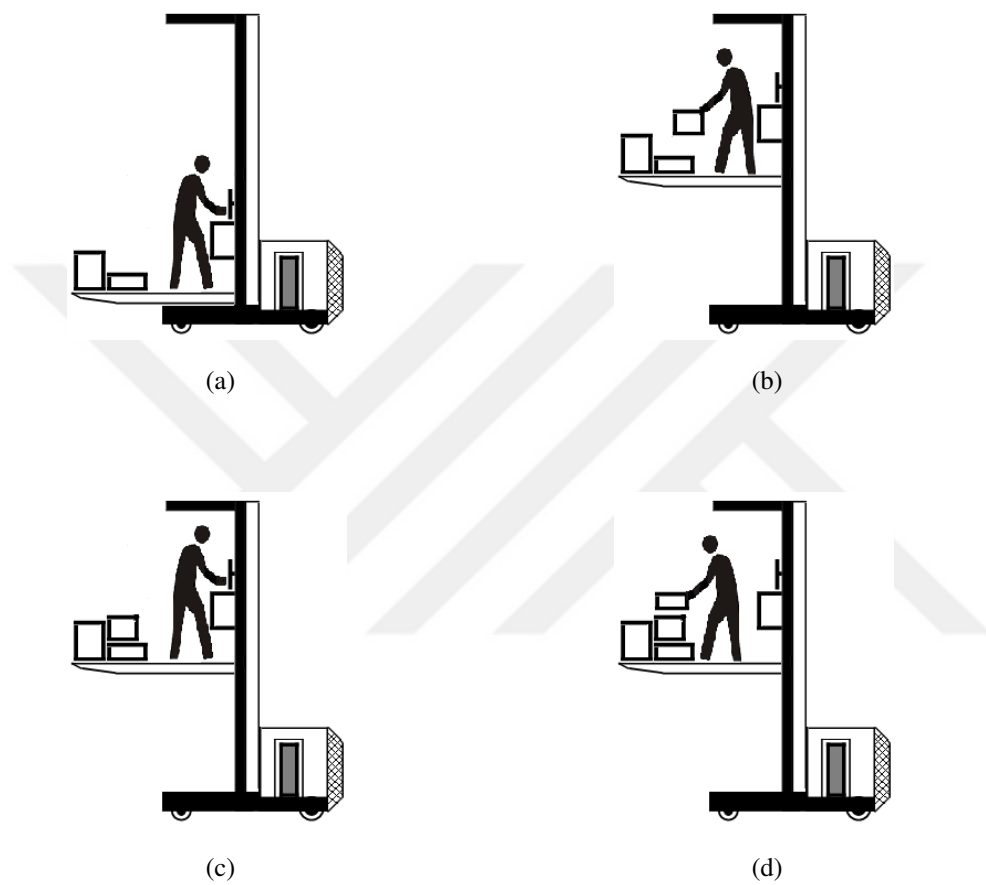


Figure 1.1: An operator riding the order picker forklift when its fork is lowered (a) or lifted (c) and picking items (b,d)

1.3 The Outline of the Thesis

The rest of this thesis is organized as follows. Chapter 2 studies the green Weber problem (GWP) and the time-dependent green Weber problem (TD-GWP) and provides exact solution approaches for them by the second order cone programming formulations. The computational experiments with extensive analysis on the solutions of the GWP and the TD-GWP are also given in the same chapter. The multi-facility version of the GWP, i.e., the multi-facility green Weber problem (MF-GWP), is studied in Chapter 3. In this chapter, we face the challenges for solving the MF-GWP with the mixed-integer second order cone programming formulation exactly. The heuristic solution methods for the MF-GWP are then proposed in the same chapter. Moreover, we provide the computational results of different solution approaches for the MF-GWP. Chapter 4 introduces the energy minimizing order picker forklift routing problem (EMFRP) that focuses on energy efficiency in the material handling systems in warehouses. In this chapter, for the EMFRP, we provide a mixed-integer programming formulation and a dynamic programming approach for solving it exactly, and heuristic solution approaches to be able to solve larger size instances. Then, the experimental results are discussed to compare the solution approaches and to provide new insights into routing order picker forklifts in more energy-efficient ways in warehouses. Finally, Chapter 5 concludes the thesis and outlines some future research directions.



CHAPTER 2

TIME-DEPENDENT GREEN WEBER PROBLEM (TD-GWP)

We consider an extension of the classical Weber problem, named as the green Weber problem (GWP), in which the customers have one-sided time windows. The GWP decides on the location of the single facility in the plane and the speeds of the vehicles serving the customers from the facility within the one-sided time windows so as to minimize the total amount of carbon dioxide emitted in the whole distribution system. We also introduce time-dependent congestion which limits the vehicle speeds in different time periods and call the resulting problem as the time-dependent green Weber problem (TD-GWP). In the TD-GWP, the vehicles are allowed to wait during more congested time periods. We formulate the GWP and TD-GWP as second order cone programming problems both of which can be efficiently solved to optimality. We show that if the traffic congestion is non-increasing, then there exists an optimal solution in which the vehicles do not wait at all. Computational results are provided comparing the locations of the facility and the resulting carbon dioxide emissions of the classical Weber problem with those of the GWP and comparing the GWP with the TD-GWP in terms of carbon dioxide emissions in different traffic congestion patterns.

2.1 The Green Weber Problem (GWP)

We assume that the amount of CO₂ emitted by a vehicle is proportional to its fuel consumption which is aligned with the related literature (see e.g., Demir et al. 2011). As the fuel consumption model, we use the comprehensive modal emission model (CMEM), suggested by Barth et al. (2005), for heavy-good vehicles. For a review and comparison of different vehicle emission models, the reader is referred to Demir

et al. (2011). According to the CMEM, the amount of fuel consumed in liter, f , by a vehicle which travels a distance of z km at a speed of v km/h and with a load of L kg is given by

$$f = \lambda k N V \frac{z}{v} + \lambda w \gamma \alpha z + \lambda \gamma \alpha L z + \lambda \beta \gamma v^2 z, \quad (2.1)$$

where $\lambda, k, N, V, w, \gamma, \alpha$, and β are the fuel consumption parameters depending on the vehicle type and the road conditions. Introducing new parameters $\alpha_1 = \lambda k N V$, $\alpha_2 = \lambda w \gamma \alpha$, $\alpha_3 = \lambda \gamma \alpha$, and $\alpha_4 = \lambda \beta \gamma$, the fuel consumption equation can be rewritten as follows.

$$f = \alpha_1 \frac{z}{v} + \alpha_2 z + \alpha_3 L z + \alpha_4 v^2 z \quad (2.2)$$

Note that the fuel consumption function given in (2.2) is a convex function of v and is minimized when the speed of the vehicle is equal to $(\alpha_1/2\alpha_4)^{1/3}$. Higher or lower speeds than this optimal speed results in higher fuel consumption rate per unit distance traveled. Figure 2.1 shows the fuel consumption rate in liter per 100 km with respect to speed in km/h for a particular vehicle type with no load according to Equation 2.2 and is obtained by using the parameter values in Table 2.1 taken from (Demir et al. 2012). As it can be seen from Figure 2.1, below the speed of 55.2 km/h, the fuel consumed by the vehicle per unit distance traveled decreases as the speed increases. Above the speed of 55.2 km/h, the fuel consumed per unit distance traveled increases with the speed. Note that the optimal speed, i.e., the speed at which the fuel consumption rate is the smallest, may change from vehicle to vehicle, but for a particular vehicle, it is independent of the load, i.e., constant. Changing the load of the vehicle from L_1 kg to L_2 kg shifts the fuel consumption rate curve by the amount $\lambda \gamma \alpha (L_2 - L_1)$.

We assume as in Bektaş & Laporte (2011), Demir et al. (2011) that the amount of CO_2 emitted in kg by a vehicle is proportional to the amount of fuel consumed in liters. Assuming that 1 liter of gasoline contains c kg of CO_2 , we obtain that the amount of CO_2 emitted in kg, C , by a vehicle traveling a distance of z km at a speed of v km/h and with a load of L kg is equal to

$$C = c \left(\alpha_1 \frac{z}{v} + \alpha_2 z + \alpha_3 L z + \alpha_4 v^2 z \right). \quad (2.3)$$

Table 2.1: Values of the parameters used in the fuel consumption calculations

Notation	Description	Values
λ	Constant	3.09636×10^{-5}
k	Engine friction factor (kilojoule/revolution/liter)	0.2
N	Engine speed (revolution/second)	33
V	Engine displacement (liter)	5
w	Vehicle curb weight (kilogram)	6350
γ	Constant	2.77778×10^{-3}
α	Constant	9.81×10^{-2}
L	Vehicle's maximum load (kilogram)	3650
β	Constant	1.64865

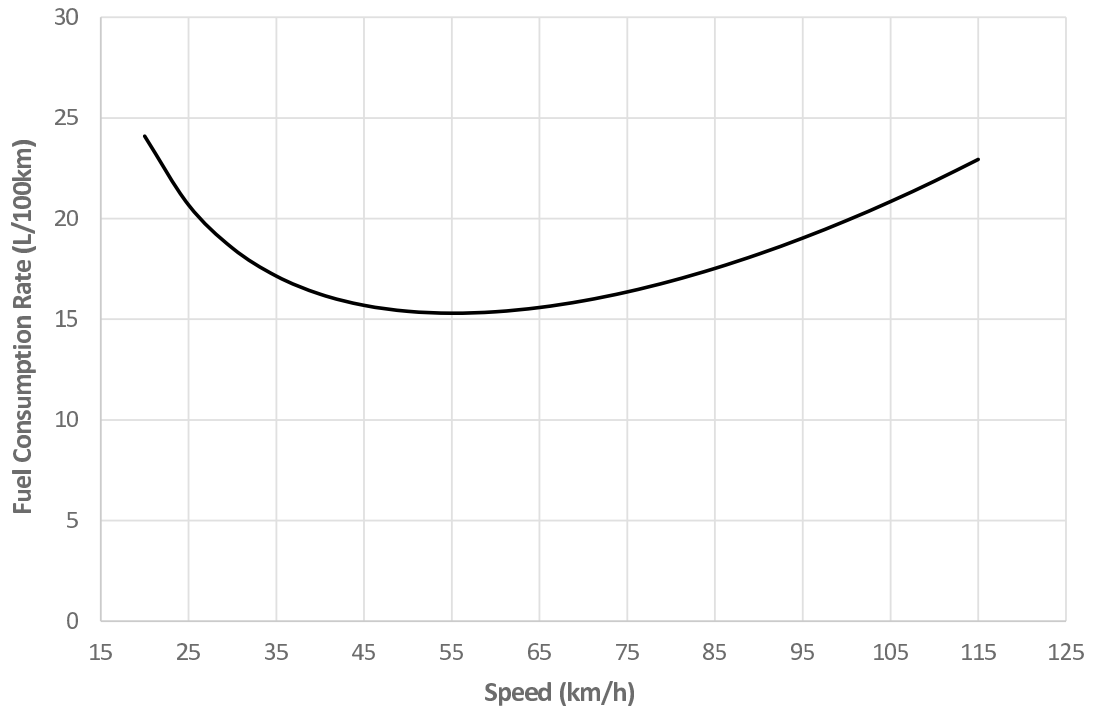


Figure 2.1: Fuel consumption in liter per 100 km as a function of the vehicle speed

In the remaining part of this section, after introducing second order cone programming (SOCP), we formulate the GWP and TD-GWP as SOCP problems showing that both problems can be solved in polynomial time.

2.1.1 Second Order Cone Programming

An SOCP problem is a convex optimization problem of the form

$$\min_{x \in \mathbb{R}^n} e^T x \quad (2.4)$$

subject to

$$\|A_i x + b_i\| \leq c_i^T x + d_i \quad i = 1, 2, \dots, m, \quad (2.5)$$

where, $x \in \mathbb{R}^n$ is the vector of decision variables, and $e \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{n_i \times n}$, $b_i \in \mathbb{R}^{n_i}$, $c_i \in \mathbb{R}^n$, $d_i \in \mathbb{R}$ are the problem parameters. The constraints in (2.5) are called as the second order cone constraints. SOCP problems generalize linear programming (LP) problems as any linear inequality $\alpha^T x \leq \beta$ can be written as a second order cone constraint in the fashion $\|0x + 0\| \leq -\alpha^T x + \beta$. Similar to LP problems, SOCP problems can be solved in polynomial time. Several efficient and numerically stable implementations are available for the solution of SOCP problems. The reader is referred to the survey papers (Alizadeh & Goldfarb 2003, Lobo et al. 1998) for an overview of second order cone programming, algorithms, and application areas.

2.1.2 An SOCP Formulation of the GWP

In the GWP, we assume that there are n customers and customer i has a time limit ℓ_i . The time limits are taken as hard, therefore the vehicle(s) dispatched from the facility for customer i , should arrive at the location of the customer no later than ℓ_i . The aim of the GWP is to find the location of the facility and determine the speeds of the vehicles sent to the customers so that the total fuel emission cost in the distribution system is minimized. The weight of customer i , w_i , represents the number of vehicles sent to the customer. For convenience, we refer all the vehicles sent to customer i as vehicle i . Letting $I = \{1, 2, \dots, n\}$, we formulate the GWP as the following

nonlinear optimization problem.

(GWP-NLP1)

$$\min_{z, v \in \mathbb{R}^n, (x, y) \in \mathbb{R}^2} c \sum_{i \in I} w_i \left(\alpha_1 \frac{z_i}{v_i} + \alpha_2 z_i + \alpha_3 L z_i + \alpha_4 v_i^2 z_i \right) \quad (2.6)$$

subject to

$$z_i \geq \|(x, y) - (a_i, b_i)\| \quad \forall i \in I \quad (2.7)$$

$$\ell_i v_i \geq z_i \quad \forall i \in I \quad (2.8)$$

$$v_i \geq 0 \quad \forall i \in I \quad (2.9)$$

The objective function of (GWP-NLP1) minimizes the total amount of CO₂ emission in kg in the distribution system. The variables are the location (x, y) of the facility, the speed v_i of the vehicle i , and the distance z_i between the facility and the customer i . The constraints and the objective function make sure that $z_i = \|(x, y) - (a_i, b_i)\|$ for every customer $i \in I$ in an optimal solution. Constraints in (2.8) enforce that the vehicles arrive no later than the time limits for all the customers. Constraints in (2.9) are the non-negativity constraints for the speed variables.

The formulation (GWP-NLP1) in its current form is not an SOCP formulation. We introduce a new variable t_i for every customer $i \in I$ that represents the time it takes for vehicle i to reach customer i . Note that $t_i = z_i/v_i$ for each $i \in I$ if z_i is nonzero and $t_i = 0$ if $z_i = 0$. Using the equation $v_i = z_i/t_i$, we rewrite (GWP-NLP1) without the speed variables as follows

(GWP-NLP2)

$$\min_{z, t \in \mathbb{R}^n, (x, y) \in \mathbb{R}^2} c \sum_{i \in I} w_i \left(\alpha_1 t_i + \alpha_2 z_i + \alpha_3 L z_i + \alpha_4 \frac{z_i^3}{t_i^2} \right) \quad (2.10)$$

subject to

$$z_i \geq \|(x, y) - (a_i, b_i)\| \quad \forall i \in I \quad (2.11)$$

$$\ell_i \geq t_i \quad \forall i \in I \quad (2.12)$$

$$t_i \geq 0 \quad \forall i \in I \quad (2.13)$$

Note that the constraints (2.8) and (2.9) in formulation (GWP-NLP1) are replaced by the constraints (2.12) and (2.13) in (GWP-NLP2).

Introducing new variables $g_i, i \in I$ and linearizing the objective function, the GWP

is formulated as

(GWP-SOCP)

$$\min_{z, t, g \in \mathbb{R}^n, (x, y) \in \mathbb{R}^2} c \sum_{i \in I} w_i (\alpha_1 t_i + \alpha_2 z_i + \alpha_3 L z_i + \alpha_4 g_i) \quad (2.14)$$

subject to

$$z_i \geq \|(x, y) - (a_i, b_i)\| \quad \forall i \in I \quad (2.15)$$

$$g_i t_i^2 \geq z_i^3 \quad \forall i \in I \quad (2.16)$$

$$\ell_i \geq t_i \quad \forall i \in I \quad (2.17)$$

$$t_i, g_i \geq 0 \quad \forall i \in I \quad (2.18)$$

Note that the objective function and the constraints (2.17) and (2.18) in (GWP-SOCP) are linear. The constraints (2.15) consist of second order cone constraints. We will show that the GWP can be formulated as an SOCP problem by showing that the constraints (2.16) can be written as second order cone constraints.

In (GWP-SOCP), $z_i \geq 0$ is implied by (2.15). Consider the constraints $g_i t_i^2 \geq z_i^3, t_i, g_i, z_i \geq 0$. This set of constraints is equivalent to $g_i t_i^2 z_i \geq z_i^4, t_i, g_i, z_i \geq 0$ which can be written as $g_i z_i \geq u_i^2, u_i t_i \geq z_i^2, u_i, t_i, g_i, z_i \geq 0$ after introducing the new variable u_i . The latter set of constraints is then equivalent to $\left\| \begin{pmatrix} u_i \\ (g_i - z_i)/2 \end{pmatrix} \right\| \leq (g_i + z_i)/2, \left\| \begin{pmatrix} z_i \\ (u_i - t_i)/2 \end{pmatrix} \right\| \leq (u_i + t_i)/2, u_i, t_i, g_i, z_i \geq 0$ consisting of second order cone constraints.

Note that in formulations (GWP-NLP1) and (GWP-NLP2), the variables v_i and t_i , respectively, cannot take the value zero as otherwise the objective function value would be undefined. This issue, however, is naturally resolved in (GWP-SOCP). Also note that if z_i is zero, then t_i is forced to zero by the objective function in (GWP-SOCP) and if z_i is nonzero, then t_i is forced to take a nonzero value by (2.16) which implies a nonzero speed v_i .

In the next section, we introduce the time-dependent green Weber problem and show that it can also be formulated as an SOCP problem and hence can be efficiently solved to optimality as well.

2.2 The Time-dependent Green Weber Problem (TD-GWP)

We assume that due to congestion, within certain time periods the vehicle speeds are limited. We consider M time periods and the vehicle speeds cannot exceed V_m during the m^{th} time period, $m \in \overline{M} = \{1, 2, \dots, M\}$. The length of the m^{th} time period is denoted by $P_m > 0$. In the TD-GWP, the decisions to be made are the location of the facility and the speed of each vehicle in each time period. The aim of the TD-GWP is again to minimize the total amount of CO₂ emission in the distribution system under hard one-sided time windows. We define the following decision variables: (x, y) represents the coordinates of the facility, z_i^m denotes the distance traveled by vehicle i in the m^{th} time period, z_i denotes the total distance traveled by vehicle i , t_i^m stands for the active time of vehicle i in the m^{th} time period, and t_i represents the total active time spent by vehicle i . In a time period, we allow the vehicles to wait, and the waiting times are not considered in the computation of the active times. Let $\delta_i \in \overline{M}$ represent the index of the last time period containing ℓ_i . We have that $\sum_{m=1}^{\delta_i-1} P_i < \ell_i \leq \sum_{m=1}^{\delta_i} P_i$.

The TD-GWP can be formulated as

(TD-GWP-SOCP)

$$\min \quad c \sum_{i \in I} \sum_{m=1}^M w_i \left(\alpha_1 t_i^m + \alpha_2 z_i^m + \alpha_3 L z_i^m + \alpha_4 \frac{(z_i^m)^3}{(t_i^m)^2} \right) \quad (2.19)$$

subject to

$$z_i \geq \|(x, y) - (a_i, b_i)\| \quad \forall i \in I \quad (2.20)$$

$$\ell_i \geq t_i \quad \forall i \in I \quad (2.21)$$

$$V_m t_i^m \geq z_i^m \quad \forall i \in I, \forall m \in \overline{M} \quad (2.22)$$

$$P_m \geq t_i^m \quad \forall i \in I, \forall m \in \overline{M} \quad (2.23)$$

$$z_i = \sum_{m=1}^M z_i^m \quad \forall i \in I \quad (2.24)$$

$$t_i = \sum_{m=1}^M t_i^m \quad \forall i \in I \quad (2.25)$$

$$t_i^m = 0, z_i^m = 0, \text{ if } m > \delta_i \quad \forall i \in I \quad (2.26)$$

$$\ell_i - \sum_{m=1}^{\delta_i-1} P_i \geq t_i^{\delta_i} \quad \forall i \in I \quad (2.27)$$

$$t_i^m, z_i^m \geq 0 \quad \forall i \in I, \forall m \in \overline{M} \quad (2.28)$$

The objective function (2.19) minimizes the total amount of CO₂ emission in kg in the distribution system, i.e., the sum of CO₂ emission of vehicles traveling in different time periods. The constraints (2.20) and the objective function enforce that $z_i = \|(x, y) - (a_i, b_i)\|$ for every customer $i \in I$ in an optimal solution. Time limits of customers are satisfied by constraints (2.21). Constraints in (2.22) ensure that the speeds of vehicles do not exceed the speed limit in the corresponding time period. Constraints (2.23) indicate that the active time of each vehicle in each time period is not more than the length of the time period. Constraints (2.24) and (2.25) are used to calculate the total distance traveled and total active time spent for each vehicle, respectively. Constraints in (2.27) limit the active time of vehicle i in time period δ_i , i.e., $t_i^{\delta_i}$, by the maximum amount of active time that the vehicle i is allowed to travel in time period δ_i . Constraints in (2.26) force t_i^m and z_i^m to zero whenever m is greater than δ_i and hence together with (2.27) make sure that each vehicle reaches its customer within its time limit. Constraints (2.28) are the non-negativity constraints.

Note that the constraints in (2.21) are redundant as they are implied by (2.23), (2.25), (2.26), and (2.27). Moreover constraints (2.25) are also not necessary in the formulation as there is no t_i in the objective function.

Similar to (GWP-NLP2), the objective function of the formulation (TD-GWP-SOCP)

contains the nonlinear terms $(z_i^m)^3/(t_i^m)^2$. By introducing new variables g_i^m , the nonlinear terms can be linearized similar to what we have in (GWP-SOCP). The constraints $g_i^m(t_i^m)^2 \geq (z_i^m)^3$ can then be written as SOCP constraints as discussed previously.

2.2.1 The TD-GWP with Non-decreasing Congestion

In this section, we assume that there is non-decreasing congestion during the delivery times. This may be reasonable in real life if all the deliveries are expected to be completed early in the morning before the morning traffic peak time.

The notation used earlier will be followed, that is, there are M time periods and during the m^{th} time period which has length $P_m > 0$, the vehicle speeds cannot exceed V_m for every $m \in \overline{M} = \{1, 2, \dots, M\}$. Moreover, we have a congestion pattern that is non-decreasing, i.e., $V_1 \geq V_2 \geq \dots \geq V_M$.

We will show that under these conditions, there exists an optimal solution of (TD-GWP-SOCP) in which none of the vehicles waits on its way during the delivery. Consider an optimal solution $\overline{t}_i^m, \overline{z}_i^m, \overline{z}_i, \overline{t}_i, i \in I, m \in \overline{M}, \overline{x}, \overline{y}$ of (TD-GWP-SOCP) from which the optimal values of the speeds at each time period for each vehicle $\overline{v}_i^m, i \in I, m \in \overline{M}$ is obtained from $\overline{v}_i^m = \overline{z}_i^m / \overline{t}_i^m$, if $\overline{t}_i^m \neq 0$.

Assume that m is the first time period in which vehicle i waits in the optimal solution, i.e., m is the smallest positive integer such that $\overline{t}_i^m < P_m$, and there exists an index $n > m$ such that $\overline{t}_i^n > 0$. Let us take the smallest such n and let v^* be the speed level at which the fuel consumption rate is the smallest, i.e., $v^* = (\alpha_1/2\alpha_4)^{1/3}$. If $V_m \geq v^*$ and $\overline{t}_i^m \neq 0$, then we have that $\overline{v}_i^m = v^*$. If the speed at time period n is different from v^* , we can then increase the active time in time period m (by decreasing the active time in time period n), go at the optimal speed for more, and reduce the fuel consumption. As the current solution is optimal, it should be that $\overline{v}_i^n = v^*$ as well. On the other hand if $V_m \leq v^*$ and $\overline{t}_i^m \neq 0$, then we have that $\overline{v}_i^m = V_m$ by the convexity of the fuel consumption rate function. In this case if $\overline{v}_i^n$ is less than V_m , then we can again increase the active time in time period m , and reduce the fuel consumption. Therefore $\overline{v}_i^n$ has to be equal to $\overline{v}_i^m$ as $\overline{v}_i^n \leq V_n \leq V_m$.

We now claim that if the first time period in which the vehicle i waits in the optimal solution is m , then the speed is the same for all the time periods with non-zero active times after and including time period m . To prove this, let us assume that the first time period after time period m with a different speed value is n , i.e., assume that in time periods m through $n - 1$ with non-zero active times, the speed is v and at time period n , it is $\bar{v}_n \neq v$. v is either equal to v^* in which case the vehicle can drive more at time period m at the optimal speed and reduce fuel consumption, or v is different from v^* . In the latter case with $v \neq v^*$, v has to be smaller than v^* and $\bar{v}_n$ cannot be larger than v as otherwise we would increase the speed v towards v^* without violating the speed limits and decrease the fuel consumption. So, we have that $\bar{v}_n$ is smaller than v . Now, similarly the vehicle can drive more in time period m at the speed v and reduce the fuel consumption as the fuel consumption rate at speed v is smaller than that at speed $\bar{v}_n$. We have thus proven that including and after the time period in which the vehicle waits in the optimal solution for the first time, all the speeds have to be the same in all periods having non-zero active times. The optimal solution of (TD-GWP-SOCP) can easily be turned into one in which no vehicle waits just by shifting all active times to the earlier time periods without changing the speeds.

2.3 Illustrative Example

In this section, an illustrative example is presented. The solutions of the Weber problem (WP) and the GWP, and the solutions of the GWP and the TD-GWP are compared.

Consider an instance of the Weber problem with four customers in the plane, namely customer 1, 2, 3, and 4. The coordinates of the customers are $(0, 0)$, $(1000, 0)$, $(0, 1000)$, and $(1000, 1000)$ in kilometers as seen in Figure 2.2. We assume identical unit weights for all customers, i.e., $w_1 = w_2 = w_3 = w_4 = 1$. The optimal location of the facility for the corresponding WP is given by $(500, 500)$ which is the center of the square in Figure 2.2. Thereafter, we assign a time limit of ℓ_4 hours to customer 4, i.e., the vehicle 4 should arrive at the location of customer 4 on or before ℓ_4 . For the other customers, no time limit is imposed, i.e., $\ell_1 = \ell_2 = \ell_3 = \infty$. We first consider different ℓ_4 values to see its impact on total amount of CO₂ emission in kg in the

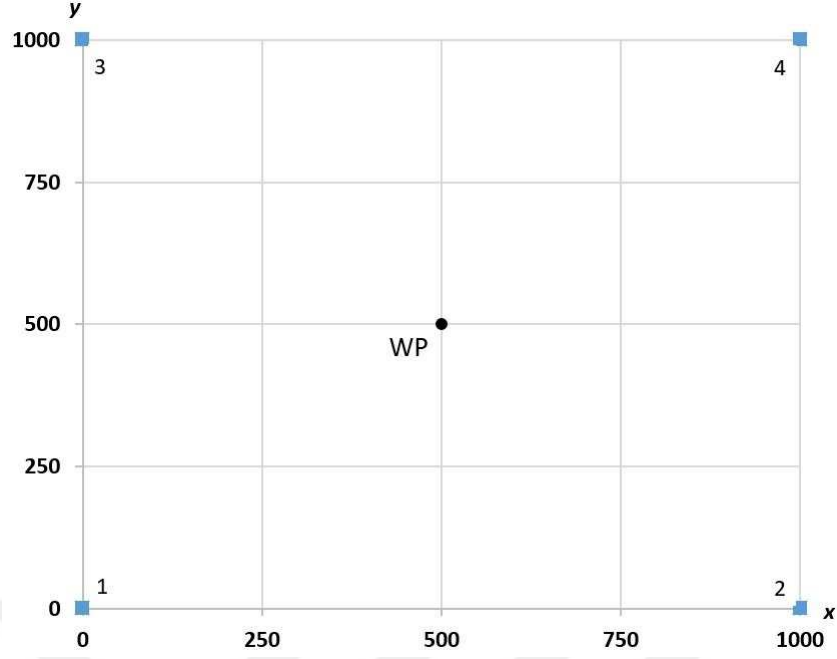


Figure 2.2: An instance of the Weber problem with four customers

distribution system and the optimal location of the facility. The following parameters are obtained by using the values from Table 2.1:

$$\begin{aligned}\alpha_1 &= 3.66350068 \text{ l/h}, \\ \alpha_2 &= 5.33605218 \times 10^{-2} \text{ l/km}, \\ \alpha_3 &= 8.40323179 \times 10^{-6} \text{ l/km/kg}, \\ \alpha_4 &= 1.08968703 \times 10^{-5} \text{ lh}^2/\text{km}^3, \\ L &= 3650 \text{ kg}, \forall i \in I, \\ c &= 2.32 \text{ kg/l}.\end{aligned}$$

When the time limit ℓ_4 is imposed on customer 4, the resulting GWP was solved by CPLEX 12.6 IBM (2012) using the formulation (GWP-SOCP). Different values (in hours) were used for ℓ_4 : 10, 8, 6, 4, and 1 in addition to $\ell_4 = \infty$. As the time limit of customer 4 is tightened, the optimal solution of the GWP gets closer to customer 4. This is to prevent vehicle 4 from speeding too much to satisfy the corresponding time limit and consuming excessive amount of fuel. The optimal facility locations of the GWP for different ℓ_4 values can be seen in Figure 2.3. Note that the optimal facility location of the Weber problem corresponds to the case with $\ell_4 = \infty$.

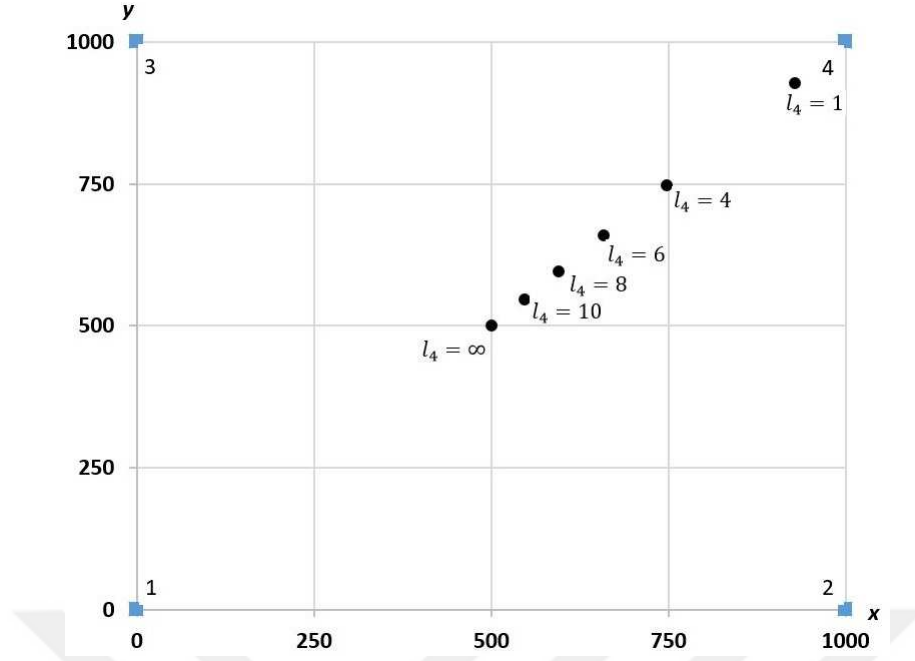


Figure 2.3: Locations of the facilities obtained by the GWP for different ℓ_4 values in the illustrative example

We now compare the total amount of CO₂ emission of the solutions of the WP and the GWP. Note that as the solution of the WP does not take the time limits into account, it does not change as ℓ_4 is changed. However, when the solution of the WP is used as the location of the facility, vehicle 4 may need to speed up to reach customer 4 on time. This will increase the total amount of CO₂ emission of the solution of the WP in comparison with that of the solution of the GWP.

In Table 2.2, the rows with the headers WP and GWP refer to the values obtained by solving the WP and the GWP, respectively, for different ℓ_4 values. Columns with the header “CEM” give the total amount of CO₂ emission in kg and columns with the header $\overline{v}_4$ give the speed of vehicle 4 ($\overline{v}_4$) in km/h resulted from the solutions of the WP and the GWP. The last row designates the percent deviation (%dev) of the total amount of CO₂ emission and the speed of vehicle 4 obtained by solving the WP from those by solving the GWP, i.e., %dev = 100 × (the result of the WP - the result of the GWP) / the result of the GWP. As it is seen in the table, the solutions of the GWP improve on the solutions of the WP with respect to both cost and speed $\overline{v}_4$ for every ℓ_4 value. The percent deviations of CEM and $\overline{v}_4$ increase as ℓ_4 gets smaller. For

Table 2.2: Comparison of the solutions of the WP and the GWP in the illustrative example

	$\ell_4 = 10$		$\ell_4 = 8$		$\ell_4 = 6$		$\ell_4 = 4$		$\ell_4 = 1$	
	CEM	$\overline{v_4}$	CEM	$\overline{v_4}$	CEM	$\overline{v_4}$	CEM	$\overline{v_4}$	CEM	$\overline{v_4}$
WP	1215	70	1249	88	1341	118	1634	177	2708	707
GWP	1210	64	1226	72	1253	80	1298	90	2150	101
%dev	0.4	8.6	1.9	18.2	6.6	32.2	20.6	49.2	610	595

instance, when ℓ_4 is taken as 6 and the facility is located based on the solution of the WP, the vehicle 4 needs to travel at a speed of 118 km/h to reach customer 4 on time resulting in a total amount of CO₂ emission of 1341 kg in the distribution system. On the other hand, with the same ℓ_4 value, if the location of the facility is determined by the solution of the GWP, then the speed of vehicle 4 turns out to be 80 km/h resulting in a total amount of CO₂ emission of 1253 kg. Note that in both cases, all the other three vehicles travel at the optimal speed, i.e., $v^* = 55.2$ km/h, as no time limit is imposed on them.

Figure 2.4 illustrates the changes in the total amount of CO₂ emission resulting from the WP and the GWP and the percent deviations as the time limit ℓ_4 changes. The left vertical axis stands for the total amount of CO₂ emission in kg and the right one for the percent deviation. It can be seen from the figure that the percent deviations increase as ℓ_4 gets smaller, i.e., the rate of increase of CEM per one unit decrease in ℓ_4 is more for the WP than the GWP.

We now consider two congestion patterns, one with 2 time periods and the other with 3. In the case of 2 time periods, the problem is referred to as TD-GWP-2. We assume that the length of the first time period (P_1) is one fourth of the corresponding time limit of customer 4, e.g., $P_1 = 2.5$ h when $\ell_4 = 10$ h and $P_1 = 1$ h when $\ell_4 = 4$ h, during which due to limited congestion, it is assumed that $V_1 = 110$ km/h. In the second time period, there is a speed limit of $V_2 = 40$ km/h due to a heavier traffic congestion.

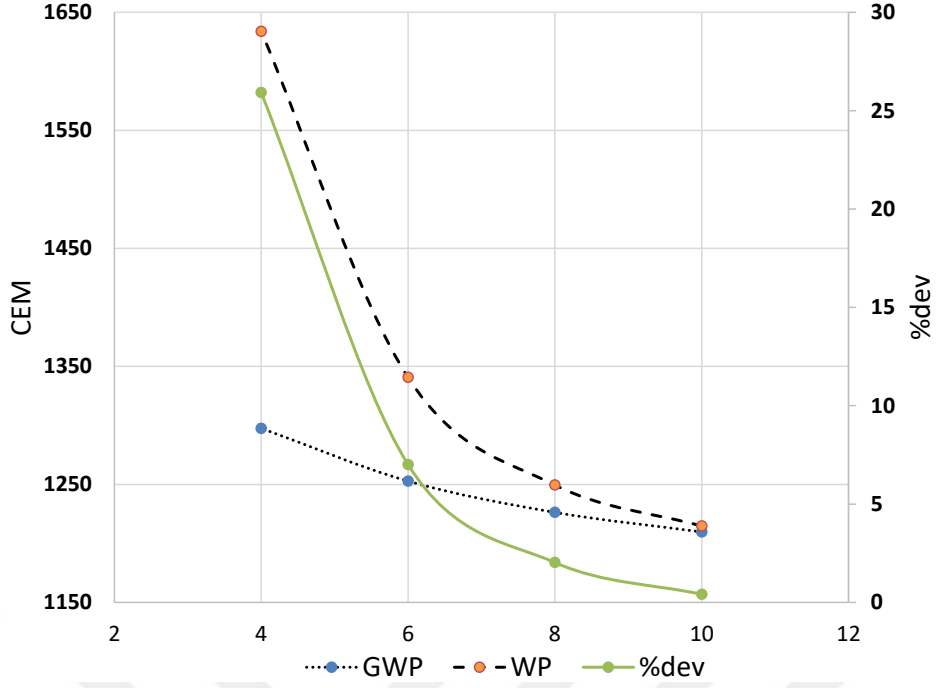


Figure 2.4: Total amounts of CO₂ emission of the solutions of the WP and the GWP together with the percent deviations as ℓ_4 changes in the illustrative example

In the case of 3 time periods, the problem is referred to as TD-GWP-3. It is assumed that the lengths of both the first and the second time periods, P_1 and P_2 , are one fourth of the time limit ℓ_4 of customer 4. In the first, second, and the third time periods, it is assumed that there is a speed limit of $V_1 = 110$ km/h, $V_2 = 40$ km/h, and $V_3 = 30$ km/h, respectively.

In Table 2.3, the row with the header GWP refers to the results with no traffic congestion. Different ℓ_4 values are considered. In the row with the header TD-GWP-2, the results of the solutions of TD-GWP-2 are shown for different ℓ_4 values. The results of TD-GWP-3 are displayed in the row with the header TD-GWP-3. The solutions are compared in terms of the total amount of CO₂ emission (CEM) in the distribution system and the maximum speed at which vehicle 4 travels. The rows with the header %dev include the percent deviations of the displayed results of TD-GWP-2 from the GWP and of TD-GWP-3 from TD-GWP-2, respectively, from top to bottom.

It can be seen from Table 2.3 that the heavier the traffic congestion, the higher the value of the maximum speed of vehicle 4 and the more the total amount of CO₂

Table 2.3: Comparisons of the solutions of the GWP and TD-GWP-2; and TD-GWP-2 and TD-GWP-3 in the illustrative example

	$\ell_4 = 10$		$\ell_4 = 8$		$\ell_4 = 6$		$\ell_4 = 4$	
	CEM	$\overline{v_4^1}$	CEM	$\overline{v_4^1}$	CEM	$\overline{v_4^1}$	CEM	$\overline{v_4^1}$
GWP	1210	64	1226	72	1253	80	1298	90
TD-GWP-2	1288	81	1317	88	1359	95	1407	100
%dev	6.4	26.6	7.4	22.2	8.5	18.8	8.4	11.1
TD-GWP-3	1399	90	1443	97	1498	103	1561	108
%dev	8.6	11.1	9.6	10.2	10.2	8.4	11.0	8.0

emission in the distribution system. As an example, consider the case with $\ell_4 = 8$. If there is no congestion at all, then the total amount of CO₂ emission in the system is 1226 kg and $\overline{v_4^1} = 72$ km/h. In the scenario with two time periods and $V_1 = 110$ km/h and $V_2 = 40$ km/h, the total amount of CO₂ emission increases to 1317 kg and $\overline{v_4^1}$ to 88 km/h resulting in percent deviation values of 7.4% and 22.2%, respectively. These increases are mainly due to the increased congestion in the second time period. In the highest congestion scenario, i.e., three time periods with $V_1 = 110$ km/h, $V_2 = 40$ km/h, and $V_3 = 30$ km/h, the total amount of CO₂ emission further increases up to 1443 kg and the highest speed of vehicle 4 to 97 km/h resulting in a percent deviation of 9.6% and 10.2%, respectively, with respect to the solution of TD-GWP-2.

2.4 Computational Experiments

In this section, our aim is to compare the total amount of CO₂ emission in large scale randomly generated distribution systems with and without traffic congestion. We generate two sets of instances. In the first set, there are 500 customers in the distribution system and in the second one 1000 customers.

For the first set of instances, we consider a square with side length of 234 km. Cus-

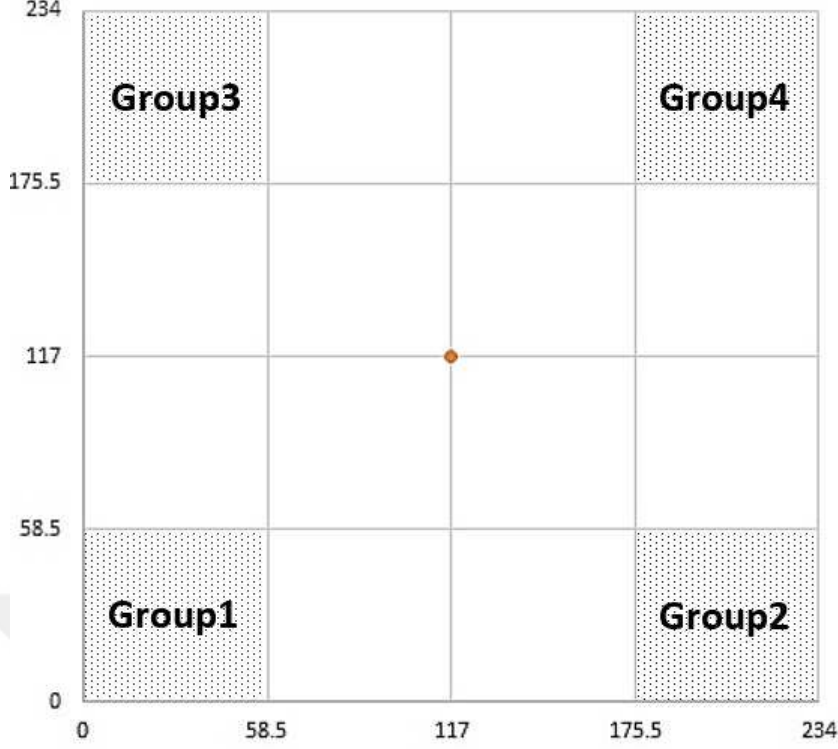


Figure 2.5: Locations of the customers in the 500 customer case

customers with identical unit weights are distributed in this square in 4 groups. Each group consists of 125 customers located uniformly at random in the corresponding area as shown in Figure 2.5. Five problem instances each with a total of 500 customers are generated and the resulting SOCP problems are solved by CPLEX 12.6 IBM (2012) on a server with 3.20 GHz speed and 16Gb RAM. For each instance, it took CPLEX less than a second to solve it. For the GWP, we use the total amount of CO₂ emission parameters mentioned in Section 2.3 and consider a time limit of 1 hour for the customers in group 4 with a speed limit of 110 km/h for all vehicles. No time limit is imposed for the customers in the other groups. In TD-GWP-2, the length of the first time period is taken as one fourth of ℓ_4 , i.e., $P_1 = 15$ minutes, during which the vehicles have a speed limit of $V_1 = 110$ km/h and for the second time period the speed limit is taken as $V_2 = 40$ km/h. For the TD-GWP-3, we take the length of both the first and second time periods as $P_1 = P_2 = 15$ minutes. The speed limit in the first, second, and third time periods are taken as 110km/h, 40km/h, and 30km/h, respectively.

Table 2.4: Comparison of 4 problems with respect to the average total amounts of CO₂ emission over 5 instances in the 500 customer case

	WP	GWP	TD-GWP-2	TD-GWP-3
CEM	30631	28221	33077	37021
%dev		-8.5	17.2	11.9

Table 2.5: Comparison of 4 problems with respect to the average of the average speeds of the vehicles over 5 instances in different time periods in the 500 customer case

Time Periods	Average of average speed (%dev from optimal speed)			
	WP	GWP	TD-GWP-2	TD-GWP-3
1	63.5 (15.4)	57.2 (3.8)	58.1 (5.4)	62.8 (14.0)
2	↓	↓	40.0 (27.4)	40.0 (27.4)
3	↓	↓	↓	30.0 (45.6)

In Table 2.4, the total amounts of CO₂ emission resulting from the solutions of different problems are given. The last row of the table depicts the percent deviation of the average total amount of CO₂ emission of a problem from that of the problem in the previous column. Note that the value in the column with the header WP refers to the average total amount of CO₂ emission resulting from locating the facility as the solution of the WP and imposing the time limits as in the GWP. It can be seen from Table 2.4 that when there is no congestion, locating the facility based on the solution of the GWP instead of that of the WP results in about 8.5% reduction in the average total amount of CO₂ emission. As the congestion increases, the average total amount of CO₂ emission increases as well from the solution of the GWP in the direction of the solution of the TD-GWP-3.

Table 2.5 shows the average of the average speeds of all vehicles over 5 instances in different time periods for different problems. First column refers to the 3 time periods. Note that the first time period of the WP and the GWP corresponds to the union of the first, second, and the third time periods of TD-GWP-3, and the second

Table 2.6: Comparison of 3 problems with respect to the average amounts of CO₂ emission over 10 instances in the 1000 customer case

Time Limit Type	Average CEM (%dev from previous problem)		
	GWP	TD-GWP-2	TD-GWP-3
TL1	40161	41385 (3.0)	41984 (1.5)
TL2	38837	40931 (5.4)	41594 (1.6)

time period of TD-GWP-2 corresponds to the union of the second and third time periods of TD-GWP-3 (a down arrow was used in Table 2.5 to indicate this). For each problem, the corresponding absolute value of the percent deviation from the optimal speed v^* is given in parenthesis. It is seen that in the GWP, a lower speed is required on average for the vehicles to satisfy the time limits of the customers when compared with the WP. Imposing time dependent traffic congestion and speed limit in the second time period, the resulting average speed for the vehicles increases in the first time period of TD-GWP-2 which is to satisfy the customers' time limits. For the TD-GWP-3, the average speed of the vehicles in the first time period is even higher due to more restriction of the speed in the third time period. It can be concluded that more restriction of the speed of vehicles in congested time periods results in a higher average speed in the first time periods. Successively, we see higher deviations from the optimal speed in cases of higher traffic congestion corresponding to an increased total amount of CO₂ emission.

For the second set of instances, 1000 customers are uniformly generated at random in a square region with side length of 234 km. All customers are again assumed to have identical unit weights. Two types of time limits are imposed. For the first one, which we call as TL1, the time limit for each customer in the north east quarter of the region is obtained by $d_i/100$ where d_i is the distance of customer i to the center of the region. No time limit is imposed on the customers located in the other quarters. For the second one, referred to as TL2, the time limit for all customers in the north east quarter of the region is taken as 1.5 h. Similarly, for the other customers, no time limit is imposed. For the GWP, and the first time periods of the TD-GWP-2 and

Table 2.7: Comparison of 3 problems with respect to the average of the average speeds of the vehicles over 10 instances in different time periods in the 1000 customer case under the time limit type TL1

Time Periods	Average of average speed (%dev from optimal speed)		
	GWP	TD-GWP-2	TD-GWP-3
1	59.1 (7.3)	62.4 (13.2)	64.7 (17.4)
2	↓	40.0 (27.4)	40.0 (27.4)
3	↓	↓	30.0 (45.6)

Table 2.8: Comparison of 3 problems with respect to the average of the average speeds of the vehicles over 10 instances in different time periods in the 1000 customer case under the time limit type TL2

Time Periods	Average of average speed (%dev from optimal speed)		
	GWP	TD-GWP-2	TD-GWP-3
1	57.4 (4.2)	59.1 (7.3)	61.2 (11.0)
2	↓	40.0 (27.4)	40.0 (27.4)
3	↓	↓	30.0 (45.6)

TD-GWP-3 which have both 1 h length, the speed limit of 110 km/h is considered. For the second time periods of TD-GWP-2 and TD-GWP-3, the speed limit is taken as $V_2 = 40$ km/h. The length of the second time period of TD-GWP-3 is considered to be 1 h and the speed limit for the third time period of TD-GWP-3 is chosen as 30 km/h. For the computational experiments, 10 instances with 1000 customers are generated for the GWP, TD-GWP-2, and TD-GWP-3.

In Table 2.6, the average total amounts of CO₂ emission resulting from the solutions of the three problems are given together with the percent deviations from solution of the problem in the previous column in parenthesis.

The average total amounts of CO₂ emission of the solutions of the TD-GWP-2 are

more than those of the GWP due to the speed limit of 40 km/h imposed in TD-GWP-2 after 1 h. Similarly, the increase in the average total amounts of CO₂ emission of the solutions of TD-GWP-3 with respect to those of TD-GWP-2 are due to the speed limit imposed in the third time period of TD-GWP-3.

Table 2.7 and Table 2.8 show the average of the average speeds of all the vehicles over 10 instances in different time periods for the time limit types TL1 and TL2, respectively. Similar to Table 2.5, absolute values of the percent deviations from the optimal speed are given in parenthesis. It is observed that the average speeds of the vehicles in the first time period are higher in the more congested traffic patterns resulting in higher average total amounts of CO₂ emission.

When the solution times of the instances are examined, it is observed that they are all less than a second for the instances with 500 customers and less than three seconds for the instances with 1000 customers. As SOCP problems are efficiently solvable, large scale instances can be solved in reasonable times.

The congestion patterns considered in the computational experiments are non-decreasing. By the result in Section 2.2.1, we know that for each instance, there exists an optimal solution in which no vehicle waits on its way. When we examine the optimal solutions of the instances, we see that no vehicle waits in any solution. This can be attributed to the tight time limits imposed in the instances and low speed limits imposed in the second and third time periods.

2.5 Concluding Remarks

The GWP is an extension of the classical Weber problem and it determines the location of the single facility in the plane and the speeds of the vehicles serving the customers such that the total CO₂ emission in the distribution system is minimized. The customers have hard time limits and the vehicles serving the customers must finish their service on or before the time limits. This chapter provides an SOCP formulation for the GWP in which if there is no time limit or all time limits are relaxed enough, the optimal facility location of the GWP corresponds to that of the WP. This is one of the few studies in the literature that optimizes the vehicle speeds without

using any discretization. This chapter also introduces the TD-GWP with SOCP formulation, in which time-dependent congestion limits the vehicle speeds in different time periods. Due to the second order cone programming formulation of the introduced problems, instances with 1000 customers are solved within a couple of seconds in this study. In the computational experiments, it is shown that locating the facility based on the solution of the WP (without taking the time limits into account) instead of the GWP results in a higher total CO₂ emission amounts in the distribution system. Also, as expected, it is seen that the higher the traffic congestion, the higher the total-fuel emission cost in the distribution system.





CHAPTER 3

MULTI-FACILITY GREEN WEBER PROBLEM (MF-GWP)

Locating facilities to satisfy the demands of customers is a strategic decision for a distribution system. In this article, we study the multi-facility green Weber problem (MF-GWP), an extension of the classical multi-facility Weber problem, that considers environmental concerns in a distribution system in the context of a planar facility location problem. In the MF-GWP, the vehicles are sent directly from the facilities to the assigned customers to satisfy their demands. Each customer has a deadline and the vehicles serving the customer must arrive at the location of the customer no later than the deadline. The MF-GWP determines the locations of p facilities on the plane, $p > 1$, allocations of customers to the facilities, and the speeds of the distribution vehicles so as to minimize the total amount of carbon dioxide emission in the distribution system. We formulate this problem as a mixed integer second order cone programming (MISOCP) problem. This formulation turns out to be weak and therefore only small size instances can be solved to optimality within four hours. For larger size instances, a local search heuristic is proposed and some well-known heuristics developed for the multi-facility Weber problem, namely “location-allocation”, “transfer follow-up”, and “decomposition” are adapted for the MF-GWP. We use second order cone programming and the proposed MISOCP formulation as subproblems within the heuristics. For example, the local search heuristic uses a strengthened and reduced-size MISOCP formulation and is itself employed within the decomposition heuristic. We provide our computational experiments to compare the proposed solution methods in terms of solution quality and time. The results show that within a fixed computational time, even though the location-allocation heuristic is able to make more replications, the improvement heuristics considered, i.e., transfer or transfer followed by decomposition, usually find better solutions while using less number of replications.

We also investigate how the total amount of carbon dioxide emitted by distribution vehicles changes with respect to the number of facilities located. Moreover, we show the applicability of the MF-GWP within an assembly line system as an illustrative example, where the stations are fed by dedicated rail-guided vehicles.

3.1 Problem Description and Notation

The multi-facility green Weber problem (MF-GWP) is an extension of the classical multi-facility Weber problem. In the MF-GWP, there are n customers on the plane to be served from p facilities. Let $I = \{1, 2, \dots, n\}$ and $J = \{1, 2, \dots, p\}$. We assume that the facilities are uncapacitated. Each customer has to be served by one facility. However, a facility can serve more than one customer. The locations of the customers are known and are denoted by $(a_i, b_i) \in \mathbb{R}^2, i \in I$. We denote by G the vector of the locations of the customers, i.e., $G(i) = (a_i, b_i)$. Assuming that customer i is served by facility j , the demand of customer i is satisfied by sending $w_i > 0$ vehicles directly from facility j to customer i , each carrying a load L_i . For convenience, by “vehicle i ”, we mean all vehicles sent to customer i . Each customer has a hard deadline and vehicle i has to arrive at the location of customer i on or before the deadline ℓ_i . We denote by $\mathcal{L}$ the vector of the deadlines of the customers, i.e., $\mathcal{L}(i) = \ell_i$. The decisions to be made in the MF-GWP are the locations $(x_j, y_j), j \in J$ of the facilities and the speeds $v_i, i \in I$ of the vehicles to be sent to the customers from the facilities; and the objective is to minimize the total amount of CO_2 emitted by the vehicles in the distribution system.

Denoting the distance between the facility and customer i by $z_i, i \in I$, and the time it takes vehicle i to reach customer i by $t_i, i \in I$, we can use Equation (2.3) to calculate the amount of CO_2 emitted by vehicle i as

$$C_i = cw_i \left(\alpha_1 t_i + \alpha_2 z_i + \alpha_3 L_i z_i + \alpha_4 \frac{z_i^3}{t_i^2} \right), \quad (3.1)$$

which is obtained from (2.3) by replacing the speed variable by z_i/t_i and multiplying the expression by the number of vehicles serving customer i , w_i .

Note that the MF-GWP with $p = 1$ is called as the green Weber problem (GWP). The GWP is introduced and studied in Chapter 2 where it is formulated as a second order

cone programming (SOCP) problem in Section 2.1.2. In this respect, we use the same techniques to handle the nonlinearity in 3.1 while using this equation in the objective function of the MF-GWP with the mixed-integer second order cone programming (MISOCP) formulation provided in the next section.

3.2 Solution Approaches for the MF-GWP

The MF-GWP is NP-hard since it is a generalization of the NP-hard classical multi-facility Weber problem Megiddo & Supowit (1984). To show this, consider an instance of the multi-facility Weber problem with the objective function $\sum_{i=1}^n \beta_i z_i$, where each β_i is a rational number. We will reduce this to an instance of the MF-GWP. Assume that in the MF-GWP, all the deadlines are infinity, all the vehicles are of the same type, and all the loads are identical. Then the objective function of the MF-GWP becomes $\sum_{i=1}^n w_i \zeta z_i$, where ζ is the amount of CO₂ emitted by a vehicle per unit distance traveled. We need to set the values of w_i 's so that the instance of the classical multi-facility Weber problem reduces to an instance of the MF-GWP. To do this, we find positive multiplier m so that $\beta_i m / \zeta$ is a positive integer for every i . We let $w_i = \beta_i m / \zeta$ and hence obtain $m \sum_{i=1}^n \beta_i z_i = \sum_{i=1}^n w_i \zeta z_i$.

Next, we discuss the solution methods we propose for the MF-GWP. We first show that the MF-GWP can be formulated as an MISOCP problem which is an SOCP problem with the additional restriction that some or all of the variables are integer-valued. This formulation is only able to solve small size instances to optimality within four hours. To solve larger size instances, ideas from the well-known heuristics developed for the multi-facility Weber problem are utilized in addition to a newly developed local search heuristic.

3.2.1 An MISOCP Formulation for the MF-GWP

In order to formulate the MF-GWP, the notation described in Section 3.1 is used. We use $\alpha_1^i, \alpha_2^i, \alpha_3^i$, and α_4^i to denote the parameters used in the emission model for vehicle i . In other words, the vehicles do not need to be homogeneous. Moreover, we introduce a binary variable h_{ij} for every $i \in I, j \in J$ which takes the value 1 if

and only if customer i is served by facility j . The distance between customer i and facility j is denoted by $q_{ij}, i \in I, j \in J$. We also use auxiliary variables $g_i, i \in I$ to linearize the objective function as is done in the SOCP formulation of the GWP in Section 2.1.2.

An MISOCP formulation of the MF-GWP is given below.

(MF-GWP-MISOCP)

$$\min \quad c \sum_{i \in I} w_i (\alpha_1^i t_i + \alpha_2^i z_i + \alpha_3^i L_i z_i + \alpha_4^i g_i) \quad (3.2)$$

subject to

$$q_{ij} \geq \|(x_j, y_j) - (a_i, b_i)\| \quad \forall i \in I, j \in J \quad (3.3)$$

$$z_i \geq q_{ij} - M_i(1 - h_{ij}) \quad \forall i \in I, j \in J \quad (3.4)$$

$$\sum_{j \in J} h_{ij} = 1 \quad \forall i \in I \quad (3.5)$$

$$g_i t_i^2 \geq z_i^3 \quad \forall i \in I \quad (3.6)$$

$$\ell_i \geq t_i \quad \forall i \in I \quad (3.7)$$

$$h_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J \quad (3.8)$$

$$t_i, g_i \geq 0 \quad \forall i \in I \quad (3.9)$$

The objective function (3.2) of (MF-GWP-MISOCP) minimizes the total amount of CO₂ emission in kg in the distribution system. Constraints (3.3) and (3.4) and the objective function (3.2) make sure that if $h_{ij} = 1$, then $z_i = \|(x_j, y_j) - (a_i, b_i)\|$ for customer i and facility j in an optimal solution. Constraints (3.5) make sure that each customer is served by exactly one facility. Constraints (3.6) arise when the terms $z_i^3/t_i^2, i \in I$ are taken to the constraints by linearizing the objective function. Constraints (3.7) enforce that the vehicles arrive no later than the deadlines for all the customers. Constraints (3.8) are used to force that h_{ij} is a binary variable for each $i \in I, j \in J$. Constraints (3.9) are the non-negativity constraints. Note that the non-negativity of each $q_{ij}, i \in I, j \in J$ and each $z_i, i \in I$ are implied by constraints (3.3)-(3.5).

For each customer i and facility j , the big-M value M_i in (3.4) can be taken as the maximum distance between customer i and the other customers. This is because the

facility locations will be in the convex hull of the locations of the customers in the optimal solution.

The objective function (3.2) and constraints (3.4), (3.5), (3.7), and (3.9) are linear, constraints (3.3) are SOCP constraints, while constraints (3.6) can be rewritten as SOCP constraints as discussed in Section 2.1.2).

There are some issues with the formulation (MF-GWP-MISOCP). Firstly, the formulation is weak due to the big-M values in constraints (3.4). Secondly, the formulation is highly symmetric. Consider a set of facility locations. The facility locations can be assigned to the location variables (x_j, y_j) 's by permuting the facility indices. Each permutation results in almost the same solution where the only difference is the labels of the facilities. This symmetry inflates the size of the branch-and-bound tree. To reduce the symmetry in the formulation, we propose the following symmetry breaking (SB) constraints whose effect is to be investigated in Section 3.3.

$$x_1 \leq x_2 \leq \dots \leq x_p. \quad (3.10)$$

The SB constraints (3.10) ensure that the leftmost facility is called as facility 1, the second leftmost facility is called as facility 2 and so on.

The above MISOCP formulation can be extended in several ways. For example, by introducing the inequalities $\underline{v}_i t_i \leq z_i \leq \bar{v}_i t_i$, one can enforce that the speed of vehicle i takes a value in the interval $[\underline{v}_i, \bar{v}_i]$. By adding constraints $(x_j, y_j) \in P_j$, where P_j is any MISOCP-representable set, e.g., the union of finitely many polytopes or a disk, one can restrict the possible location of facility j . Moreover, the capacities of the facilities can be restricted by adding the constraints $\sum_{i \in I} w_i L_i h_{ij} \leq \sigma_j$, where σ_j denotes the capacity of facility j . The three types of constraints discussed in this paragraph are all used in the illustrative example discussed in Section 3.4.

3.2.2 Heuristics for the MF-GWP

The MISOCP formulation (MF-GWP-MISOCP) proposed for the MF-GWP has some weaknesses (discussed at the end of Section 3.2.1) and therefore it is only able to solve small size instances to optimality within four hours (see Section 3.3). To be able to solve larger instances of the MF-GWP, we propose several heuristics in this section.

3.2.2.1 Alternate Location-allocation Heuristic

The alternate location-allocation (ALA) heuristic is proposed by Cooper (1964) to solve multi-facility location-allocation problems. The heuristic starts with given initial facility locations and alternates between the following allocation and location steps until convergence is achieved.

1. (Allocation Step) Given the locations of the facilities, assign each customer to a facility optimizing the considered objective.
2. (Location Step) Given the allocations of the customers to the facilities, find the optimal locations of the facilities optimizing the considered objective.

The ALA heuristic is a descent heuristic, i.e., the objective function value never increases during the algorithm. The solution obtained, however, is highly dependent on the initial facility locations. Therefore, a common practice is to run the ALA heuristic several times, i.e., doing several replications, with different sets of initial facility locations and taking the best solution obtained.

In the context of the MF-GWP, in the allocation step, each customer is assigned to the closest facility as this results in the minimum CO₂ emission. In the location step, with known allocations, p green Weber problems are solved using the SOCP formulation (GWP-SOCP) given in Section 2.1.2. In our implementation, the initial facility locations are randomly generated from the convex hull of the locations of the customers. The steps of the ALA heuristic are shown in Algorithm 1.

The inputs of Algorithm 1 are the vector of the customers' locations, G , the number of facilities to be located, p , the vector of initial facility locations, X_0 , and the vector of the customers' deadlines, D . Steps 5 and 6 in Algorithm 1 are the allocation and location steps, respectively. Even though G and D do not appear explicitly in the algorithm, G is used in both steps 5 and 6, and D is used in step 6. The outputs of the algorithm are the final locations of the facilities, X , and the vector of allocations of the customers, H .

Algorithm 1 Alternate Location-allocation Heuristic

```
1: procedure LOCATION-ALLOCATION( $G, p, X_0, D$ )
2:    $k = 0$ 
3:   while There is an improvement in the objective function value do
4:      $k \leftarrow k + 1$ 
5:     Allocate customer  $i$  to the closest facility considering the locations in
        $X_{k-1}$  and let  $H(i)$  be the index of the facility customer  $i$  is allocated to, for each
        $i \in I$ . ▷ Allocation Step
6:     Locate facility  $j$  by solving (GWP-SOCP) (see Section 2.1.2) using only
       the customers allocated to facility  $j$  and let the solution be the  $j^{th}$  component of
        $X_k$ , for each  $j \in J$ . ▷ Location Step
7:   end while
8:    $X \leftarrow X_k$ 
9: end procedure
```

3.2.2.2 Local Search

Given any set of initial facility locations $(\bar{x}_j, \bar{y}_j), j \in J$, the aim of the local search (LS) is to search for a better solution in a small neighbourhood of the current solution. For this purpose, we draw circles around the current facility locations and allow each facility to move within the circle drawn around it resulting in a reduced search space. To do this, we add the following set of constraints to the formulation (MF-GWP-MISOCP).

$$\|(x_j, y_j) - (\bar{x}_j, \bar{y}_j)\| \leq r_j, j \in J. \quad (3.11)$$

Addition of constraints (3.11) makes the formulation (MF-GWP-MISOCP) stronger by

1. restricting the possible locations of the facilities,
2. allowing one to replace the M_i 's in constraint set (3.4) with M_{ij} 's for different customer-facility pairs, and use smaller big-M values,
3. allowing the possibility of fixing some binary variables in advance (which also reduces the size of the formulation).

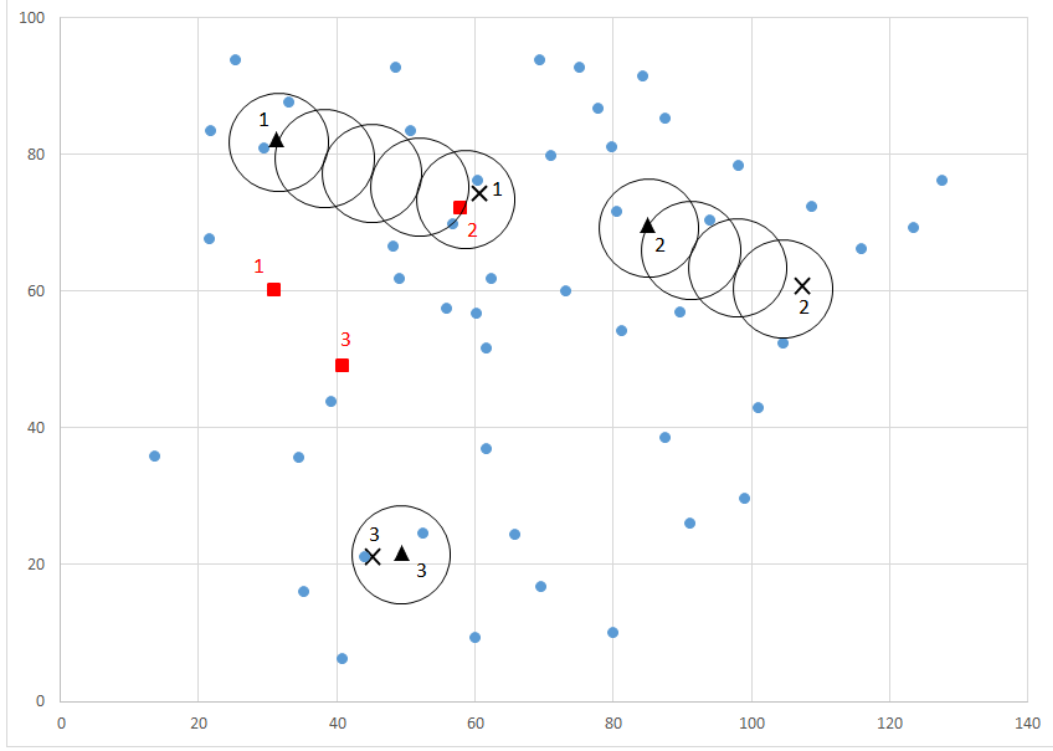


Figure 3.1: Illustration of the steps of the ALA heuristic followed by the LS heuristic

In reference to the second item above, in the implementation of the LS, we take the value of M_{ij} as

$$\min\{M_i, \|(a_i, b_i) - (\bar{x}_j, \bar{y}_j)\| + r_j\}, \quad (3.12)$$

which is an upper bound on the distance between the location of customer i and the new location of facility j .

In reference to the third item above, h_{ij} is fixed to 0 if there exists another facility j' such that

$$\|(a_i, b_i) - (\bar{x}_j, \bar{y}_j)\| - r_j > \|(a_i, b_i) - (\bar{x}_{j'}, \bar{y}_{j'})\| + r_{j'} \quad (3.13)$$

holds true. The inequality (3.13) implies that the closest possible distance between the location of customer i and the new location of facility j is strictly greater than the maximum possible distance between the location of customer i and the new location of facility j' . In this case, customer i has no chance to be allocated to facility j in the new solution. In addition, if the inequality (3.13) holds true for every j different from j' , then we fix $h_{ij'}$ to 1 as well.

Figure 3.1 illustrates the LS heuristic on a test instance with $n = 49$ customers and $p = 3$ facilities. The indices of the facilities are displayed in the figure to trace the location changes. The initial locations of the facilities, generated from the convex hull of the customers' locations uniformly at random, are displayed by the squares. Then, this initial solution is improved by the ALA heuristic resulting in the new solution displayed by the triangles. Lastly, we use the LS heuristic (with a fixed constant radius for all circles) to further improve the solution of the ALA heuristic yielding the final solution represented by the crosses. As seen in Figure 3.1, in the implementation of the LS, we first draw 3 circles around the facility locations obtained by the ALA heuristic and improve the ALA solution. Afterwards, we continue drawing circles around the new solutions until no improvement is obtained. Note that once a new solution is obtained by the LS, one can also use the ALA heuristic on the new solution, instead of continuing with the LS, to move it to a new local optimal solution.

In each iteration of the LS, an MISOCP problem is solved. If the search space, i.e., the total area searched inside the circles, is small, then it may not be possible for the LS to escape from the current local solution. If the search space is increased, then the MISOCP formulation will take longer to solve, but it will be more likely for the LS to move from the local solution to an improved solution. Depending on how much of the search space the user would like to cover, we propose a way to choose the radii of the circles in Section 3.3.

As solving MISOCP problems are expensive, the LS is likely to be more beneficial if it is applied after some other solution method, like the ALA heuristic, or if it is used within other heuristics as a subroutine. We next describe two other improvement heuristics for the MF-GWP, namely “transfer” and “decomposition”, which aim to improve the local solution obtained by the ALA. We will argue that the LS can naturally be used within the decomposition heuristic.

3.2.2.3 Transfer

The transfer heuristic, proposed in Brimberg & Drezner (2013), aims to improve the solution of the ALA heuristic, called as the initial solution here, by reallocating a customer to its second nearest facility. Following the reallocation, the ALA heuristic

is run on the new solution starting with the Location Step to get to a hopefully new local solution. After the reallocation, the obtained solution may be worse at first, but the final solution found by the ALA heuristic may still be a better solution (than the initial solution).

If the transfer heuristic succeeds to find a better solution than the initial solution in the first trial, then the procedure stops and the solution is updated. Otherwise, taking the initial solution, another customer is selected to be transferred to its second nearest facility in the second trial. The trials continue until an improvement is observed or at most a fixed number, say N_T , of trials are made. The customers are selected by the following procedure. For every customer i , the difference between its distance from the closest (currently allocated) facility and its distance from the second closest facility is calculated and called as Δ_i . The original version of the transfer heuristic is applied to the customers one by one in the order of non-decreasing Δ_i values.

We next explain how we implement the transfer heuristic for the MF-GWP which includes several modifications on the original approach. First, for the given solution, the algorithm checks if there are any empty facilities, i.e., facilities serving no customer. Assume that there are $k \geq 1$ such facilities. We then compute, for each non-empty facility j , the total amount of CO₂ emitted from the vehicles departing from it, called as e_j . k facilities having the largest e_j values are determined and from each such facility a single customer causing the highest CO₂ emission is selected and reallocated to one of the empty facilities. So, in this initial step, we do k many transfers at the same time. After all the transfers, initially the Location Step of the ALA heuristic is applied as a result of which the empty facilities become non-empty and each of them (possibly) gets closer to one of the facilities that cause a high amount of CO₂ emission. This may result in other reallocations of the customers during the following steps of the ALA heuristic and therefore may lead to a further reduction in the total amount of CO₂ emitted in the distribution system.

Second, after handling the empty facilities if there are any, the customers that are to be transferred to their second nearest facilities are selected. In this step, in addition to the Δ_i values used in the original approach, we also take the speeds of the vehicles sent to the customers into account. Our aim here is that if vehicle i is emitting a

Algorithm 2 Transfer Heuristic Part 1

- 1: **procedure** TRANSFER(G, p, X_0, D, N_T, N'_T)
 - 2: Let Z be the index set of the empty facilities.
 - 3: Let e_j be the total amount CO₂ emitted from vehicles departing from facility $j, j \in J \setminus Z$.
 - 4: Let $j_z, z \in Z$ be the indices of the $|Z|$ facilities with the highest emission values.
 - 5: For each $z \in Z$, select among the customers served by facility j_z the one that causes the highest amount of CO₂ emission and reallocate it to facility z .
 - 6: Call the ALA heuristic starting with the Location Step and let X_1 denote the set of locations of the facilities in the returned solution.
 - 7: $X \leftarrow X_1$.
 - 8: **for** $i \leftarrow 1$ to n **do**
 - 9: Let d_i^1 be the distance of customer i to its closest facility.
 - 10: Let d_i^2 be the distance of customer i to its second closest facility.
 - 11: Let $\Delta_i = d_i^2 - d_i^1$.
 - 12: **end for**
 - 13: Sort Δ_i 's in non-decreasing order and let $r_i^1, i \in I$ be the rank of customer i in the sorted list. Denote by S , the set of indices i of customers with $r_i^1 \leq N'_T$.
 - 14: For each $i \in S$, compute $w_i v_i$ and let r_i^2 be the rank of customer i when the N'_T customers are ordered according to non-increasing values of $w_i v_i$.
 - 15: Sort $r_i^1 \times r_i^2$'s where $i \in S$ in non-decreasing order and let r_i^3 be the rank of customer i in the ordered list.
-

Algorithm 2 Transfer Heuristic Part 2

```
16:   for  $k \leftarrow 1$  to  $N_T$  do
17:       Reallocate customer  $i$  having  $r_i^3 = k$  to its second closest facility.
18:       Call the ALA heuristic starting with the Location Step and let  $X_{k+1}$ 
        denote the set of locations of the facilities in the returned solution.
19:       if The the solution is not improved then
20:           Return the transfered customer to its original facility (i.e., take the
            locations of the facilities given in  $X_1$  and allocate each customer to its closest
            facility.).
21:       else
22:            $X \leftarrow X_{k+1}$ .
23:       break
24:       end if
25:   end for
26: end procedure
```

high amount of CO₂ due to its speed, a small decrease in this speed may result in a significant reduction in the emission. For this purpose, we first rank the customers in the order of non-decreasing values of Δ_i and let r_i^1 denote the rank of customer i . We then rank all the customers with r_i^1 values less than or equal to N'_T , where $N_T \leq N'_T$, in the order of non-increasing $w_i \times v_i$ values and let r_i^2 denote the resulting rank of customer i . Finally we rank the customers in non-decreasing $r_i^1 \times r_i^2$ values and denote by r_i^3 the resulting rank of customer i . The customers are transferred one after another in the order of non-decreasing r_i^3 values. For ease of reference, the notation specific to the transfer heuristic is summarized in Table 3.1.

The details of the proposed transfer heuristic are given in Algorithm 2. The inputs of the algorithm are the vector of the customers' locations, G , the number of facilities to be located, p , the vector of initial facility locations, X_0 , the vector of the customers' deadlines, D , the maximum number of trials that will be made, N_T , and an integer N'_T satisfying $N_T \leq N'_T$. Here N'_T is a parameter used to make sure that a customer with rank $r_i^1 > N'_T$ will not be considered in any of the trials. If N'_T is taken as n , then this parameter can be dropped from the algorithm.

Table 3.1: Notation used in the transfer heuristic

Notation	Description
N_T	Maximum number of trials in each repetition
d_i^1	Distance of customer i to its closest facility
d_i^2	Distance of customer i to its second closest facility
Δ_i	Difference between distance from closest and second closest facility for customer i , i.e., $\Delta_i = d_i^2 - d_i^1$
r_i^1	Rank of customer i when all customers are ordered in non-decreasing Δ_i values
N'_T	Number of candidate customers for transfer
r_i^2	Rank of candidate customer i when the candidate customers are ordered in non-increasing $w_i \times v_i$ values
r_i^3	Rank of candidate customer i when the candidate customers are ordered in non-decreasing $r_i^1 \times r_i^2$ values
rep_{max}	Maximum number of repetitions

In Algorithm 2, the empty facilities are handled between Steps 2 and 7. The r_i^1 values are computed between Steps 8 and 13. The r_i^2 values are computed in Step 14 and the order of the customers that will be transferred one after another is determined in Step 15. Finally between Steps 16 and 25, trials are made until an improving solution is found or a maximum number N_T of trials is reached. The output of the algorithm is the final locations of the facilities, X , and the vector of allocations of the customers (which is not explicitly stated in the algorithm).

Each call of Algorithm 2 is called a repetition. In the implementation of the transfer heuristic in our computational experiments, repetitions continue until no improvement is obtained or a maximum number of repetitions is reached (whichever occurs first). Therefore, we take the maximum number of repetitions of Algorithm 2, denoted by rep_{max} , as a parameter in our computational experiments reported in Section 3.3.

3.2.2.4 Decomposition

The decomposition heuristic, proposed in Drezner et al. (2016), is an improvement heuristic. It aims to improve a given initial solution by decomposing the set of customers into smaller groups. This is done by selecting a predetermined $\hat{p} < p$ number of facilities and taking the union of the customers allocated to these facilities. A smaller size (reduced) problem is solved with an exact or a heuristic approach by considering only these customers and as a result the selected $\hat{p}$ facilities are relocated (the locations may remain the same as well). If no improvement is obtained, then another trial is performed with another set of $\hat{p}$ facilities. Trials continue until some stopping criterion is met.

We apply the decomposition heuristic for the MF-GWP in the following way. After $\hat{p}$ facilities are selected, we use the LS to solve the reduced problem as an MISOCP problem by drawing circles only around the selected facilities. As solving an MISOCP problem is expensive, we take the $\hat{p}$ value as 2 in our computational experiments. In each iteration of the LS, if any improvement is observed, we apply the ALA heuristic on the new solution. Trials continue by selecting a new pair of facilities until exactly N_D many trials are made.

Table 3.2: Notation used in the decomposition heuristic

Notation	Description
$\hat{p}$	Number of facilities selected for the decomposition heuristic
β	Ratio of the area of the reduced search space to that of the original search space
$s(j_1, j_2)$	Number of potential customers whose allocations may change when the decomposition is applied for the facilities j_1 and j_2
N_D	Maximum number of trials

In the decomposition heuristic, the pairs of facilities are selected in the following way. First, a common value, r , is determined as the radii of the circles. Then for each pair of facilities $j_1 < j_2$, a score $s(j_1, j_2)$ is computed. For each customer of facility j_1 (j_2), if it has a chance to be assigned to facility j_2 (j_1) in the reduced problem, then the value of $s(j_1, j_2)$ is increased by 1. More formally, assuming that customer i is originally assigned to facility j_1 , if

$$\|(a_i, b_i) - (x_{j_2}, y_{j_2})\| - r \leq \|(a_i, b_i) - (x_{j_1}, y_{j_1})\| + r \quad (3.14)$$

holds true, then customer i 's allocation may change in the solution of the reduced problem, and hence $s(j_1, j_2)$ is increased by 1. After their computation, we sort the $s(j_1, j_2)$ values in non-increasing order, and in the first trial, the pair of facilities j_1 and j_2 having the largest $s(j_1, j_2)$ value is selected. Then the pair of facilities with the second largest $s(j_1, j_2)$ value is selected and trials continue until N_D trials are made. The idea behind using such a scoring function is as follows. If the number of customers that are close to the border of a pair of facilities is higher, the chance of improving the solution of the reduced problem may be greater as a slight change in the locations of the facilities may lead to a change in the allocations of more customers. Changes in the allocations, in turn, result in another change in the locations of the facilities followed by possibly reallocations of more customers during the steps of the ALA heuristic which is performed after the decomposition. Once the ALA heuristic converges to a local solution, another decomposition trial is performed by choosing the next pair of facilities. For ease of reference, the notation specific to the

decomposition heuristic is summarized in Table 3.2.

Algorithm 3 Decomposition Heuristic

```

1: procedure DECOMPOSITION( $G, p, X_0, D, r, N_D$ )
2:   For each pair of facilities  $j_1 < j_2$ , compute the score of the pair  $s(j_1, j_2)$ .
3:   Sort the scores in non-increasing order and let  $j_1^k$  and  $j_2^k$  be the pair of facilities having the  $k^{th}$  largest score.
4:   for  $k \leftarrow 1$  to  $N_D$  do
5:     Take the locations of the facilities given in  $X_{k-1}$  and allocate each customer to its closest facility.
6:     Let  $G_k$  be the set of customers served by facilities  $j_1^k$  and  $j_2^k$ .
7:     Solve the reduced problem with the customers in  $G_k$  as an MISOCP problem by drawing circles with radii  $r$  around facilities  $j_1^k$  and  $j_2^k$ .
8:     Let the solution be  $X_k$ .
9:     if  $X_{k-1} \neq X_k$  then
10:      Call the ALA heuristic and update  $X_k$ .
11:    end if
12:  end for
13:   $X \leftarrow X_k$ .
14: end procedure

```

The details of the proposed decomposition heuristic is presented in Algorithm 3. The inputs of the algorithm are the vector of the customers' locations, G , the number of facilities to be located, p , the vector of initial facility locations, X_0 , the vector of the customers' deadlines, D , the radii of the circles, r , and the maximum number of trials that will be made, N_D .

In Algorithm 3, the scores are computed for each pair of facilities in Step 2 and are sorted in Step 3. After selecting a pair of facilities, the reduced problem is solved in Step 7. If the locations of the facilities change, then the ALA heuristic is called in Step 10. The algorithm stops when N_D decomposition trials are performed. The output of the algorithm is the final locations of the facilities, X , and the vector of allocations of the customers (which is not explicitly stated in the algorithm).

3.3 Computational Results

In this section, we compare the exact and heuristic solution methods for the MF-GWP in terms of solution quality and time. The results of the computational experiments are presented on test instances generated from three data sets taken from the literature. These data sets are

1. the Lozano data set with $n = 49$ customers (Lozano et al. 1998),
2. the Ruspini data set with $n = 75$ customers (OR Library), and
3. the Bongartz data set with $n = 287$ customers (Bongartz et al. 1994).

In addition to the customer locations, w_i values are also provided in these data sets. In the Ruspini data set, each w_i is equal to 1. In the Bongartz data set, w_i 's are integer but not all the same. In the Lozano data set, however, the w_i values are fractional and add up to 1. We solve the instances of the Lozano data set with the original w_i values as well. Note that one can also multiply the w_i 's in the Lozano data set by a suitable number, say θ , to turn them into integers which inflates the objective function values by θ but does not change the optimal solution.

The computational experiments are run on a computer with 3 GHz speed and 16Gb RAM. Heuristics are coded in C++ while CPLEX 12.6 IBM (2012) with default parameters is used to solve the MISOCP problems.

The values of the parameters used in the emission calculations are the same as those provided in Section 2.3 assuming that the vehicles are homogenous.

To compare the performances of the proposed algorithms with that of the exact solution method, i.e., the MISOCP formulation given in Section 3.2.1, we first solve small size instances.

3.3.1 Small Size Instances

The small size instances are generated from the Lozano data set by taking a subset of the customers and fixing the number of facilities to be located. For each customer i ,

the big-M value in the MISOCP formulation, M_i , is taken as the maximum distance between customer i and the other customers. The instances are solved under three different deadline settings:

1. no deadline (no DL),
2. wide deadlines (wide DL), and
3. tight deadlines (tight DL).

For the no deadline setting, the deadlines are taken as infinity. For the other settings, the deadlines, i.e., the d_i values, are determined as follows. First, the instance with $n = 30$ customers and $p = 3$ facilities is solved with the MISOCP formulation without imposing any deadlines and the best solution obtained within 4 hours is taken. Denoting by t_i the time it takes vehicle i to reach customer i in the MISOCP solution obtained, d_i is taken as $t_i \times \delta_i$, where δ_i is chosen uniformly at random from the interval $[0.75, 1.25]$ for the wide deadlines setting, and from the interval $[0.5, 1]$ for the tight deadlines setting.

The run time of the solver on solving an instance is limited to 4 hours, i.e., 14400 seconds.

In the first set of computational experiments, the instances are solved using the MISOCP formulation with and without the SB constraints given in (3.10) under no deadline setting and the results are given in Table 3.3. The gap values for unsolved instances are the relative optimality gaps returned by CPLEX at the termination when the 4 hour time limit is reached. While the MISOCP formulation without the SB constraints is able to solve only 3 of the 8 instances to optimality within 4 hours, the formulation with the SB constraints is able to solve 5 of them to optimality. With the addition of the SB constraints, the computational times to solve the instances with $n = 20$ and $p = 3$, $n = 25$ and $p = 2$, and $n = 30$ and $p = 2$ decreased from 694 seconds to 512 seconds, from 62 seconds to 38 seconds, and from 2098 seconds to 1602 seconds, respectively. Moreover, instances with $n = 20$ and $p = 4$ and $n = 25$ and $p = 3$ are solved within 11920 and 8455 seconds, respectively, after the introduction of the SB constraints. These two instances are not solvable without the SB

Table 3.3: Performance of the MISOCP formulation in solving small size instances with and without the SB constraints

Instance		WO SB Constraints		With SB Constraints	
n	p	Time (s)	GAP (%)	Time (s)	GAP (%)
20	3	694	0.0	512	0.0
20	4	14400	18.2	11920	0.0
20	5	14400	45.8	14400	37.8
25	2	62	0.0	38	0.0
25	3	14400	2.7	8455	0.0
25	4	14400	46.9	14400	43.6
30	2	2098	0.0	1602	0.0
30	3	14400	39.3	14400	32.0

constraints within 4 hours. As the SB constraints are effective in reducing the computational times, they are always included in the MISOCP formulation in the following experiments.

In the second set of experiments, the performance of the MISOCP formulation is evaluated under three deadline settings. The results are displayed in Table 3.4. It can be seen from the table that the instances become easier to solve as the deadlines get tighter. While the MISOCP formulation is able to solve only 5 instances to optimality within 4 hours under no deadline setting, it is able to solve 7 instances to optimality under wide deadlines setting. The only instance that is unsolvable within 4 hours under wide deadlines setting is the instance with $n = 25$ and $p = 4$ for which the relative optimality gap at the termination is 10.4%. All of the instances are solved to optimality under tight deadlines setting. The decrease in the computational times (given in seconds in the table) is also notable as the deadlines get tighter. As an example, the instance with $n = 30$ and $p = 2$ is solved within 1602 seconds, 158 seconds, and 22 seconds under no deadline setting, wide deadlines setting, and tight deadlines setting, respectively.

For the instances that are solved to optimality within 4 hours, the (optimal) objective

Table 3.4: Performance of the MISOCP formulation in solving small size instances under different time limit settings

Instance		No DL		Wide DL			Tight DL		
n	p	OFV	Time	$\bar{v}$	OFV	Time	$\bar{v}$	OFV	Time
20	3	2.020	512	62.7	2.154	58	69.4	2.207	41
20	4	1.636	11920	58.4	1.704	868	63.2	1.740	470
20	5	NA	14400	58.3	1.392	3951	62.5	1.415	1844
25	2	3.435	38	95.4	4.675	10	125.4	6.289	9
25	3	2.711	8455	59.4	2.736	182	68.8	2.837	105
25	4	NA	14400	NA	NA	14400	66.5	2.284	4473
30	2	4.799	1602	86.7	6.440	158	123	8.829	22
30	3	NA	14400	61.1	3.972	4458	68.2	3.994	648

function values (OFV) in kg and the average speeds of the vehicles ($\bar{v}$) in km/h are also given in Table 3.4. The total CO₂ emission in the distribution system increases as the deadlines get tighter. This is expected, as in order to arrive at the customer on time, the vehicles may need to speed up resulting in higher emissions. Under no deadline setting, all the vehicles travel at the optimal speed, i.e., 55.2 km/h. As the deadlines get tighter, the average speeds of the vehicles increase. For example, for the instance with $n = 25$ and $p = 3$, the average speed increases from the optimal speed to 59.4 km/h under wide deadlines setting and further increases to 68.8 km/h under tight deadlines setting.

We compare the performances of the ALA heuristic, the local search (LS) heuristic, and the MISOCP formulation (with the SB constraints) on the generated small size instances under no deadline setting. For the LS heuristic which aims to improve the solution of the ALA heuristic, four alternatives are considered to see the effects of variable fixing and the use of smaller big-M values on the computational performances:

1. LS with no tuning, i.e., with standard big-M values and no variable fixing, (ALA-LS-N),

2. LS with the use of smaller big-M values (ALA-LS-M),
3. LS with variable fixing (ALA-LS-F), and
4. LS with both variable fixing and the use of smaller big-M values (ALA-LS-B).

In the LS heuristic, we choose the radius of each circle r in such a way that the ratio of the search area of the reduced search space to that of the original search space is equal to a predetermined constant β , i.e.,

$$\beta = \frac{p \times \pi r^2}{\text{Original Area}}.$$

For the ALA heuristic and the alternatives of the LS heuristic, 10 replications are performed. In each instance, the same 10 initial solutions each consisting of randomly generated p facility locations are used for the heuristics to be able to make a fair comparison. For the alternatives of the LS heuristic, β is taken as 0.2. Each iteration of the LS heuristic is given a 1 hour time limit. If the best solution found by CPLEX within 1 hour is not an improving solution, then the LS heuristic terminates. In Table 3.5, the average computational time and the average percent deviation (% dev) from the best solution out of 10 replications are displayed for the heuristics. Note that for each instance, the best solution will be the solution obtained by the MISOCP formulation if it is able to find the optimal solution within 4 hours. Otherwise, the best solution is obtained by taking the solution of the MISOCP formulation (not necessarily the optimal solution) and the heuristics' solutions for all 10 replications into account.

When the computational times of the heuristics are examined in Table 3.5, it can be seen that the ALA heuristic is the fastest one as expected. For the alternatives of the LS heuristic, ALA-LS-B is the fastest and ALA-LS-N is the slowest one. The variable fixing is more effective in reducing the computational time than the use of smaller big-M values. The best results for the LS heuristic are obtained by using smaller big-M values in addition to variable fixing.

When the average percent deviations are examined, it can be seen that the solutions found within 4 hours by the MISOCP formulation for the instances with $n = 20$ and $p = 5$ and $n = 25$ and $p = 4$ are not optimal as the percent deviations of these solutions are 1.6 and 4.8, respectively. That is, the best solutions for these 2 instances are obtained by the heuristics. Moreover, the solutions found by the ALA heuristic

Table 3.5: Performances of the MISOCP formulation, the ALA heuristic, and the LS heuristic with different alternatives

Instance		Time (s) (Deviation (%))					
n	p	Exact	ALA	ALA-LS-N	ALA-LS-M	ALA-LS-F	ALA-LS-B
20	3	512 (0.0)	0.5 (4.5)	135.4 (3.4)	20.3 (3.4)	3.5 (3.4)	1.4 (3.4)
20	4	11920 (0.0)	0.4 (4.6)	515.7 (1.8)	284.0 (1.8)	5.1 (1.8)	3.2 (1.8)
20	5	14400 (1.6)	0.5 (3.8)	1449.6 (1.0)	30.2 (1.0)	5.8 (1.0)	4.1 (1.0)
25	2	38 (0.0)	0.6 (1.7)	147.5 (0.0)	30.2 (0.0)	3.9 (0.0)	2.2 (0.0)
25	3	8455 (0.0)	0.5 (6.4)	6319.9 (1.2)	1394.6 (1.2)	17.1 (1.2)	8.1 (1.2)
25	4	14400 (4.8)	0.6 (9.3)	5769.8 (9.1)	2178.4 (6.1)	47.1 (5.1)	16.0 (5.1)
30	2	1602 (0.0)	0.4 (2.6)	1798.4 (0.2)	98.2 (0.0)	33.6 (0.0)	8.3 (0.0)
30	3	14400 (0.0)	0.5 (2.5)	6130.8 (1.1)	4383.4 (0.5)	229.0 (0.5)	11.0 (0.5)

are successfully improved by the LS heuristics as the average percent deviation values are smaller for the LS heuristics.

Table 3.6 displays the number of times (out of 10 replications) each heuristic finds the best solution. For the instance with $n = 20$ and $p = 3$, the ALA heuristic finds the optimal solution in only one replication. The alternatives of the LS heuristic are able to improve the ALA solutions and find the optimal solution in 6 replications. For the instances which are not solved to optimality within 4 hours by the MISOCP formulation, the values in parenthesis in Table 3.6 represent the number of times (out of 10 replications) each heuristic finds a solution which is at least as good as the solution provided by the MISOCP formulation. For the instance with $n = 20$ and $p = 5$, while the ALA heuristic is able to find the best solution in only one replication, it finds solutions that are at least as good as the solution of the MISOCP formulation in 5 replications. For the instance with $n = 25$ and $p = 2$, all 4 alternatives of the LS heuristic find the best solution (which is optimal) in all replications.

We investigate the effect of the value of β on the solution quality and time of the LS heuristic on the Lozano instance with $n = 49$ and $p = 4$. As the β values, we use 0.05 and 0.1 resulting in the radii values of 6.3 and 8.9, respectively. For both beta values, the same five initial solutions each consisting of randomly generated 4 facility loca-

Table 3.6: Comparison of the ALA heuristic and alternatives of the LS heuristic in terms of the number of times the best solution is found

Instance		# of Times Best (No Worse Than Exact) Solution Found				
n	p	ALA	ALA-LS-N	ALA-LS-M	ALA-LS-F	ALA-LS-B
20	3	1	6	6	6	6
20	4	1	6	6	6	6
20	5	1 (5)	6 (9)	6 (9)	6 (9)	6 (9)
25	2	2	10	10	10	10
25	3	3	9	9	9	9
25	4	3 (3)	3 (3)	4 (4)	5 (5)	5 (5)
30	2	0	7	9	9	9
30	3	2 (2)	3 (3)	4 (4)	4 (4)	4 (4)

Table 3.7: Performance of the LS heuristic with different β values for the problem instance with $n = 49$ and $p = 4$

Rep.	β	ALA OFV	ALA t	LS # Iter.	LS OFV	LS t	% Imp.
1	0.05	8.229	0.7	2	8.223	60.1	0.1
2	0.05	8.685	1.6	1	8.685	3.6	0.0
3	0.05	7.775	1.5	2	7.750	18.7	0.3
4	0.05	9.548	0.7	7	8.682	42.5	9.1
5	0.05	8.655	0.8	7	7.802	21.2	9.9
1	0.10	8.229	0.7	2	8.223	1950.5	0.1
2	0.10	8.685	1.6	3	8.659	5553.7	0.3
3	0.10	7.775	1.5	2	7.750	518.7	0.3
4	0.10	9.548	0.7	6	8.657	9457.5	9.3
5	0.10	8.655	0.8	6	7.750	608.7	10.5

tions are used. In other words, 5 replications are made with each beta value. In each replication, first, the random solution is improved by the ALA heuristic. Then the LS heuristic is run on the ALA solution. If an improved solution is found, then the LS heuristic is repeatedly run (with the same r value) on the newly obtained solution until no improvement is observed. The results of this computational experiment are given in Table 3.7. Here, the objective function values (in kg) obtained by the ALA and the LS heuristics are given in columns titled as “ALA OFV” and “LS OFV”, respectively. The column “% Imp.” displays the average percent improvement obtained over the ALA objective function value by the LS heuristic. The computational time (in s) of the ALA and the LS heuristics are given in columns titled as “ALA t” and “LS t”, respectively. The column “LS # Iter.” displays the number of LS iterations, i.e., the number of times circles are drawn around the facility locations, until no improvement is obtained.

When Table 3.7 is examined, it can be seen that the computational time of each of the replications increases by a factor of at least 27 when β is increased from 0.05 to 0.1. In contrast, we do not see such a huge increase in percent improvement values when β is increased. For each beta value, the percent improvement values are close to 10 in two replications. In the other replications, the percent improvement values are very small. It can be seen from the table that in its current form, the LS heuristic is expensive as even when β is equal to 0.05 the computational time of it is much larger than that of the ALA heuristic. For this purpose, in the following experiments, instead of using the LS heuristic by itself, we use it as a subroutine in the decomposition heuristic as discussed in Section 3.2.2.4.

We now compare the performances of the ALA heuristic, the transfer heuristic, and the decomposition heuristic on the Lozano data set with $n = 49$ customers and with different number of facilities $p = 3, \dots, 10$. Each replication of the ALA heuristic starts with a randomly generated initial solution. The solution obtained by the ALA heuristic, called as the ALA solution, is then taken as the initial solution of the transfer heuristic. Therefore the transfer heuristic can be considered to contain 2 stages, namely the ALA stage and the transfer stage. Similarly, the solution obtained by the transfer heuristic, called as the transfer solution, is taken as the initial solution for the decomposition heuristic. This is how we implement the decomposition heuristic

Table 3.8: Average computational time spent in different stages over 10 replications for the problem instances with $n = 49$

p	Av. ALA t	Av. TR t	Av. DE t	Av. Total t
3	0.93	5.57	36.04	42.54
4	0.90	6.23	55.58	62.70
5	1.32	7.54	8.11	16.97
6	1.30	7.97	7.59	16.86
7	1.39	7.44	7.58	16.41
8	1.76	9.35	5.51	16.62
9	1.74	10.11	6.37	18.23
10	1.84	9.00	5.81	16.66

from now on and therefore we say that the decomposition heuristic consists of the ALA stage, the transfer stage, and the decomposition stage. As we move from the randomly generated solution to the ALA solution and then to the transfer solution, and finally to the decomposition solution, the objective function value never increases.

For each instance, 10 replications are performed with different randomly generated initial solutions. The parameters N_T and N'_T used in the transfer stage are taken as 10 and 20, respectively. We take the maximum number of repetitions in the transfer stage as 4. After the transfer stage, we apply the decomposition stage with N_D equal to 3. The β value in the decomposition stage is taken as 0.15. To determine the customers' deadlines, the instance with $p = 3$ without any customer deadline is solved using the decomposition heuristic for 10 replications. Taking t_i from the best solution obtained among the 10 replications, d_i is calculated as $t_i \times \delta_i$, where δ_i is chosen uniformly at random from the interval $[0.5, 1]$ for each customer i . The obtained d_i values are used in all instances.

In Table 3.8, the average (over 10 replications) computational time spent in each stage (columns 2, 3, and 4), as well as the average total solution time (column 5), i.e., the sum of the computational times spent in the ALA, transfer, and the decomposition stages, are given. The table shows that the ALA stage is the fastest and the average

Table 3.9: Objective function values of the best solutions in different stages and average percent improvements obtained with respect to the solution of the previous stage over 10 replications for the problem instances with $n = 49$

p	Av. ALA %imp	Min ALA	Av. TR %imp	Min TR	Av. DE %imp	Min DE
3	53.41	9.88	2.30	9.88	0.25	9.88
4	52.67	7.77	2.98	7.75	0.61	7.75
5	60.15	6.49	2.78	6.48	0.27	6.48
6	51.52	5.49	6.80	5.35	1.88	5.35
7	51.79	4.95	5.77	4.77	2.15	4.73
8	53.58	4.35	3.01	4.33	2.18	4.30
9	54.02	3.96	4.58	3.92	2.03	3.91
10	54.49	3.77	4.66	3.71	3.02	3.69

computational time spent in the ALA stage increases with p . A similar increase in the average computational time can be more or less seen in the transfer stage. For the decomposition stage, the average computational times are the highest when p is 3 or 4. This can be attributed to the larger size reduced problems solved when p is small. Note, however, that for larger size instances (see Section 3.3.2), we did not observe such a behaviour in the decomposition stage.

Table 3.9 shows the objective function value of the best solution found (over 10 replications) in all stages and the average percent improvement in the objective function value as we move from one stage to the next. The columns titled as “Av. ALA %imp”, “Av. TR %imp”, and “Av. DE %imp” show the average percent improvement (over 10 replications) of the objective function value as we go from the random initial solution to the ALA solution, from the ALA solution to the transfer solution, and from the transfer solution to the decomposition solution, respectively. The columns titled as “Min ALA”, “Min TR”, and “Min DE” show the objective function value of the best solution found in the ALA, transfer, and decomposition stages, respectively. In the table, bold values indicate that the best solution found by a stage improves upon that found by the previous stage.

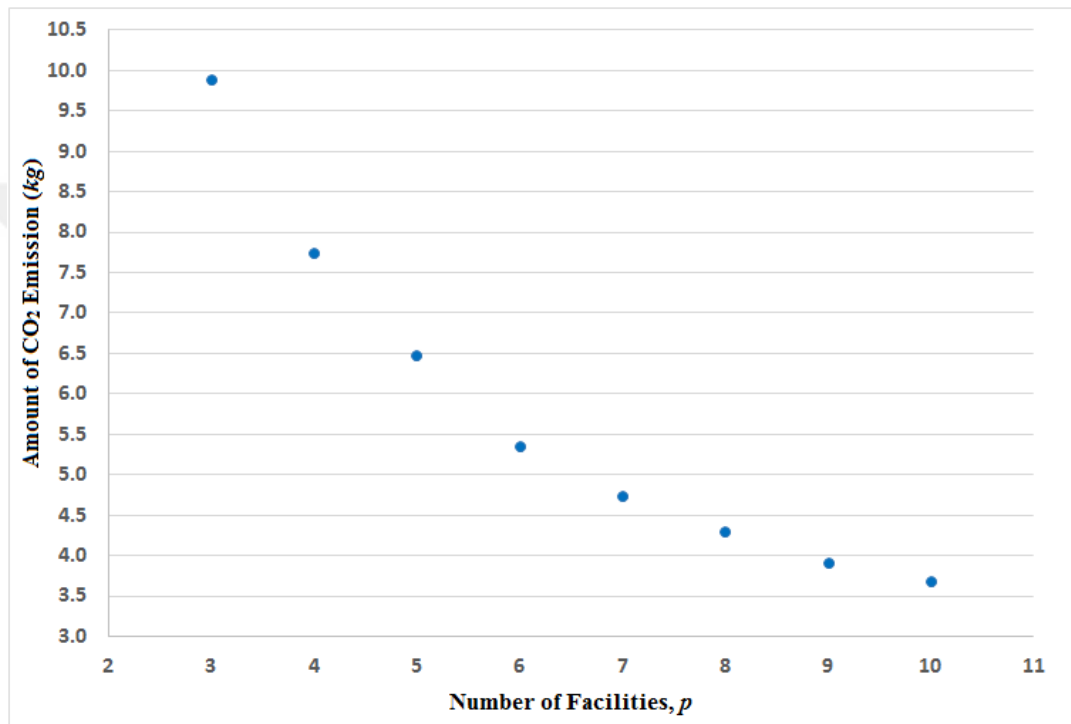


Figure 3.2: Amount of CO₂ emission of the distribution system with respect to the number of facilities for the problem instances with $n = 49$

Examining Table 3.9, we can observe that the best solution found during the ALA stage is improved by the transfer heuristic in most of the instances. On the other hand, in half of the instances, the best solution found during the transfer stage is improved by the decomposition heuristic. Note that the instances in which the decomposition improves the best transfer solution are the instances with the larger p values. Compared to the average percent improvements during the transfer stage, the improvements during the decomposition stage are smaller. Consider the problem instance with $p = 6$. Here, the average percent improvement the transfer stage has obtained on the ALA solutions is 6.80 and the best transfer solution (which has an ofv of 5.35) is better than the best ALA solution (which has an ofv of 5.49). While the decomposition stage has obtained a 1.88% improvement on the transfer solutions on average, it has found no better solution than the best transfer solution. In another instance with $p = 8$, the best solution of the decomposition stage (which has an ofv of 4.30) is better than that of the transfer stage (which has an ofv of 4.33).

For the problem with $n = 49$, when we investigate the amount of CO₂ emitted in the distribution system with respect to the number of facilities that are opened, it can be seen that the marginal reduction in the CO₂ amount for each increase in the number of facilities decreases as p gets larger (See Figure 3.2). A decision maker can compare the cost of opening an additional facility with the marginal reduction in the CO₂ amount emitted in the distribution system to determine the number of facilities that will be opened.

3.3.2 Medium and Large Size Instances

We provide the results of our computational experiments on instances generated from the Ruspini and Bongartz data sets by taking all of the customers, i.e., $n = 75$ for the former and $n = 287$ for the latter, and different p values. For the Ruspini and Bongartz data sets, the number of facilities, p , is taken from the sets $\{3, 4, \dots, 15\}$ and $\{3, 4, \dots, 10, 15, 20, 25, 30\}$, respectively. For these instances, we determine the β value by keeping the product of β and the number of customers, n , in different instances approximately the same as the product 0.15×49 we had for the instances with $n = 49$.

Similar to the last experiment done in Section 3.3.1, we compare three solution methods, namely the ALA heuristic, the transfer heuristic, and the decomposition heuristic, where the transfer heuristic has two stages (the ALA stage and the transfer stage) and the decomposition heuristic has three stages (the ALA stage, the transfer stage, and the decomposition stage).

For each data set, to determine the customers' deadlines, the instance with $p = 3$ without any customer deadline is solved using the decomposition heuristic for 10 replications. Taking t_i from the best solution obtained from the 10 replications, ℓ_i is set to $t_i \times \delta_i$, where δ_i is chosen uniformly at random from the interval $[0.5, 1]$ for each customer i . Once the deadlines are determined for a data set, they are used in all the instances generated from that data set.

In the computational experiments, 10 replications are performed for each instance in each data set. The parameters N_T and N'_T used in the transfer stage are set to 10 and 20, respectively, and the maximum number of repetitions in the transfer stage is taken as 2 for each instance of each data set. The decomposition parameters N_D and β are taken as 3 and 0.10, respectively, for the instances of the Ruspini data set and as 2 and 0.025, respectively, for the instances of the Bongartz data set.

We first report the computational results for the instances of the Ruspini data set. In Table 3.10, the average (over 10 replications) computational time spent in each stage (columns 2,3,4) and the average total solution time (column 5) are given. It can be seen from the table that as p gets larger, the average time spent in each stage increases in general. Note that the increase in the average time spent in the decomposition stage is less regular. Moreover, while the ALA stage is the fastest stage, the average time spent in the decomposition stage constitutes the biggest portion of the average total solution time.

Table 3.11 shows the objective function values of the best solution found (over 10 replications) in all stages and the average percent improvement in the objective function values as we move from one stage to the next. The column titles are the same as those in Table 3.9. Similarly, the bold values indicate that the best solution found by a stage improves upon that found by the previous stage.

Table 3.10: Average computational time spent in different stages over 10 replications for the problem instances with $n = 75$

p	Av. ALA t	Av. TR t	Av. DE t	Av. Total t
3	0.8	4.4	5.6	10.7
4	0.9	5.0	22.4	28.3
5	1.2	6.7	28.5	36.4
6	1.4	7.4	53.7	62.5
7	1.5	8.0	58.2	67.7
8	2.1	6.9	65.9	74.9
9	1.8	6.8	64.6	73.2
10	2.7	10.3	65.2	78.2
11	3.5	12.5	66.0	82.0
12	3.7	15.2	73.1	91.9
13	3.5	18.4	67.3	89.3
14	4.3	19.3	54.0	77.7
15	5.0	26.9	74.2	106.1

Table 3.11: Objective function values of the best solutions in different stages and average percent improvements obtained with respect to the solution of the previous stage over 10 replications for the problem instances with $n = 75$

p	Av. ALA %imp	Min ALA	Av. TR %imp	Min TR	Av. DE %imp	Min DE
3	76.4	756.5	1.3	756.5	0.0	756.5
4	67.1	528.2	6.5	417.9	18.3	417.9
5	78.8	364.9	2.4	364.9	7.6	364.9
6	82.4	317.5	1.7	317.5	8.5	317.1
7	88.0	282.5	12.9	282.1	0.3	282.0
8	76.9	260.4	7.8	260.4	0.7	260.4
9	81.7	256.2	14.9	239.7	1.4	239.5
10	68.8	239.6	13.8	225.2	1.0	225.1
11	79.6	232.4	20.5	204.3	1.0	204.3
12	79.9	251.7	22.3	195.3	1.5	195.3
13	79.5	223.8	19.2	189.4	0.9	187.9
14	75.7	232.4	25.3	176.4	0.6	175.9
15	75.1	214.9	27.6	171.5	1.2	171.4

Examining Table 3.11, we can observe that the best solution found during the ALA stage is improved by the transfer heuristic in 9 out of 13 instances. On the other hand, in 7 of the instances, the best solution found during the transfer stage is improved (by a small amount) by the decomposition heuristic. Compared to the average percent improvements during the transfer stage, the improvements during the decomposition stage are much smaller.

Note that the customer locations in the Ruspini data set are clustered into 4 groups. For this reason, if a random initial facility location falls outside of the clusters, it may remain empty in the ALA solution. The issue with the empty facilities are handled during the transfer stage, and therefore we observe huge average percent improvements by the transfer heuristic over the ALA solution for large p values. In the Bongartz data set, however, the customer locations are scattered on the plane and therefore the facilities are not likely to remain empty in the ALA solution. For this data set, the improvements obtained during the transfer stage will be mainly due to customer transfers to non-empty facilities.

Figure 3.3 displays how the amount of CO₂ emission changes as the number of facilities, p , increases. In this figure, we observe a huge decrease in the amount of CO₂ emission when p is increased from 3 to 4. This is in accordance with our expectations as there are 4 clusters of customers in this data set. In general, the marginal decrease in the emission amount as p is increased by 1 gets smaller as p gets larger.

As an additional experiment, for the instance with $p = 10$ of the Ruspini data set, 500 replications are performed. Figure 3.4 displays the histograms of the objective function values of the ALA, transfer, and decomposition solutions. The histogram clearly shows how the ALA solutions are improved by the transfer heuristic, and how the transfer solutions are improved by the decomposition heuristic. In this experiment, the average percent improvement of the transfer heuristic on the ALA solutions is equal to 14.8; and the the average percent improvement of the decomposition heuristic on the transfer solutions is equal to 1.3.

We report the computational results for the instances of the Bongartz data set. For these computational experiments, during each trial of the decomposition stage, a time limit of 2 minutes is imposed on the reduced MISOCP problem and the best solution

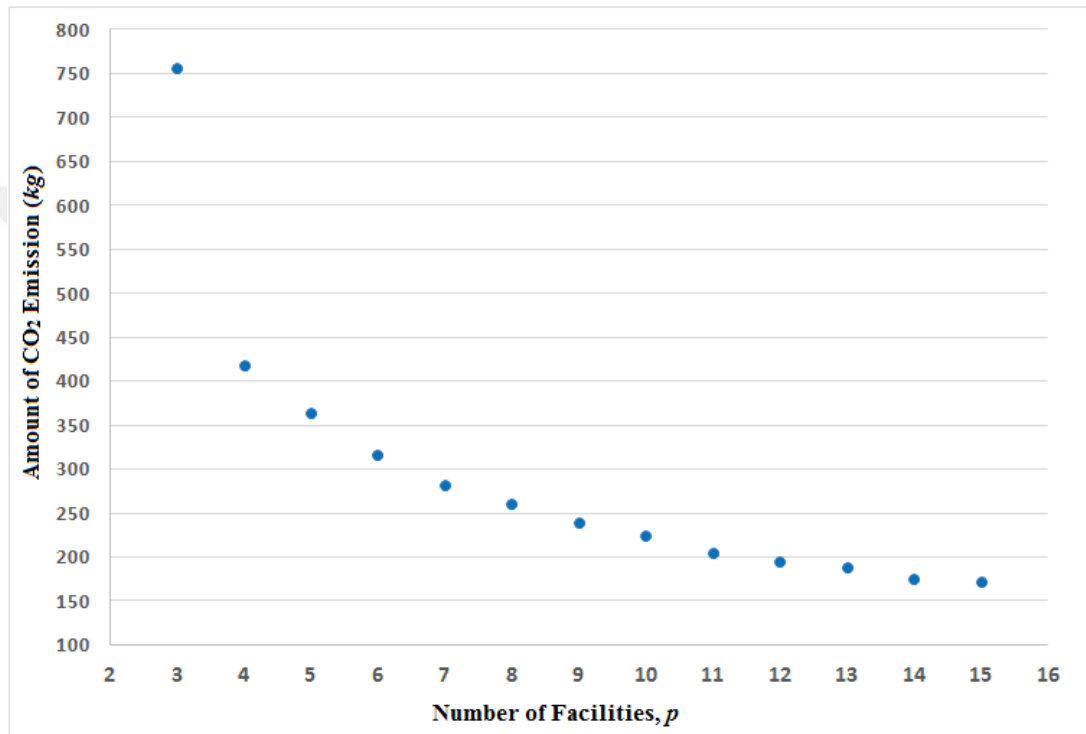


Figure 3.3: Amount of CO₂ emission of the distribution system with respect to the number of facilities for the problem instances with $n = 75$

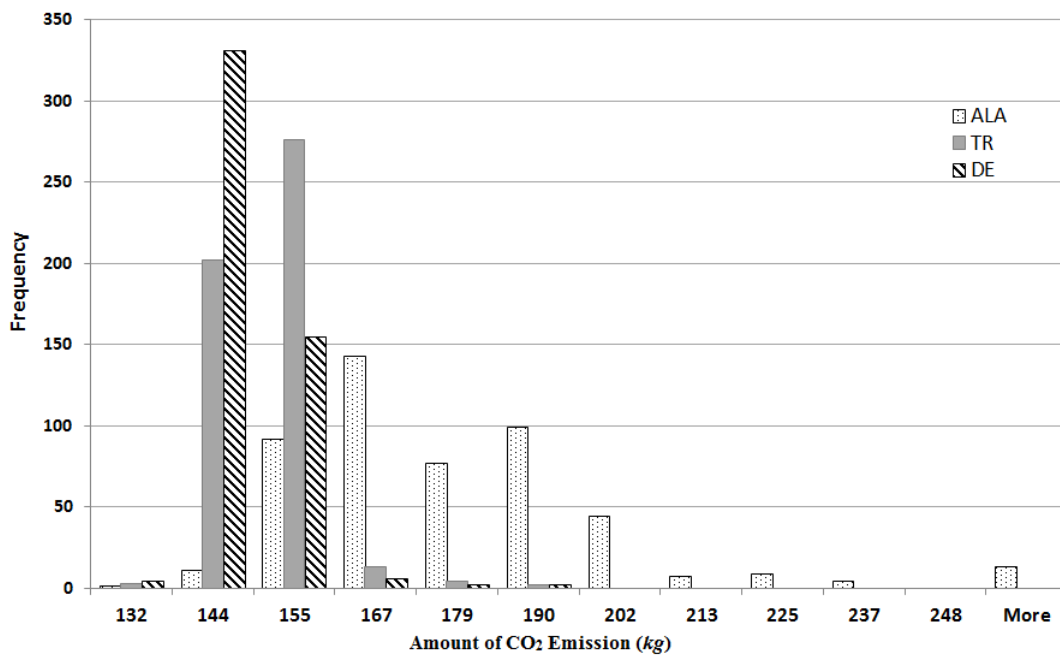


Figure 3.4: Histograms of the objective function values of the ALA, transfer, and decomposition solutions for 500 replications for the instance with $p = 10$ of the ruspini data set.

Table 3.12: Average computational time spent in different stages over 10 replications for the problem instances with $n = 287$

p	Av. ALA t	Av. TR t	Av. DE t	Av. Total t
3	2.8	8.4	240.9	252.1
4	5.0	11.8	241.5	258.3
5	6.5	15.2	242.5	264.2
6	9.8	18.2	243.5	271.5
7	11.9	28.4	244.3	284.6
8	14.9	30.4	243.8	289.1
9	19.6	34.9	243.6	298.1
10	21.7	48.5	242.3	312.5
15	53.2	91.3	250.8	395.3
20	94.7	102.4	253.2	450.3
25	200.1	288.2	259.4	747.7
30	244.1	394.9	289.9	928.9

obtained within the time limit is taken as the solution of the reduced problem. In Table 3.12, the average (over 10 replications) computational time spent in each stage (columns 2,3,4) and the average total solution time (column 5) are given. It can be seen from the table that as p gets larger, the average time spent in the ALA and the transfer stages increases. The increase in the average time spent in the decomposition stage is less notable because of the imposed time limit.

Table 3.13 shows the objective function values of the best solutions found (over 10 replications) in all stages and the average percent improvement in the objective function values over 10 replications as we move from one stage to the next. Examining Table 3.13, we can observe that the best solution found during the ALA stage is improved by the transfer heuristic in 9 out of 12 instances. On the other hand, in 8 of the instances, the best solution found during the transfer stage is improved by the decomposition heuristic. As an example, consider the instance with $p = 3$. For this example, the best solution found by the ALA, transfer, decomposition heuristics have objective function values of 5895.5, 5704.9, and 5597.5, respectively. Note that the average per-

Table 3.13: Objective function values of the best solutions in different stages and average percent improvements obtained with respect to the solution of the previous stage over 10 replications for the problem instances with $n = 287$

p	Av. ALA %imp	Min ALA	Av. TR %imp	Min TR	Av. DE %imp	Min DE
3	98.2	5895.5	1.3	5704.9	1.7	5597.5
4	96.8	5454.3	1.1	5378.6	2.1	5236.7
5	94.9	5001.8	1.2	4910.4	3.8	4740.1
6	88.3	4858.1	1.2	4549.7	6.4	4370.7
7	94.3	4068.0	1.2	4068.0	4.8	4061.3
8	89.9	3909.0	0.9	3909.0	1.8	3786.9
9	89.6	3698.9	4.3	3674.4	3.4	3674.4
10	93.5	3437.4	1.4	3437.4	0.8	3437.4
15	94.2	3021.5	3.2	2947.2	1.4	2821.1
20	89.9	2708.3	3.3	2569.6	1.3	2569.6
25	94.7	2016.0	2.1	1981.0	0.4	1981.0
30	92.7	2082.8	5.4	1958.4	1.5	1912.3

cent improvements obtained by the transfer heuristic on the ALA solutions are much smaller for this data set when compared with the Ruspini data set. This is because, in none of the replications, except one for $p = 30$, an empty facility is returned by the ALA heuristic. On the other hand, the average percent improvements obtained by the decomposition heuristic on the transfer solutions are notable.

We investigate the performances of the ALA, transfer, and decomposition heuristics when the total computational time is fixed. Here, we want to observe whether or not the best solution found by doing more replications of the ALA heuristic will be able to beat those found by doing less replications of the transfer and decomposition heuristics within a fixed time. Similarly, we want to compare the performances of the transfer and decomposition heuristics when the computational time is fixed. We present the computational results for the instances of the Ruspini and Bongartz data sets, where the total computational time for each instance is fixed to 10 and 60 min-

Table 3.14: Best Objective function value found by different heuristics for the problem with $n = 75$ and fixed 10 minutes solution time

p	ALA		TR		DE	
	# rep.	Min OFV	# rep.	Min OFV	# rep.	Min OFV
3	951	756.5 (0.0)	126	756.5 (0.0)	39	756.5 (0.0)
4	726	417.9 (0.0)	93	417.9 (0.0)	24	417.9 (0.0)
5	512	364.9 (0.0)	80	364.9 (0.0)	19	364.9 (0.0)
6	433	317.1 (0.0)	66	317.1 (0.0)	15	329.9(4.1)
7	373	282.0 (0.0)	68	282.0 (0.0)	12	282.0 (0.0)
8	306	260.4 (0.0)	63	260.8(0.2)	8	260.4 (0.0)
9	251	239.6 (0.0)	58	239.7(0.0)	7	240.4(0.3)
10	219	229.9 (2.2)	45	225.1 (0.0)	7	225.1 (0.0)
11	188	225.9(10.6)	35	204.3 (0.0)	8	204.3 (0.0)
12	167	211.0 (9.9)	34	192.0 (0.0)	7	195.3(1.7)
13	148	197.1 (5.8)	29	186.3 (0.0)	5	190.0(2.0)
14	127	194.6 (9.3)	24	178.0 (0.0)	6	180.3(1.3)
15	119	205.6(19.9)	20	171.5 (0.0)	4	174.3(1.6)

Table 3.15: Best objective function value found by different heuristics for the problem with $n = 287$ and fixed 60 minutes solution time

p	ALA		TR		DE	
	# rep.	Min OFV	# rep.	Min OFV	# rep.	Min OFV
3	749	5597.5 (0.0)	221	5597.5 (0.0)	12	5597.5 (0.0)
4	556	5020.6 (1.6)	170	4942.1 (0.0)	12	4942.4(0.0)
5	414	4492.7 (0.0)	135	4492.7 (0.0)	10	4532.0(0.9)
6	317	4258.0 (3.5)	104	4114.9 (0.0)	12	4200.4(2.1)
7	250	3938.2 (0.1)	83	3933.9 (0.0)	10	4084.0(3.8)
8	192	3723.1 (3.4)	64	3600.1 (0.0)	12	3813.4(5.9)
9	154	3525.2 (0.0)	54	3608.3 (2.4)	11	3696.4(4.9)
10	126	3497.2 (4.0)	43	3366.7 (0.1)	10	3362.3 (0.0)
15	57	2910.6 (7.0)	21	2720.9 (0.0)	8	2844.7(4.5)
20	30	2424.7 (0.6)	13	2490.4 (3.3)	6	2410.3 (0.0)
25	21	2270.1(16.5)	8	1948.7 (0.0)	5	1981.0(1.7)
30	13	2073.1(21.0)	6	1949.1(13.8)	4	1713.0 (0.0)

utes, respectively. In Tables 3.14 and 3.15, the columns under the title “ALA”, “TR”, and “DE” show the results when the instances are solved using only the ALA heuristic, transfer heuristic (which has 2 stages), and decomposition heuristic (which has 3 stages), respectively. For each instance and each heuristic, the replications continue until the fixed computational time is reached. The columns titled as “# rep.” give the number of replications performed by the corresponding heuristic within the fixed computational time. The columns titled as “Min OFV” give the objective function value of the best solution found by the corresponding heuristic among the replications performed within the fixed computational time. Note that, next to the objective function values of the best solutions found, we also report the percent deviation of the “Min OFV” from the best solution found by all the heuristics in parenthesis. The bold values in both tables indicate that the best solution for an instance is found by the corresponding heuristic. Note that the best solution can be found by more than one heuristic.

We first analyze the results for the Ruspini data set given in Table 3.14. It can be seen from the table that the ALA heuristic finds the best solution in 7 instances. These instances are the ones with small p values. The transfer heuristic is able to find the best solution in all but 2 instances. For these 2 instances, the percent deviations from the best solution found are very small. Similar to the ALA heuristic, the decomposition heuristic finds the best solution in 7 instances. Note, however, that for large p values the percent deviations for the decomposition heuristic are much smaller than those for the ALA heuristic. When averages of the percent deviations of the “Min OFV” from the best solutions found over all 13 instances are computed, it can be seen that the transfer heuristic has the smallest value which is 0.02, the decomposition heuristic has the second smallest value which is 0.85, and the ALA heuristic has the largest value which is 4.44.

Consider the results of the computational experiments for the instance with $p = 11$ in Table 3.14. Within 10 minutes, 188 replications of the ALA heuristic is performed resulting in the minimum objective function value of 225.9 kg. Within the same computational time, 35 replications of the transfer heuristic and 8 replications of the decomposition heuristic are performed resulting in the best objective function value of 204.3 kg for both. This shows that one can prefer a smaller number replications

of the transfer or decomposition heuristic over a larger number of replications of the ALA when especially p is large.

We next analyze the results for the Bongartz data set given in Table 3.15. Here, the ALA, transfer, and decomposition heuristics find the best solution in 3, 8, and 4 instances, respectively, within the 60-minute fixed computational time. While the relative performance of the ALA heuristic gets worse as p increases, it becomes better for the decomposition heuristic for larger p values. The averages of the percent deviations of the “Min OFV” from the best solutions found over all 12 instances are 4.81, 1.63, and 1.98 for the ALA, transfer, and decomposition heuristics, respectively.

From Tables 3.14 and 3.15, it can be concluded that the transfer heuristic has the overall best performance when the computational resources are fixed. Moreover, the decomposition heuristic may be an alternative to the transfer heuristic when the number of facilities is large.

3.4 Application Areas of the MF-GWP and an Illustrative Example

In Section 1.1, we described some application areas of the MF-GWP without going into the details. In this section, we show that several emission or energy consumption formulae in the literature resemble the one in (2.2) and therefore the solution approaches, in particular the MISOCP formulation proposed in Section 3.2, can be employed directly or after some problem specific modifications in order to tackle different planar facility location problems with emission or energy consumption considerations where deliveries are made by direct shipments.

The formula in (2.2) contains terms of the form $\frac{z}{v}$, z , and zv^2 which constitute the objective function of (GWP-NLP1) in Section 2.1.2. Thereby, the formulations (GWP-SOCP) in Section 2.1.2 and (MF-GWP-MISOCP) in Section 3.2.1 are obtained by linearizing techniques applied on the objective function of (GWP-NLP1) and rewriting the resulting constraints as SOCP constraints. These techniques can be also used to handle similar nonlinear terms found in different fuel and energy consumption and emission formulations. For example, for any integer k , one can show that an objective function containing terms of the form zv^k (multiplied with non-negative constants

for $k \in \mathbb{Z}/\{-1, 0\}$) can be recast as an SOCP formulation after linearization. The three terms we have in (2.2) are terms of the form zv^k with $k = -1, 0$ and 2 , respectively. We next discuss some fuel and energy consumption formulae from the literature where the terms in each formula are all of the form zv^k .

Aktürk et al. use in their study a fuel consumption formula for aircrafts which includes terms of the form $\frac{z}{v^3}$, $\frac{z}{v^2}$, zv , and zv^2 , see Equation (2) in Aktürk et al. (2014). The technical report written by Senzig and Cumper Senzig & Cumper (2013) features three fuel consumption formulae for helicopters each containing terms of the form $\frac{z}{v}$, z , zv , and zv^2 . In Tokekar et al. (2011), Tokekar et al. use an energy consumption formulation for car-like robots that contains terms of the form $\frac{z}{v}$, z , and zv . The study in Gürel et al. (2019) employs an energy consumption formula for material handling robots that is of the form

$$\mu zv^\gamma, \quad (3.15)$$

where μ is a constant that depends on the type of the robot, its load, and frictional forces and $\gamma \geq 1$ is a constant that represents the relation between the speed and the energy consumption of the robot. In all these mentioned areas, one can face with problems similar to the one studied in this paper, and therefore the proposed solution approaches may find use after problem specific adaptations. We next give another application area involving the use of rail-guided vehicles (RGVs) within an assembly line system, where we modify the MISOCP formulation proposed in this study in accordance with the specifications of the problem. Moreover, we carry out some computational experiments on an illustrative example and discuss the results.

Consider an assembly line where the feeding tasks are performed by a number of RGVs (not necessarily homogeneous) delivering the parts directly from certain locations, namely depots, to the stations. Each station is fed by a dedicated RGV within a given deadline and at a certain frequency. The deadlines naturally arise as a function of the load of the RGVs and the cycle time of the assembly line. If in each trip, an RGV carries a load that can satisfy the demand of a station for two cycles, then the deadline of the station would be twice the cycle time of the assembly line. The energy consumption of each RGV depends on its speed, load, and distance to the station it feeds. The aim of the MF-GWP here is to find the optimal locations of the depots while determining the speeds of the RGVs so as to minimize the total amount of en-

energy consumed by all RGVs. Figure 3.5 shows a schematic diagram of an assembly line where the boxes and circles represent the stations and their feeding points, respectively. The rectangular area in the middle is where the depots are allowed to be established. It is assumed that the RGVs serve the stations loaded and come back to the depots empty. Therefore the speed of an RGV when it is loaded may be different from the one when it is empty. We take the energy consumption of an RGV as in (3.15) with $\gamma = 3$, i.e., $\mu z v^3$. For this application, the MF-GWP is reformulated with some problem specific modifications. In this reformulation, we use some of the previously defined variables and parameters which are not redefined here. We assume that station i is served by RGV i and μ_{e_i} and μ_{l_i} are constant coefficients differentiating the energy consumption for the empty and loaded moves of RGV i , respectively. We denote by σ_j the capacity of depot j and by λ_i the sum of the loading time of the RGV i at the depot and the unloading time at station i . t_i represents the time it takes RGV i to go from depot to station i and τ_i represents the time it takes RGV i to come back to the depot. w_i and L_i represent the delivery frequency, and the load of RGV i , respectively. Moreover, we assume that the speed of RGV i is restricted to be in the interval $[\underline{v}_i, \bar{v}_i]$. We next give a mixed-integer nonlinear programming formulation for this problem which can be rewritten as an MISOCP problem by using the techniques described in Sections 2.1.2 and 3.2.

$$\begin{aligned} \min \quad & \sum_{i \in I} \mu_{e_i} w_i \frac{z_i^4}{\tau_i^3} + \sum_{i \in I} \mu_{l_i} w_i \frac{z_i^4}{t_i^3} \\ \text{subject to} \quad & \end{aligned} \tag{3.16}$$

$$q_{ij} \geq \|(x_j, y_j) - (a_i, b_i)\| \quad \forall i \in I, j \in J \quad (3.17)$$

$$z_i \geq q_{ij} - M_i(1 - h_{ij}) \quad \forall i \in I, j \in J \quad (3.18)$$

$$\sum_{j \in J} h_{ij} = 1 \quad \forall i \in I \quad (3.19)$$

$$\sigma_j \geq \sum_{i \in I} w_i L_i h_{ij} \quad \forall j \in J \quad (3.20)$$

$$(x_j, y_j) \in P_j \quad \forall j \in J \quad (3.21)$$

$$\bar{v}_i t_i \geq z_i \geq \underline{v}_i t_i \quad \forall i \in I \quad (3.22)$$

$$\bar{v}_i \tau_i \geq z_i \geq \underline{v}_i \tau_i \quad \forall i \in I \quad (3.23)$$

$$\ell_i \geq \tau_i + t_i + \lambda_i \quad \forall i \in I \quad (3.24)$$

$$h_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J \quad (3.25)$$

$$t_i, \tau_i \geq 0 \quad \forall i \in I \quad (3.26)$$

The objective function (3.16) minimizes the total energy consumption of RGVs considering their moves from depots to the stations and back from stations to the depots. Constraints (3.17)-(3.19) are the same as (3.3)-(3.5) in (MF-GWP-MISOCP). Constraints (3.20) are the capacity constraints on depots. Constraints (3.21), which are linear in the considered instance, restrict the possible location of facility j to the depots area. Constraints (3.22) and (3.23) enforce the speed limits on RGVs. Constraints (3.24) make sure that the deadlines of the stations are respected. Note that, one can also add the SB constraints, i.e., $x_1 \leq x_2 \leq \dots \leq x_p$, in order to reduce the symmetry inherent in this formulation. Moreover, if two depot locations, say depots j_1 and j_2 , are to be well-separated, a constraint of the form $x_{j_2} - x_{j_1} \geq \text{constant}$ can be added to the formulation.

We now provide the details of the illustrative example given in Figure 3.5. The coordinates of the feeding points of the stations and the corners of the depots area are given in the Appendix. For each i , the frequency w_i , in $\frac{1}{\text{hour}}$, of the deliveries to station i is generated uniformly at random from the interval $[10, 80]$. For each i , λ_i and ℓ_i are taken as 30 and $3600/w_i$ seconds, respectively. We assume that $\underline{v}_i = 0.2 \text{ m/s}$ and $\bar{v}_i = 2.0 \text{ m/s}$ for each RGV i . The constants μ_{e_i} and μ_{l_i} are chosen as 1 and 2, respectively, for every i . We assume that the loads of the RGVs are all the same, i.e., $L_i = L$ for all i . Solving this problem instance with 3 uncapacitated depots yields the

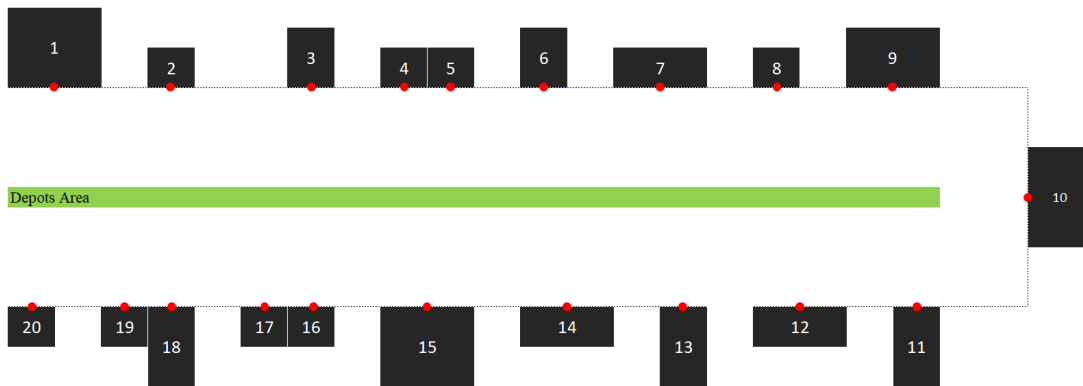


Figure 3.5: The layout of the assembly line system

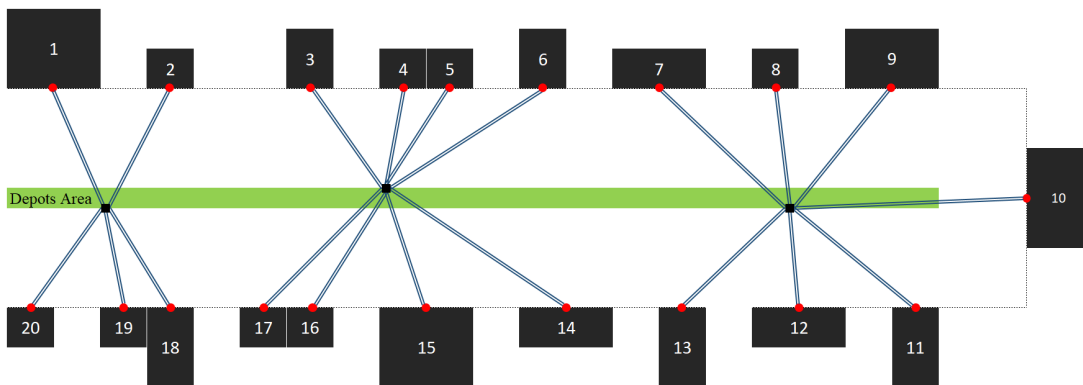


Figure 3.6: Solution of the uncapacitated problem

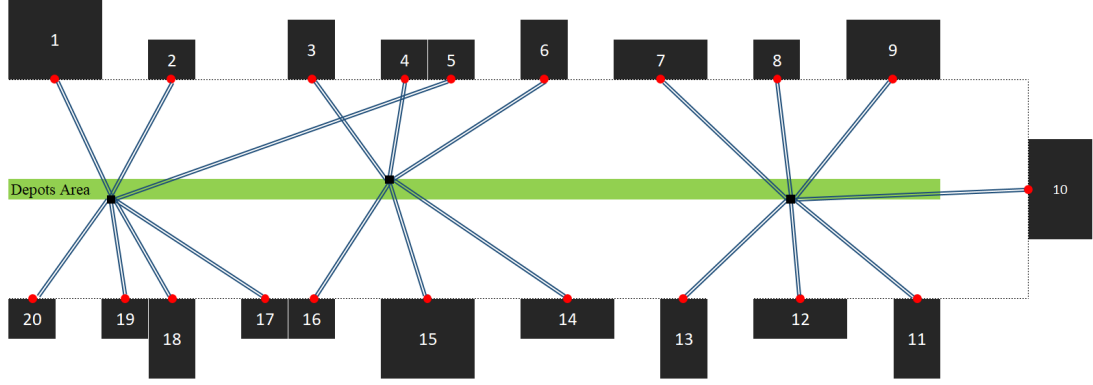


Figure 3.7: Solution of the capacitated problem with intersecting rails

solution shown in Figure 3.6, where the locations of the depots and the allocations of the stations to the depots are depicted. We next consider an identical capacity of $330L$ for each depot and solve the problem together with the capacity constraints given in (3.20). In this case, we obtain the solution given in Figure 3.7, where the total energy consumption of the RGVs increased by 2.5% with respect the solution of the uncapacitated case. Note that, this solution is not applicable since there are intersecting rails. One can handle this by adding problem specific constraints enforcing that the rails must not intersect. When we add the following constraints

$$h_{11} \geq h_{21} \geq \dots \geq h_{91}, \quad (3.27)$$

to our formulation, we obtain the solution shown in Figure 3.8. For this solution, we observe a 7.5% increase in the energy consumption of the RGVs when it is compared with the solution of the uncapacitated problem. Note that in all three cases, it took CPLEX less than 3 seconds to arrive at the optimal solution.

For the solution of the uncapacitated problem, the average speed of the loaded and empty RGVs turn out to be 0.88 m/s and 1.03 m/s , respectively. On the other hand, for the final solution obtained for the capacitated problem, these values are found as 0.93 m/s and 1.09 m/s , respectively.

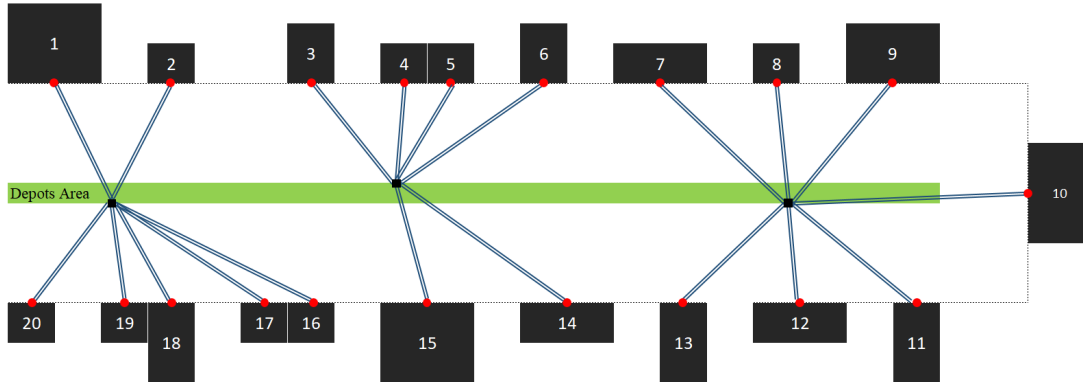


Figure 3.8: Solution of the capacitated problem with no intersecting rails

3.5 Concluding Remarks

In this chapter, we provided an MISOCP formulation for the multi-facility green Weber problem, a facility location problem with environmental considerations. After investigating some features of the formulation, we develop heuristic solution methods to solve larger problem instances in reasonable time. The proposed solution approach combines a local search heuristic, and modified versions of the transfer and decomposition heuristics as the improvement stages following the location-allocation procedure. The computational results show the effect of each stage on improving the solution quality. The results also indicate that within a fixed computational time, even though the location-allocation heuristic is able to make more replications, the improvement heuristics considered, i.e., transfer or transfer followed by decomposition, usually find better solutions using less number of replications. We also investigate how the total amount of CO₂ emitted by the distribution vehicles changes with respect to the number of facilities located. Such considerations are helpful when the number of facilities is not fixed a priori.

Table 3.16: Coordinates and deadlines of the stations in the illustrative example

Station	x	y	Deadline (s)
1	6	45	124.1
2	21	45	189.5
3	39	45	48.0
4	51	45	46.8
5	57	45	360.0
6	69	45	58.1
7	84	45	80.0
8	99	45	73.5
9	114	45	120.0
10	132	28.5	116.1
11	117	12	97.3
12	102	12	46.2
13	87	12	75.0
14	72	12	80.0
15	54	12	76.6
16	39	12	171.4
17	33	12	156.5
18	21	12	112.5
19	15	12	45.6
20	3	12	59.0

Table 3.17: Coordinates of the corners of the depots area in the illustrative example

Corner	x	y	Position
1	0	27	Southwest
2	0	30	Northwest
3	120	30	Northeast
4	120	27	Southeast



CHAPTER 4

ENERGY MINIMIZING FORKLIFT ROUTING PROBLEM (EMFRP)

The material handling systems used in warehouses involve important operations such as the usage of order picker forklifts. These order picker forklifts provide a more efficient utilization of the limited storage space by their ability to move in narrow aisles and pick items from high level racks. Routing the order picker forklifts to pick ordered items belongs to the operational decision making level and is done in high frequency. Therefore, finding an energy-efficient route for an order picker forklift can yield significant savings in the energy consumption in warehouses and the resulting CO₂ emission. In this paper, we introduce and study the energy minimizing order picker forklift routing problem (EMFRP) which aims to find an energy-efficient route for an order picker forklift to pick a given list of items. To our knowledge, this is the first study that considers the high-tech order picker forklifts in the context of the order picking problem. We calculate the forklift's energy consumption in both horizontal and vertical moves considering the effects of friction forces, the acceleration and deceleration of the forklift, and its load. A mixed integer programming formulation and a dynamic programming approach are developed to solve the EMFRP exactly. Since exact solution approaches are only able to solve small size instances to optimality (within a given time limit), we provide some tour construction and tour improvement heuristics for the problem and integrate them into a single solution approach. Computational results show that the proposed solution approach finds high quality solutions. Moreover, it is observed that significant energy savings can be achieved by solving the EMFRP instead of the classical distance minimization problem.

4.1 Notation and Problem Description

In this section, we provide the notation, problem description, and methodology used to evaluate the amount of energy consumed in a tour of an order picker forklift. Given the locations of a number of ordered items, the EMFRP deals with finding the most energy-efficient tour of the forklift that starts from the depot location, picks the ordered items from their locations, and brings them to the depot location. The assumptions of the EMFRP are:

- all of the ordered items are to be picked in a single tour and the features of the items are such that their mass and volume do not violate the specifications of the forklift,
- the order picker forklift can be ridden horizontally while its fork is not lowered to the floor,
- the constant speed of the order picker forklift in every horizontal movement is identical and the constant vertical speed of the fork in every lifting and lowering action is identical (these constant speeds are achieved after acceleration),
- there is no due time for bringing the ordered items to the depot location.

Some inputs of the EMFRP regarding the ordered items to be picked are: the number of items to be picked, n ; the horizontal distance between each pair of items i and j , d_{ij} ; the height of the location of item i , h_i ; and the mass of item i , l_i . The other parameters used in formulating the EMFRP are described in Section 4.2.1.

Now we provide some calculations to recognize the energy consumption of every movement of the order picker forklift while going from an item location to the next one in detail. We will use these calculations in the next section to come up with an MIP formulation in order to find an order picking tour that minimizes the total energy consumption of the order picker forklift.

We use the basic physics formulations Meriam & Kraige (2012) for energy consumption calculations in each horizontal and vertical movement of the order picker forklift. Unless otherwise stated, in all the statements in this thesis, the units are meter (m) for distance/displacement, second (s) for time, m/s for speed, m/s^2 for acceleration and

deceleration, Newton (N) for force, kilogram (kg) for mass, and joule (J) for work and energy amount.

Suppose that an object at rest having mass m_i is to be moved from location i to location j , where d_{ij} represents the horizontal distance between them. We assume that the object accelerates uniformly until it reaches a speed of v_x after a distance of d_a and travels at this speed until it is d_d close to its destination. After this point, it decelerates uniformly until it stops at location j . Moreover, the horizontal kinetic friction coefficient is assumed to be equal to μ during this travel.

For this purpose, first an acceleration force F_x^a is needed to make the object reach the speed of v_x after a distance of d_a beating the kinetic frictional force meanwhile. This force can be computed as

$$F_x^a = m_i a_x^+ + m_i g \mu, \quad (4.1)$$

where a_x^+ represents the acceleration which is given by $a_x^+ = v_x^2 / (2d_a)$.

Once the object reaches the speed of v_x , it moves at constant speed until it is d_d close to its destination. In this part of the travel, to beat the kinetic frictional force, an amount of force F_x^c is needed which is given by

$$F_x^c = m_i g \mu. \quad (4.2)$$

When the object is d_d close to its destination, an amount of force F_x^d is needed to stop the object. F_x^d can be calculated as

$$F_x^d = m_i a_x^- - m_i g \mu, \quad (4.3)$$

where a_x^- is the deceleration of the object that can be computed as $a_x^- = v_x^2 / (2d_d)$.

Thereby, the amount of energy/work, W_x , required for a horizontal movement of an object having mass m_i from location i to location j is obtained by multiplying the needed forces by the corresponding distances as in the following equation.

$$\begin{aligned} W_x &= d_a (m_i a_x^+ + m_i g \mu) + (d_{ij} - d_a - d_d) (m_i g \mu) + d_d (m_i a_x^- - m_i g \mu) \\ &= m_i (v_x^2 + g \mu (d_{ij} - 2d_d)) \end{aligned} \quad (4.4)$$

Suppose that an object at rest having mass m_i is to be moved from location i to location j , where $h_{ij} = h_j - h_i$ represents the vertical distance between them. Here h_i and h_j represent the heights of locations i and j , respectively. We assume that h_{ij} is positive and $d_{ij} = 0$, and therefore only an upward movement is to take place.

We assume that the object accelerates uniformly until it reaches a speed of v_y after a distance of h_a^u and travels at this speed until it is h_d^u close to its destination (note that h_d^u is a function of the other parameters). After this point, it decelerates uniformly until it stops at location j . Moreover, the kinetic frictional force during this vertical movement is assumed to be equal to F_r .

For this purpose, first an acceleration force of F_y^a is needed to make the object reach the speed of v_y after height of $h_i + h_a^u$ beating the gravity force and the kinetic frictional force meanwhile. F_y^a can be calculated as

$$F_y^a = m_i (a_y^+ + g) + F_r, \quad (4.5)$$

where a_y^+ represents the acceleration that is given by $a_y^+ = v_y^2 / (2h_a^u)$ and g represents the gravitational acceleration.

Once the object reaches the speed of v_y , an amount of force F_y^c is needed to keep it moving at constant speed until it is h_d^u close to its destination. F_y^c can be computed as

$$F_y^c = m_i g + F_r. \quad (4.6)$$

When the object is h_d^u close to its destination, eliminating the lifting force makes the object stop after displacement of $h_d^u = m_i v_y^2 / (2(m_i g + F_r))$. Thereby, the required energy/work, W_y^u , for lifting an object with mass m_i from location i (lower) to location j (higher) is obtained by multiplying the needed forces by the corresponding displacements as given in the following equation.

$$\begin{aligned} W_y^u &= h_a^u (m_i a_y^+ + m_i g + F_r) + (h_{ij} - h_a^u - h_d^u) (m_i g + F_r) \\ &= h_a^u (m_i a_y^+) + h_{ij} (m_i g + F_r) - h_d^u (m_i g + F_r) = h_{ij} (m_i g + F_r) \end{aligned} \quad (4.7)$$

Finally, suppose that an object at rest having mass m_i is to be moved from location i to location j , where h_{ij} is negative and $d_{ij} = 0$, and therefore only a downward movement is to take place. We assume that the object accelerates uniformly by gravity until it reaches the speed of v_y after a distance of h_a^d and travels at this speed until it is h_d^d close to its destination (note that h_d^d is a function of the other parameters). After this point, it decelerates uniformly until it stops at location j .

With no external force, the object reaches the speed of v_y by the gravity force after displacement of $h_a^d = m_i v_y^2 / (2(m_i g - F_r))$. After the object reaches the speed of

v_y , an amount of force $F_y'^c$ is needed to keep the speed constant that is given by

$$F_y'^c = m_i g - F_r. \quad (4.8)$$

Once the object is h_d^d close to its destination, a deceleration force of $F_y'^d$ is needed to make the object stop at location j beating the gravity force. This deceleration force is given by

$$F_y'^d = m_i (a_y^- + g) - F_r, \quad (4.9)$$

where $a_y^- = v_y^2 / (2h_d^d)$.

Thereby, the amount of energy/work, W_y^d , required for moving an object having mass m_i from location i (higher) to location j (lower) is obtained by multiplying the needed forces by the corresponding distances as

$$\begin{aligned} W_y^d &= h_d^d (m_i a_y^- + m_i g - F_r) + (h_{ij}' - h_a^d - h_d^d) (m_i g - F_r) \\ &= h_d^d (m_i a_y^-) + h_{ij}' (m_i g - F_r) - h_a^d (m_i g - F_r) = h_{ij}' (m_i g - F_r), \end{aligned} \quad (4.10)$$

where h_{ij}' represents the absolute height difference between location i and location j , i.e., $h_{ij}' = |h_{ij}|$.

Equations 4.4, 4.7, and 4.10 give the amount of energy consumption resulting from moving an object horizontally, upward, and downward, respectively. We use these calculations in the exact and heuristic solution methods proposed for the EMFRP in Sections 4.2 and 4.3.

4.2 Exact Solution Approaches for the EMFRP

In this section, we provide two exact solution approaches for the EMFRP where the energy consumption calculations presented in the previous section are used. First, we propose an MIP formulation for the EMFRP which has some characteristics of the classical TSP formulations. Note that the EMFRP is different from the classical TSP, since the energy consumption resulting from a travel from one location to another depends on the locations that are previously visited due to the dependence of the energy consumption on the load carried by the order picker forklift. With the MIP formulation, small size instances of the EMFRP can be solved easily by any MIP solver. We

then propose a dynamic programming solution approach that is an adaptation of the Bellman-Held-Karp algorithm that is developed for the TSP.

4.2.1 An MIP Formulation

In this section, we present an MIP formulation for the EMFRP. We first give the necessary notation to formulate the EMFRP. We denote by n the number of items to be picked and by $I = \{0, 1, 2, \dots, n\}$ the set of all location indices where 0 represents the depot and other indices represent the items. When we say item i , we refer to the item that is in location i . The parameters of the EMFRP are as follows.

L_f : The sum of masses of the fork and operator.

L_t : The sum of masses of the forklift and operator.

l_i : The mass of item $i \in I \setminus \{0\}$.

d_{ij} : The horizontal distance between location $i \in I$ and location $j \in I$.

h_{ij} : The vertical height difference between location $i \in I$ and location $j \in I$, i.e.,
 $h_{ij} = h_j - h_i$.

h'_{ij} : The absolute vertical height difference between location $i \in I$ and location $j \in I$, i.e., $h'_{ij} = |h_j - h_i|$.

a_{ij} : Binary parameter that is equal to 1 if and only if the horizontal distance between location $i \in I$ and location $j \in I$ is greater than 0.

v_x : The horizontal speed of the forklift.

v_y : The vertical speed of the fork.

g : The gravitational acceleration.

μ : The horizontal kinetic friction coefficient.

F_r : The constant kinetic friction force in vertical moves of the fork.

The variables used to formulate the EMFRP are as follows.

x_{ij} : Binary variable taking the value 1 if and only if the forklift travels to location $j \in I$ right after location $i \in I$.

f_{ij} : Energy consumption of the horizontal move from location $i \in I$ to location $j \in I$ if $x_{ij} = 1$ (0 otherwise).

u_{ij} : Energy consumption of the vertical move from location $i \in I$ to location $j \in I$ if $x_{ij} = 1$ (0 otherwise).

q_i : Total mass of the items on the forklift right after leaving location $i \in I$.

After introducing the necessary parameters and variables, we next provide an MIP formulation for the EMFRP.

(EMFRP-MIP)

$$\min \sum_{i \in I} \sum_{j \in I} f_{ij} + \sum_{i \in I} \sum_{j \in I} u_{ij} \quad (4.11)$$

subject to

$$\sum_{j \in I} x_{ij} = 1 \quad \forall i \in I \quad (4.12)$$

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in I \quad (4.13)$$

$$q_j + M(1 - x_{ij}) \geq q_i + l_j \quad \forall i, j \in I, j \neq 0 \quad (4.14)$$

$$f_{ij} + M'_{ij}(1 - x_{ij}) \geq a_{ij}(q_i + L_t)(v_x^2 + g\mu(d_{ij} - 2d_d)) \quad \forall i, j \in I \quad (4.15)$$

$$u_{ij} + M''_{ij}(1 - x_{ij}) \geq h'_{ij}(q_i + L_f)g + h_{ij}F_r \quad \forall i, j \in I \quad (4.16)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in I \quad (4.17)$$

$$q_i \geq 0 \quad \forall i \in I \quad (4.18)$$

$$f_{ij}, u_{ij} \geq 0 \quad \forall i, j \in I \quad (4.19)$$

The objective function of (EMFRP-MIP) minimizes the total amount of energy consumed by the order picker forklift through its complete tour. The constraints in (4.12) and (4.13) make sure that the forklift visits every location exactly once. Constraints in (4.14) make the value of q_j which is the total mass of the items on the forklift right after picking item j equal to the sum of the mass of item j and the total mass of the items on the forklift right after picking item i in case $x_{ij} = 1$ in an optimal solu-

tion. This set of constraints prevents the presence of subtours as well. Constraints in (4.15) and (4.16) are used to obtain the amount of energy consumed by the forklift in horizontal and vertical moves, respectively, when traveling to location j right after location i . Constraints in (4.17) are used to force that x_{ij} is a binary variable for each $i \in I, j \in J$. Constraints in (4.18) and (4.19) are the non-negativity constraints for the mass and energy variables. We can take the values of M , M' , and M'' in (EMFRP-MIP) as

$$\begin{aligned} M &= \sum_{i \in I} l_i, \\ M'_{ij} &= a_{ij} (M + L_t) (v_x^2 + g\mu (d_{ij} - 2d_d)), \\ M''_{ij} &= h'_{ij} (M + L_f) + h_{ij} F_r. \end{aligned}$$

Here, M is equal to sum of the loads of the items. This value for M makes q_j and q_i independent of each-other in constraints (4.14) if $x_{ij} = 0$. Values for M'_{ij} and M''_{ij} are taken in such a way that f_{ij} and u_{ij} will be equal to zero in constraints (4.15) and (4.16), respectively, if $x_{ij} = 0$.

4.2.2 A Dynamic Programming Solution Approach

The EMFRP has some characteristics of the TSP. In this section, we adapt a dynamic programming (DP) solution approach, namely the Bellman-Held-Karp algorithm Bellman (1962), Held & Karp (1962), that is originally proposed for the TSP to the EMFRP. The Bellman-Held-Karp algorithm runs in $O(2^n n^2)$ time to solve the TSP. We implement this solution method on the EMFRP as follows. Let 0 be the starting point of the tour, i.e., location of the depot. For any pair of items (i, j) , let $E_{ij}(Q)$ denote the amount of energy consumed by the forklift to travel from location i to location j with load Q . For any subset of the items S and for any $t \in S$, let $opt(S, t)$ denote the minimum amount energy consumed by the forklift to start from the depot and pick the items in $S \setminus \{t\}$ and then finally pick the item t . We have

$$opt(S, t) = \min_{j \in S \setminus \{t\}} (opt(S \setminus \{t\}, j) + E_{jt}(Q_S - q_t)). \quad (4.20)$$

where $Q_S = \sum_{s \in S} q_s$.

Thereupon, if we let N denote the set of all items to be picked, then the optimal value

of the EMFRP is

$$E^* = \min_{t \in N} (opt(N, t) + E_{t0}(Q_N)). \quad (4.21)$$

Observe that for all $j \in N$ we have $opt(\{j\}, j) = E_{0j}(0)$. Starting with these values, the recursive equation (4.20) is used to build the values $opt(S, t)$ for all $S \subseteq N$ and $t \in S$, working our way through sets with two elements, then sets with three elements, and step by step up to the full set N . Once we have the values of $opt(N, t)$ for all $t \in N$, we use (4.21) to find E^* . This algorithm runs for the EMFRP in $O(2^n n^2)$ time, too.

We coded this DP solution approach in C++ and show its performance on solving different instances of the EMFRP in Section 4.5.

4.3 Heuristic Solution Approaches for the EMFRP

The exact solution approaches proposed for the EMFRP in the previous section are unable to solve large size problem instances to optimality as is shown in Section 4.5. In this section, we propose some heuristic algorithms for the EMFRP to obtain good quality solutions for larger size instances. The considered heuristics mainly fall into two groups; namely, tour construction heuristics and tour improvement heuristics. The construction algorithms are used to obtain good quality initial solutions (routes) which are then improved by the improvement heuristics. Some of the provided approaches in this section are mainly developed for the TSP in the literature and we adapt them for the EMFRP. In addition, two matheuristic approaches, one construction and one improvement heuristic, are also developed for the EMFRP. Finally, we bring all these heuristics together to propose a single solution approach for the EMFRP.

4.3.1 Tour Construction

In our computational experiments with the heuristics, the initial tour of the order picker forklift is constructed using different methods. One way to construct the initial tour is by randomly generating the sequence of the items to be picked. In addition,

two other construction heuristics are proposed in our study; one is an adaptation of the nearest neighbor heuristic proposed for the TSP in the literature and the other is a matheuristic called as DP-Construction algorithm developed for the EMFRP where the presented DP algorithm is used as a subroutine.

4.3.1.1 TSP-based Nearest Neighbor Algorithm

This construction algorithm is based on the nearest neighbor algorithm proposed for the TSP in the literature. Here, we construct a tour of the forklift in a greedy manner. Starting at the depot location, we first move to the item location which results in the lowest energy consumption for the forklift. Any time determining the next item location to visit, among the items that are not picked yet, the one resulting in the least energy consumption is chosen. For each travel from an item location to another one, the energy consumption of the forklift is evaluated based on the calculations given in Section 4.1 considering the horizontal and vertical moves, as well as load, acceleration, deceleration, and friction forces. After determining the last item to pick, the tour is completed by returning at the depot location. The obtained solution (tour) is then taken as the initial solution for the improvement heuristics.

4.3.1.2 DP-Construction Algorithm

The DP-Construction algorithm is a matheuristic that is used as an alternative tour construction algorithm. The method is initialized with an incomplete tour that starts and ends at the depot location. The middle part of this incomplete tour is detached and the detached items together with some new items are added to the middle part to make the incomplete tour grow. To determine the ordering of the items that are to be added to the incomplete tour, the DP approach is used as an optimization tool. The incomplete tour turns into a complete one once all the items are added at which point the algorithm stops.

The steps of the DP-Construction algorithm are displayed in Figure 4.1. In the first step, K items that are closest (in terms of horizontal + vertical distance) to the depot location are taken and the DP algorithm is run to determine the ordering of these items

(see Figure 4.1a) . Then, the beginning and ending parts of the incomplete tour, called shortly as “BP” and “EP”, respectively, each with $\left(\frac{K-C}{2}\right)$ items $((K - C)$ items in total), are fixed and the remaining C items that are in the middle of the incomplete tour are detached (see Figure 4.1b). A set of items including the detached C items and $(K - C)$ new items that are closest to the depot location is formed (Figure 4.1c). A version of the EMFRP is solved by the DP algorithm to optimally insert these K items into the middle of the incomplete tour to make it grow. Note that when solving the DP algorithm, the starting point of the forklift is taken as the location of the last item in BP and the ending point of the forklift is taken as the location of the first item in EP. Moreover, the forklift starts its tour with all the items that are in BP. According to the solution of the DP algorithm, BP and EP are extended with additional $\left(\frac{K-C}{2}\right)$ items each and C items are detached from the middle of the new incomplete tour (see Figure 4.1d). The steps of the algorithm continue in the same manner with deleting C items from the middle, using the DP algorithm considering a subset of K items, and enlarging the incomplete tour until no item remains unvisited. Figures 4.1e and 4.1f show how the final subset of the items is composed and how the complete tour is constructed after solving the last DP, respectively. Note that, the last DP may be run with a subset including less than K items.

In the remaining part of the thesis, we abbreviate this algorithm as “DP-Const”. Note that as the optimization routine, one can also use the proposed MIP formulation. After some experiments, we decided to use the DP algorithm due to its superior performance (see Section 4.5.2 for the details).

4.3.2 Tour Improvement

In this section, we describe some improvement algorithms which take as input a given solution (tour) and aim to improve it.

4.3.2.1 TSP-based Improvement Algorithms

As the EMFRP has some characteristics of the TSP, we adapt two widely-used improvement algorithms from the TSP literature, i.e., 2-opt and 3-opt algorithms (see

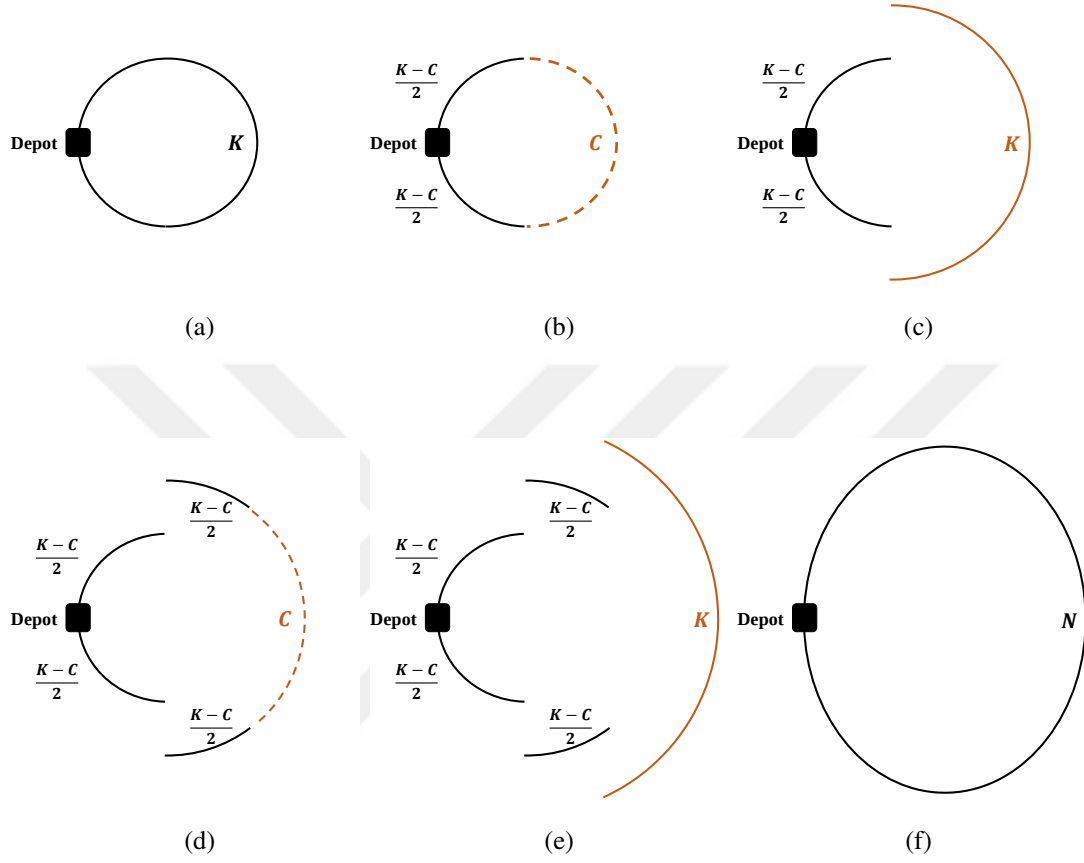


Figure 4.1: The steps of the DP-Construction algorithm, (a) in the 1st step, K closest items to the depot location are taken (b) after the DP algorithm is run, C items are detached from the middle of the incomplete tour, (c) in the second step, detached C items together with new $K - C$ items form the new subset, (d) after the DP algorithm is run, C items are detached from the middle of the incomplete tour, (e) in the third step, detached C items together with new $K - C$ items form the new subset, (f) tour construction is completed.



Figure 4.2: The initial tour (a) and a neighboring tour (b) obtained by removing the dashed arcs from (a) in the 2-opt algorithm

Nilsson (2003) for more detail). In the 2-opt heuristic, two non-adjacent arcs are deleted from a given initial tour. Then two new arcs are added to generate a neighboring tour. The objective function value, i.e., the energy consumption of the forklift, of this new tour is evaluated. Among the initial tour and all neighboring tours, the best one in terms of the objective function value is taken. If the new tour is different than the starting tour, then it is taken as the starting tour and the steps are repeated. The algorithm stops when no better neighboring tour is found. Figure 4.2 presents how a neighboring tour (see Figure 4.2b) is obtained from a given tour (see Figure 4.2a) using the 2-opt algorithm.

In the 3-opt algorithm, the neighboring tours are obtained by removing three arcs from the initial tour and reconnecting the tour by adding three arcs. The neighboring tour resulting in the least energy consumption is determined among all possible ones and if this is an improving tour, iterations are repeated. Figure 4.3 depicts how neighboring tours (see Figures 4.3b-4.3h) are obtained from a given tour (see Figure 4.3a) using the 3-opt algorithm.

Given an initial tour, one neighboring tour is obtained by deleting two non-adjacent arcs in the 2-opt algorithm (see Figure 4.2). On the other hand, in the 3-opt algorithm, seven neighboring tours are obtained when three non-adjacent arcs are deleted (see Figure 4.3) and three neighboring tours are obtained when two adjacent arcs together with one arc that is not adjacent to both are deleted. Given an initial tour, the set of all neighboring tours obtained by 2-opt operations is a subset of the set of all neighboring tours obtained by 3-opt operations. Therefore, the 3-opt algorithm is likely to be computationally more expensive, but is expected to result in better solutions. Note

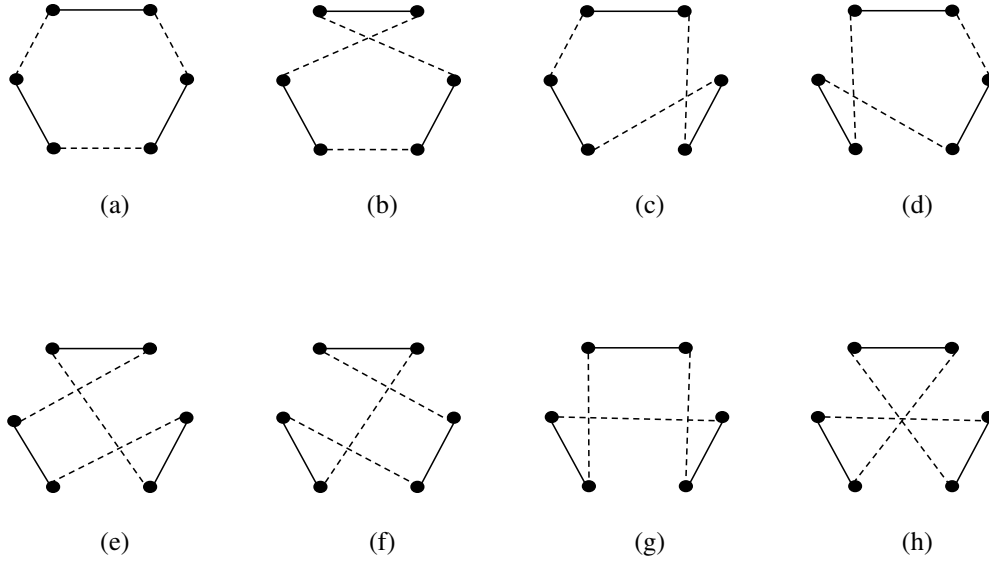


Figure 4.3: The initial tour (a) and the neighboring tours (b)-(h) obtained by removing the dashed arcs from (a) in the 3-opt algorithm

that same neighboring tour may be obtained by deleting different sets of arcs in the 3-opt algorithm. Therefore, we pay special attention in our implementation of the 3-opt algorithm in order to avoid generating the same neighboring solution more than once.

Note that, the descriptions we give for the 2-opt and 3-opt algorithms are for the classical TSP where a tour and its reverse tour have the same objective function value. For the EMFRP, a tour and its reverse tour may have different objective function values. Therefore, when applying the 2-opt and 3-opt algorithms in the context of the EMFRP, we evaluate the objective function value of the reverse tour as well for every neighboring tour as is done in the literature before (see e.g., Tachibana & Adachi (2014)).

After some preliminary experiments with the 2-opt and 3-opt algorithms, the results of which will be presented in Section 4.5.1, we came to the conclusion to employ a hybrid application of the 2-opt and 3-opt algorithms, where the latter is applied on the solution of the former. The hybrid application of the 2-opt and 3-opt algorithms is called as the “2&3-opt” algorithm in our solution approach.

4.3.2.2 DP-Improvement Matheuristic

In this section, we discuss a matheuristic developed for the EMFRP as an alternative improvement algorithm that is called the DP-Improvement algorithm. This algorithm is based on optimizing smaller parts of a given solution (tour) by solving a version of the EMFRP using the DP algorithm. Given an initial tour, the DP-Improvement algorithm selects a subset of K' consecutively visited items and determines the optimal ordering of these items by solving the DP. Note that in this process, only the orderings of the items that are in the selected subset are subject to change. The orderings of all other items are kept the same.

We implement the DP-Improvement algorithm as follows (see Figure 4.4). Given an initial tour (see Figure 4.4a), the first subset is selected to be consisting of the first K' items in the tour (see the dashed lines in Figure 4.4b). After finding the optimal ordering of the items in this subset keeping the ordering of the other items the same (see Figure 4.4c), we update the tour and move on to select the next subset of the items. This selection is done by shifting (or rotating) the previously selected subset by $K' - C'$ items. In other words, in the current tour, the K' items that come after the first $K' - C'$ items form the second subset (see Figure 4.4d). After the orderings of the items in this new subset are optimally determined (see Figure 4.4e), the current tour is updated and the third subset is formed by the K' items that come after the first $2(K' - C')$ items in the current tour (see Figure 4.4f). Figure 4.4g shows how the fifth subset is selected and Figure 4.4h displays the step of the algorithm where the last K' items form the subset. Note that in this step, to keep the size of the subset equal to K' , we may shift (or rotate) the previously selected subset by less than $K' - C'$ items. After this step, we obtain the new tour in Figure 4.4i (this is the end of the first iteration). If this new tour improves upon the initial tour given in Figure 4.4a, the iterations of the DP-Improvement algorithm continue. Otherwise, the algorithm stops. In the remaining part of the thesis, we abbreviate this algorithm as “DP-Imp”.

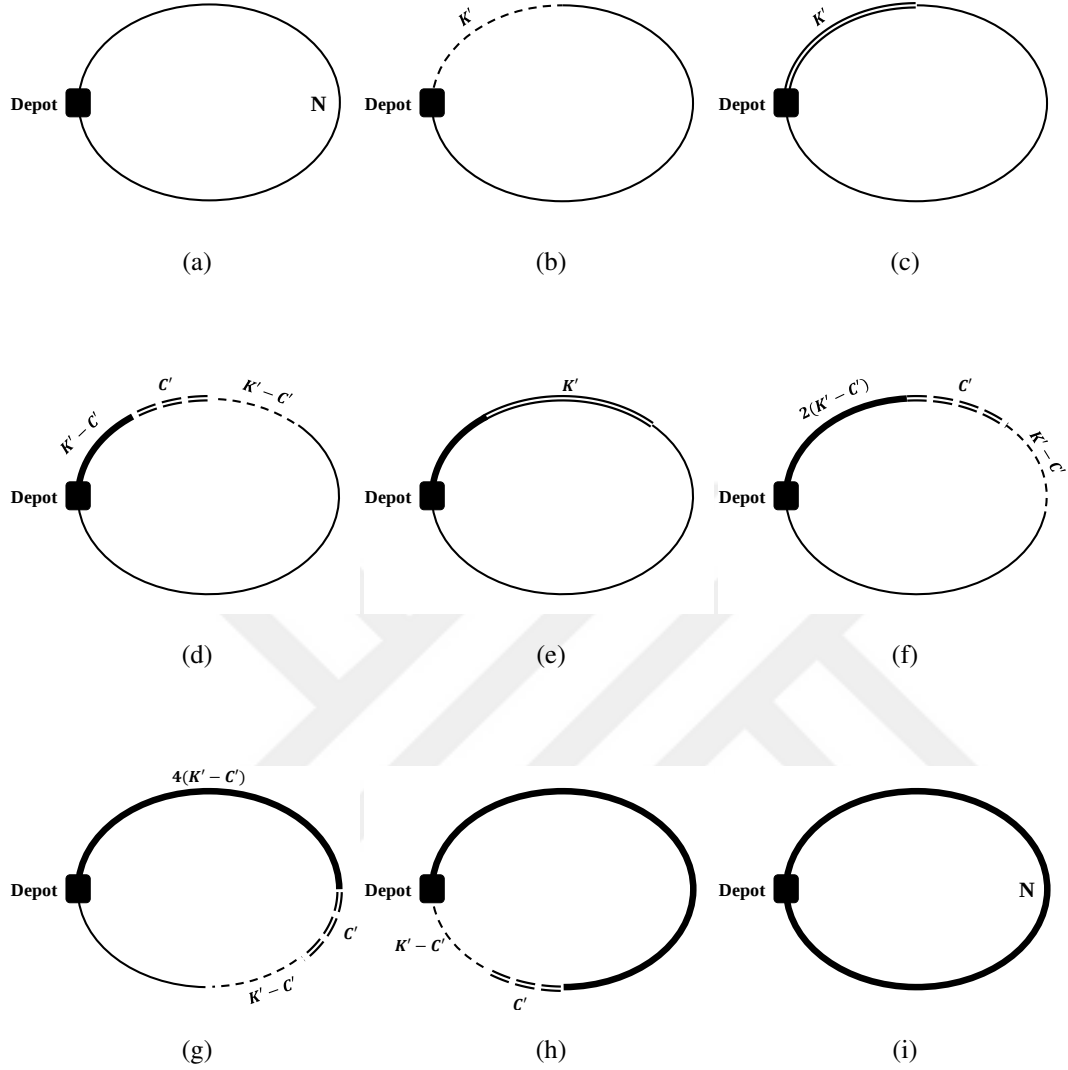


Figure 4.4: The steps of the DP-Improvement algorithm, (a) a given initial tour, (b) first subset is formed by selecting the first K' picked items, (c) the DP is run to optimally reinsert the selected items, (d) the selected subset is rotated by $K' - C'$ items to form the second subset, (e) the DP is run to optimally reinsert the selected items in the second subset, (f) the selected subset is rotated by $K' - C'$ items to form the third subset, (g) fifth subset is formed, (h) last subset is formed, (i) the solution obtained at the end of the first iteration

4.4 Illustrative Example

In this section, we present an illustrate example to compare the solutions of the EMFRP and the distance minimization problem in terms of the energy consumption of the order picker forklift. In Figure 4.5, the black cells show the locations of the items to be picked and the cell that is in the southwest corner of the warehouse shows the location of the depot where the order picker forklift starts and finishes its tour. In Figure 4.5a, two values are written in each black cell; the first one referring to the mass of the item in kg and the second one to the height of the location of the item in m . The distance between neighboring cells is assumed to be $1m$. To go from the end point of an aisle to a neighboring aisle, the forklift travels a distance of $5m$. Other parameters are as follows: sum of masses of the fork and operator (L_f): $150kg$, sum of masses of the forklift and operator (L_t): $2500kg$, forklift's horizontal speed (v_x): $2.0m/s$, fork's vertical (lifting/lowering) speed (v_y): $0.3m/s$, horizontal deceleration distance of the forklift (d_d): $0.5m$, gravitational acceleration (g): $9.81m/s^2$, horizontal kinetic friction coefficient (μ): 0.05 , and vertical constant friction force (F_r): $30N$.

We first solve the order picking problem by minimizing the total horizontal distance traveled by the forklift and display the resulting solution in Figure 4.5b, where the number in each cell refers to the order the corresponding item is picked. This solution is obtained in less than one second when we use the solver CPLEX and an MIP formulation obtained from (EMFRP-MIP) by changing the objective function to the sum of the horizontal distances traveled and removing the constraints (4.15) and (4.16). For this tour, the energy consumption of the order picker forklift is computed as 633,842 joule. We then solve the order picking problem by minimizing the total horizontal and vertical distances traveled by the forklift and display the resulting solution in Figure 4.5c. This solution is also obtained in one second and the energy consumption of the order picker forklift is computed as 588,879 joule for this tour. Finally, we solve the EMFRP and obtain the solution displayed in Figure 4.5d. In this case, it takes the same solver about 5 minutes to solve the problem and the energy consumption of the forklift turns out to be 571,818 joule. By the solution of the EMFRP, we obtain a %10 saving in energy consumption compared to the solution of the total horizontal distance minimization problem and a %3 saving compared to the

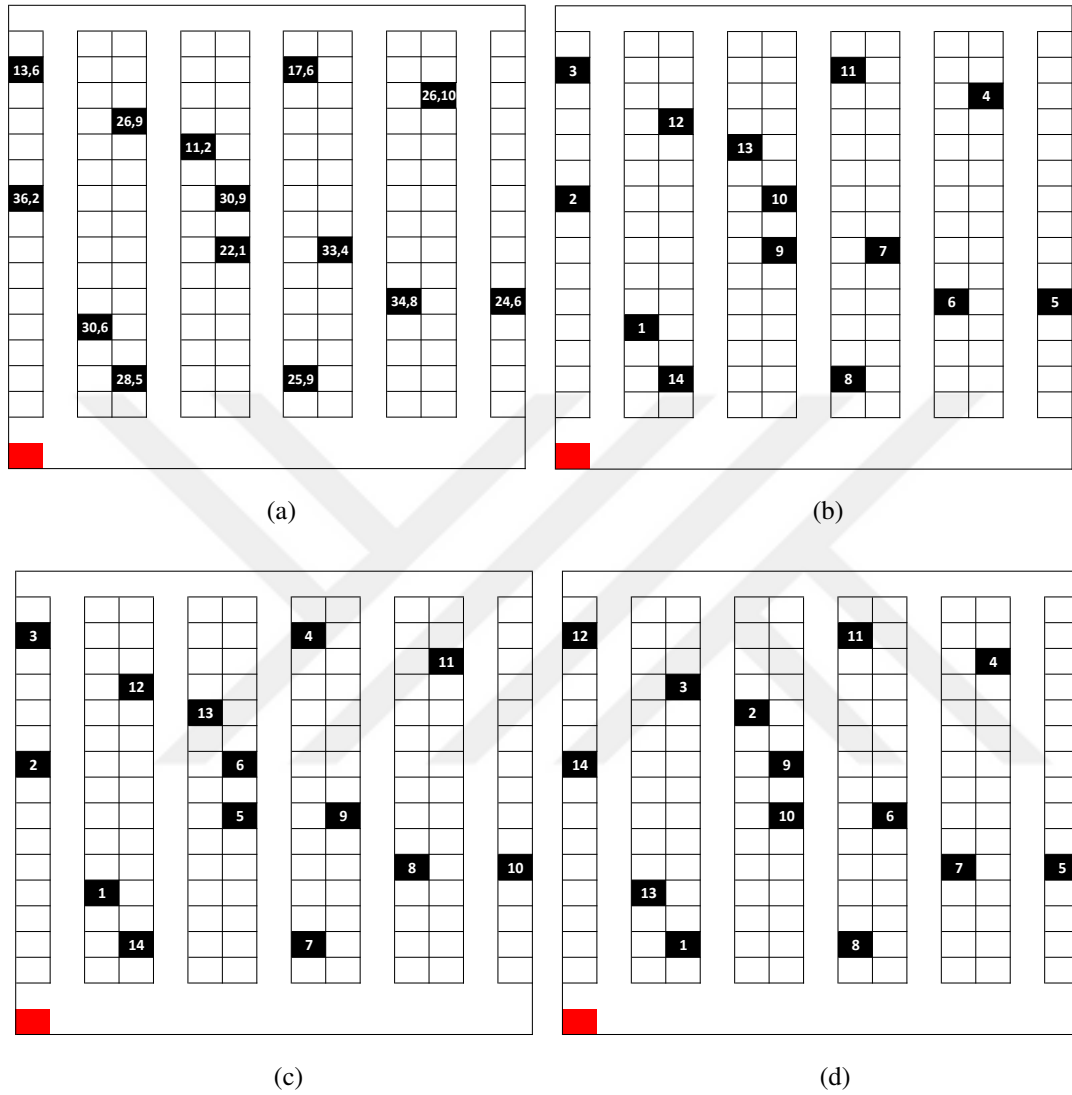


Figure 4.5: (a) an instance with 14 items, (b) solution of the horizontal distance minimization problem, (c) solution of the horizontal + vertical distance minimization problem, (d) solution of the EMFRP

solution of the total horizontal and vertical distances minimization problem.

4.5 Computational Experiments

We conduct several experiments on the instances generated considering the warehouse configuration provided in Henn & Wäscher (2012). In Henn & Wäscher (2012), a single-block warehouse with 10 two-sided aisles is considered which has the potential horizontal locations of 900 items. We generate the instances for the EMFRP as follows. First of all, similar to what is done in Scholz et al. (2016), we generate one group of instances considering all of the 10 aisles of the warehouse and another group considering only a part of the warehouse consisting of the leftmost 5 aisles. For each item, we randomly generate the horizontal locations of the items, the heights of the items, and the masses of the items. The horizontal locations of the items are generated uniformly at random from the 450 and 900 potential locations for the 5-aisle instances and 10-aisle instances, respectively. The heights of the items are generated considering three settings (low (L), medium (M), and high (H) height difference settings) from the integers in the intervals [1,4], [1,7], and [1,10] uniformly at random for the settings L, M, and H, respectively. The masses of the items are also generated considering three settings (low (L), high (H), and general (G)) from the integers in the intervals [1,20], [21,40], and [1,40] uniformly at random for the settings L, H, and G, respectively. We consider instances with $n = 10, 15, 20, 25, 30, 40, 50, 75, 100$ items. For each n value and for each combination of height-mass settings, 10 instances are generated at random. Therefore, we have in total 9 different combinations of height-mass settings and 90 different instances for each value of n . The other parameters needed for energy calculations are the same as those given in Section 4.4.

CPLEX 12.8 is used to solve the MIP formulations, where we set a time limit of 3 hours. The other solution approaches are coded in C++ where we also set the same time limit. We propose a solution approach that contains both construction and improvement algorithms. First, 11 initial solutions are generated, 9 of which at random, two of which by applying the two construction algorithms, i.e., the nearest neighbor and DP-Const algorithms. We then apply the two improvement algorithms; namely, 2&3-opt algorithm and the DP-Imp algorithm, to improve the initial solutions. Note

that the 2&3-opt algorithm is preferred over the 2-opt and 3-opt algorithms as a result of some preliminary experiments. The implementation sequence of the improvement algorithms and the parameters of DP-Const and DP-Imp algorithms are also determined after some preliminary experiments which are discussed in the next section.

4.5.1 Preliminary Experiments

The preliminary experiments are divided into four parts. In the first part, we aim to choose among the 2-opt, 3-opt, and 2&3-opt heuristics. In the second part, the parameters of the DP-Const algorithm are selected. In the third part, we decide on the sequence of the two improvement heuristics considered. Finally, in the fourth part, the parameters of the DP-Imp algorithm are selected.

4.5.1.1 Choosing among 2-opt, 3-opt, or 2&3-opt

We first provide, in Tables 4.1 and 4.2, the results of the preliminary experiments to compare the performances of the 2-opt, 3-opt, and 2&3-opt heuristics. The 2&3-opt heuristic is a hybrid approach where the 3-opt algorithm is applied after the 2-opt algorithm. In these experiments, for each n value and for each height-mass setting, there are 10 instances for each of which 11 initial solutions are generated (9 randomly, 1 by the nearest neighbor algorithm, and 1 by the DP-Const algorithm). For each instance and each initial solution, the three improvement heuristics considered here find a solution. For each instance, each improvement algorithm has a best found solution which is the best of the 11 solutions found. For an instance, the overall best solution, on the other hand, refers to the best found solution by all of these three improvement heuristics. In Tables 4.1 and 4.2, the average solution time (Av. Time), the average percent improvement (Av. %Imp.), the average percent deviation from the overall best solution (Av. %Dev.), and the average deviation of the best solution of the corresponding approach from the overall best (B. %Dev.) are given. Av. Time refers to the average of the 990 solution times. Av. %Imp. refers to the average of the 990 percent improvement values that the improvement stage of the heuristic has provided over the given initial solutions. Av. %Dev. gives the average of the 990

Table 4.1: Performances of the 2-opt, 2&3-opt, and 3-opt algorithms on 5-aisle instances with different n values

n	2-opt			2&3-opt			3-opt		
	Av.	Av.	Av. %Dev.	Av.	Av.	Av. %Dev.	Av.	Av.	Av. %Dev.
	Time	%Imp.	(B.%Dev.)	Time	%Imp.	(B.%Dev.)	Time	%Imp.	(B.%Dev.)
10	0.0	28.7	0.7 (0.0)	0.0	29.1	0.1 (0.0)	0.0	29.2	0.1 (0.0)
20	0.4	37.7	2.3 (0.1)	0.4	38.7	0.6 (0.0)	0.4	38.7	0.6 (0.0)
30	1.1	42.7	3.8 (0.7)	1.0	44.1	1.1 (0.0)	1.4	44.2	1.1 (0.0)
50	2.4	49.5	6.4 (2.3)	3.8	51.4	2.0 (0.2)	9.0	51.5	1.9 (0.2)
75	4.7	54.2	9.0 (3.4)	22.7	56.7	2.1 (0.3)	64.8	56.8	2.0 (0.3)
100	8.1	57.4	11.2 (4.0)	99.6	60.5	2.0 (0.2)	281.5	60.5	2.0 (0.3)
Av.	2.8	45.0	5.6 (1.8)	21.3	46.8	1.3 (0.1)	59.5	46.8	1.3 (0.1)

percent deviations of the objective function values of the found solutions from that of the overall best solutions, while B. %Dev. shows the average of the 90 percent deviations of the best objective function values of the instances from the overall best solutions. The last rows in these tables give the average values for the corresponding column.

Tables 4.1 and 4.2 show that in general, 3-opt heuristic provide the best tours in terms of the energy consumption, while the 2-opt heuristic gives the worst ones. In terms of solution time, the best heuristic is 2-opt while the 2&3-opt heuristic turns out to be faster than the 3-opt heuristic. Since the solution quality of the 2&3-opt heuristic is not significantly different from that of the 3-opt heuristic, and the solution time of the 2&3-opt heuristic is significantly better than that of the 3-opt heuristic, we select 2&3-opt heuristic as an improvement algorithm to be used in the proposed solution approach.

4.5.1.2 Selection of the parameters of the DP-Const algorithm

Here, we report a subset of our preliminary experiments performed to determine the parameters of the proposed tour construction algorithm, the DP-Const algorithm. We

Table 4.2: Performances of the 2-opt, 2&3-opt, and 3-opt algorithms on 10-aisle instances with different n values

n	2-opt			2&3-opt			3-opt		
	Av.	Av.	Av. %Dev.	Av.	Av.	Av. %Dev.	Av.	Av.	Av. %Dev.
	Time	%Imp.	(B.%Dev.)	Time	%Imp.	(B.%Dev.)	Time	%Imp.	(B.%Dev.)
10	0.0	26.5	0.8 (0.0)	0.0	27.0	0.1 (0.0)	0.0	27.1	0.1 (0.0)
20	0.4	35.9	2.0 (0.1)	0.4	36.7	0.6 (0.0)	0.5	36.8	0.5 (0.0)
30	1.1	40.8	3.2 (0.6)	1.0	42.0	1.0 (0.0)	1.4	42.1	0.8 (0.0)
50	2.5	46.7	5.6 (2.1)	4.0	48.6	1.6 (0.1)	9.3	48.6	1.5 (0.2)
75	4.9	51.7	8.1 (3.2)	23.4	54.2	1.8 (0.2)	65.2	54.3	1.8 (0.3)
100	8.2	54.7	10.6 (4.0)	100.9	57.8	2.0 (0.3)	282.8	57.9	1.9 (0.2)
Av.	2.8	42.7	5.1 (1.6)	21.6	44.4	1.2 (0.1)	59.9	44.5	1.1 (0.1)

denote by K the size of the subproblems and by C the number of items detached from the incomplete tours. We take the parameters of the DP-Const algorithm as K and S where S is equal to $K - C$. S represents the number of additional items whose orderings are fixed in each step of the algorithm. The performance of the algorithm in terms of the solution time and quality depends on these parameters and our aim is to choose the parameter setting that offers the best performance. Note that the solution time of the DP algorithm increases exponentially with the input size. Moreover, the alternative construction algorithms, i.e., the nearest neighbor algorithm and random initial tour, are quite fast. In order to be able to compete with them, we do not consider values larger than 16 for the parameter K as larger values of K makes the DP-Const algorithm relatively much slower. In our experiments, we take K as 16 and S as 4 or 6 and compare the performance of the DP-Const algorithm with the nearest neighbor algorithm. These experiments are performed taking the warehouse with 5 aisles and considering height-mass settings M-G, H-L, H-H, and H-G, a subset of instance sizes, and generating 7 instances for each setting.

In Table 4.3, we summarize the results of our experiments with parameter settings $K - S : 16 - 4$ and $K - S : 16 - 6$. In this table, the first column gives the number of items n in the instances. For each of the parameter settings, the column titled as “DP-

Table 4.3: Comparison of quality of solutions generated by DP-Const algorithm with $S = 4$ and $S = 6$ with that of the nearest neighbor algorithm

n	$K - S : 16 - 4$		$K - S : 16 - 6$	
	Av. DP-Const	Av. %Dev. of	Av. DP-Const	Av. %Dev. of
	Time	Const. Sol.s	Time	Const. Sol.s
20	0.5	-9.2	0.4	-8.7
25	0.8	-9.1	0.6	-8.1
30	1.2	-7.8	0.8	-7.7
40	1.9	-2.8	1.3	-2.3
50	2.5	2.7	1.7	3.3
75	4.2	8.9	2.9	9.9
100	5.8	18.4	3.9	19.2
Av.	2.4	0.2	1.7	0.8

Const Time” gives the average time spent for the initial solution generation by the DP-Const algorithm for each instance in seconds. The columns titled as “Av. %Dev. of Const. Sol.s” present the average percent deviation of the objective function values of the solutions constructed by the DP-Const algorithm from those constructed by the nearest neighbor algorithm. The last row in each table gives the average values for the corresponding column.

From Table 4.3, we see that the average time spent for the initial solution generation by the DP-Const algorithm increases with the decrease of the parameter S , as expected. A decrease in S leads to an increase in the number of optimization problems, i.e., number of steps of the algorithm, solved during the DP-Const algorithm. Considering the columns corresponding to percent deviations, we can observe that the objective function values of the initial solutions provided by the DP-Const algorithm improve by decrease in S . Consequently, we decide to choose S as 4 to get better solutions while spending little more time. Henceforth, in our final experiments, we applied the DP-Const algorithm using the parameter setting $K - S : 16 - 4$. Note that the DP-Const algorithm generates better initial solutions than the nearest neighbor

bor algorithm when the sizes of the instances are small ($n \leq 40$) and the nearest neighbor algorithm performs better for larger size instances. Therefore, none of these construction algorithms outperforms the other in general.

4.5.1.3 Determining the implementation sequence of the two improvement heuristics

Once a tour construction algorithm constructs an initial tour, we use improvement heuristics to improve this solution. For the EMFRP, we have in our hand two improvement heuristics; namely, the 2&3-opt algorithm and the DP-Imp algorithm. In our implementation, we apply one of these improvement heuristics the first and the other one the second in order to try to further improve the solution obtained by the first improvement heuristic to get good quality solutions. Now we discuss our preliminary experiments performed to determine which of the improvement heuristics to apply the first and which one to apply the second.

The alternative in which the 2&3-opt algorithm is applied the first is called the “first” sequence alternative and the alternative in which the DP-Imp algorithm is applied the first is called the “second” sequence alternative. The experiments are done taking the warehouse with 5 aisles and considering height-mass settings M-G, H-L, H-H, and H-G, a subset of instance sizes, and generating 7 instances for each setting. Moreover, for each instance nine initial solutions are generated at random which are then improved separately by the two sequence alternatives. We take different parameter settings of the DP-Imp algorithm into account to be able to make a better comparison.

In Table 4.4, for each n value, we present the average percent deviations of the objective function values obtained by the first sequence alternative from those obtained by the second alternative. The column titles refer to the parameter settings of the DP-Imp algorithm. For each instance, each sequence alternative provides nine solutions one of which gives a best objective function value. Inside the parentheses, we report the average percent deviations of the best objective function values obtained by the first sequence alternative from those obtained by the second alternative. Negative values in the table mean that the first alternative is better on average. Therefore, we can con-

Table 4.4: Average percent deviations of the objective function values of the solutions obtained by the first sequence alternative from those of the second one with different DP-Imp parameter settings and n values

n	Av. %Dev. (Best Sol. %Dev.)			
	16-4	16-6	18-4	18-6
20	-0.1 (0.0)	-0.2 (0.0)	-0.1 (0.0)	-0.1 (0.0)
25	0.0 (-0.1)	-0.1 (0.0)	-0.1 (0.0)	-0.1 (0.0)
30	-0.1 (0.0)	-0.1 (0.0)	-0.1 (0.0)	-0.1 (0.0)
40	-0.2 (0.0)	-0.2 (-0.1)	-0.1 (-0.1)	-0.3 (-0.1)
50	-0.1 (-0.1)	0.0 (-0.1)	0.0 (-0.1)	-0.1 (-0.1)
75	0.0 (0.0)	-0.2 (-0.1)	-0.1 (0.0)	0.1 (0.2)
100	-0.1 (-0.3)	0.0 (-0.3)	-0.2 (-0.4)	-0.1 (-0.2)
Av.	-0.1 (-0.1)	-0.1 (-0.1)	-0.1 (-0.1)	-0.1 (0.0)

clude that the first sequence alternative, i.e., applying the 2&3-opt algorithm the first and the DP-Imp algorithm the second, results in better solutions for all the parameter settings considered for the DP-Imp algorithm.

4.5.1.4 Selection of the parameters of the DP-Imp algorithm

The last set of preliminary experiments in our study is performed to determine the parameter setting of the proposed improvement algorithm, that is, the DP-Imp algorithm. Similar to the DP-Const algorithm, there are two parameters in the DP-Imp algorithm: K' which refers to the number of items in each subset on which the DP is run and $S' = K' - C'$ which refers to the number of items that are added to the fixed part in each step. To determine the parameters of the DP-Imp algorithm, the instances used in Section 4.5.1.3 are employed. For each instance, 9 random initial solutions are generated and are improved first by the 2&3-opt algorithm. Then, the proposed DP-Imp algorithm with four different parameter settings is applied on every 9 solutions obtained by the 2&3-opt algorithm for each instance.

Table 4.5: Comparisons of the solutions obtained by the 2&3-opt algorithm with those that are obtained after further improvement by the DP-Imp algorithm with $K' - S'$: 16 – 4 parameter setting

n	Av. 2&3-opt Time	Av. DP-Imp Time	Av. 2&3-opt %Imp.	Av. DP-Imp %Imp.	Av. % of Imp. Sol.s	Av. B. %Dev.
20	0.0	1.1	46.3	0.3	29.2	0.0
25	0.0	1.5	50.6	0.3	30.9	-0.1
30	0.1	1.8	53.1	0.3	31.3	0.0
40	0.6	3.2	56.7	0.3	40.6	-0.1
50	1.8	3.7	60.5	0.2	29.2	-0.1
75	16.3	6.4	66.1	0.1	33.7	-0.1
100	79.6	8.3	70.5	0.1	27.1	-0.1
Av.	14.1	3.7	57.7	0.22	31.7	-0.08

In Tables 4.5-4.8, we summarize the results. The values in these tables come from the average of values obtained for instances with number of items given in the first column. In each table, the second column (Av. 2&3-opt Time) shows the average time, in seconds, spent by the 2&3-opt algorithm while the third column (Av. DP-Imp Time) gives the average time spent by the DP-Imp algorithm. The column titled as “Av. 2&3-opt %Imp.” displays the average of percent improvement gained by the 2&3-opt algorithm on the random initial solutions and the column “Av. DP-Imp %Imp.” presents the average of percent improvement gained by the DP-Imp algorithm over the solutions of the 2&3-opt algorithm. The penultimate column, titled as “Av. % of Imp. Sol.s” provides the average of the percentage of the solutions of the 2&3-opt algorithm that are further improved by the DP-Imp algorithm. Lastly, the column “Av. B. %Dev.” gives the average of percent deviation of the best solution of each instance obtained by the DP-Imp algorithm from that obtained by the 2&3-opt algorithm. The average values of the columns are given in the last rows of the tables.

The second column in Table 4.5 refers to the average time spent by the 2&3-opt algorithm and is copied to Tables 4.6-4.8 to make the comparison easier with the solution

Table 4.6: Comparisons of the solutions obtained by the 2&3-opt algorithm with those that are obtained after further improvement by the DP-Imp algorithm with $K' - S'$: 16 – 6 parameter setting

n	Av. 2&3-opt Time	Av. DP-Imp Time	Av. 2&3-opt %Imp.	Av. DP-Imp %Imp.	Av. % of Imp. Sol.s	Av. B. %Dev.
20	0.0	0.7	46.3	0.3	28.5	0.0
25	0.0	1.1	50.6	0.3	30.6	-0.1
30	0.1	1.4	53.1	0.3	28.8	0.0
40	0.6	2.3	56.7	0.2	38.9	-0.1
50	1.8	2.4	60.5	0.2	26.4	-0.1
75	16.3	4.4	66.1	0.1	30.9	-0.1
100	79.6	5.6	70.5	0.1	25.0	-0.1
Av.	14.1	2.6	57.7	0.21	29.9	-0.08

Table 4.7: Comparisons of the solutions obtained by the 2&3-opt algorithm with those that are obtained after further improvement by the DP-Imp algorithm with $K' - S'$: 18 – 4 parameter setting

n	Av. 2&3-opt Time	Av. DP-Imp Time	Av. 2&3-opt %Imp.	Av. DP-Imp %Imp.	Av. % of Imp. Sol.s	Av. B. %Dev.
20	0.0	4.0	46.3	0.4	31.9	0.0
25	0.0	8.8	50.6	0.5	40.3	-0.1
30	0.1	10.8	53.1	0.4	39.9	0.0
40	0.6	16.4	56.7	0.4	49.0	-0.1
50	1.8	21.9	60.5	0.3	39.2	-0.1
75	16.3	35.3	66.1	0.2	39.9	-0.1
100	79.6	46.8	70.5	0.1	33.7	-0.1
Av.	14.1	20.6	57.7	0.31	39.1	-0.08

Table 4.8: Comparisons of the solutions obtained by the 2&3-opt algorithm with those that are obtained after further improvement by the DP-Imp algorithm with $K' - S'$: 18 – 6 parameter setting

n	Av. 2&3-opt Time	Av. DP-Imp Time	Av. 2&3-opt %Imp.	Av. DP-Imp %Imp.	Av. % of Imp. Sol.s	Av. B. %Dev.
20	0.0	3.8	46.3	0.3	31.6	0.0
25	0.0	6.5	50.6	0.4	37.8	-0.1
30	0.1	8.5	53.1	0.4	38.2	0.0
40	0.6	11.5	56.7	0.3	46.2	-0.1
50	1.8	15.1	60.5	0.3	37.5	-0.1
75	16.3	23.2	66.1	0.1	35.8	-0.1
100	79.6	30.5	70.5	0.1	30.6	-0.1
Av.	14.1	14.2	57.7	0.28	36.8	-0.08

times of the DP-Imp algorithm for different parameter settings. When comparing the third columns of the tables with each other, we observe an increase in the computational time of the DP-Imp algorithm as K' increases and S' decreases. This is expected because the total time spent in solving all the subproblems increases with K' and the number of subproblems which increases as S' decreases. The average percent improvement values obtained with the 2&3-opt algorithm over the random initial solutions, which are given in the fourth columns of the tables, are quite high. In practice, improving the solutions obtained by the 2&3-opt algorithm can be challenging and any small improvement would be desirable as it would reduce the energy consumption of the forklift even further. When comparing the 5th and 6th columns of the tables, one can observe that the values in these columns are the largest when $K' = 18$. On the other hand, looking at the last columns of the tables, one can see that the parameters of the DP-Imp algorithm used in our experiments all have the same effect on improving the best solutions obtained by the 2&3-opt algorithm. Considering all discussed above, we choose $K' - S'$ as 18 – 6, instead of 18 – 4 due to the time advantage of the former, though the latter setting could also be selected if one prefers solution quality over time. As a result, we apply the DP-Imp algorithm after

the 2&3-opt algorithm with parameters $K' - S' : 18 - 6$ in the following experiments.

So far, we have discussed the results of our preliminary computational experiments. Through these experiments, we have selected the parameters of our algorithms and have turned all tour construction and tour improvement algorithms into a single solution approach for the EMFRP. In this solution approach, for each instance we generate 9 random initial solutions, 1 initial solution by the nearest neighbor algorithm, and one initial solution with the DP-Const algorithm. These 11 initial solutions are improved first by the 2&3-opt algorithm. The resulting solutions are then improved by the DP-Imp algorithm. Finally, we choose the best solution among these 11 solutions for each instance. In the next section, we provide the results of our main experiments comparing the proposed solution approach with the exact solution methods and comparing the EMFRP with the (horizontal+vertical) distance minimization problem in terms of energy consumption and tour time of the order picker forklift.

4.5.2 Further Computational Experiments

In this section, we evaluate the proposed solution approach by comparing its performance with the exact solution approaches. As a result of this comparison, we provide a suggestion to the user to decide which solution method to use (the MIP formulation, the DP, or the proposed approach) for different instance sizes. Moreover, we evaluate the savings in energy consumption by employing the EMFRP instead of the (horizontal+vertical) distance minimization (forklift routing) problem (HVDMPFRP). We also observe that in some height-mass settings, the value of solving EMFRP instead of the distance minimization problem is much larger than in some other settings. Finally, we will see that the solutions provided by the EMFRP also reduce the tour time of the order picker forklift when compared with the solutions of the HVDMPFRP.

We first provide the results of the experiments on solving the EMFRP by the MIP formulation using the CPLEX solver. Table 4.9 provides the results for the 5-aisle instances while Table 4.10 for the 10-aisle instances. In both tables, the columns correspond to different n values and the rows correspond to height-mass settings. The values in these tables give the average of solution times (in seconds) of 10 instances for each size and setting, and the average (over 10 instances) relative MIP gap re-

Table 4.9: Average solution times and relative MIP gaps when the 5-aisle instances with different number of items (n) and height-mass settings are solved using the MIP formulation

height-mass setting	Average solution time in s (Gap%)				
	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$
L-L	0.3	6.3	1673.1	10133.4 (4.1)	10800 (9.2)
L-H	0.6	349.7	10800 (11.5)	10800 (18.4)	10800 (23.7)
L-G	0.4	56.3	9061.0 (4.5)	10800 (11.8)	10800 (16.5)
M-L	0.3	12.1	2827.3 (0.4)	10800 (5.8)	10800 (11.6)
M-H	1.1	3239.8	10800 (14.0)	10800 (21.1)	10800 (27.0)
M-G	0.4	274.1	10791.3 (6.7)	10800 (14.5)	10800 (18.6)
H-L	0.3	11.9	4397.1 (1.1)	10800 (7.8)	10800 (12.3)
H-H	1.6	2141.7	10800 (15.3)	10800 (22.5)	10800 (27.9)
H-G	0.5	104.4	10800 (8.5)	10800 (15.2)	10800 (18.9)

turned by CPLEX after the 3 hour time limit in parentheses. Note that non-existence of the parentheses implies that the corresponding value is zero, i.e., all the instances are solved to optimality within 3 hours.

Tables 4.9 and 4.10 show that none of the instances with $n = 30$ items can be solved to optimality by CPLEX in 3 hours. Moreover, among both 5-aisle and 10-aisle instances with $n = 25$ items, only 1 instance having the setting L-L is solved to optimality within 3 hours. For the 5-aisle instances with $n = 20$ items, 10, 2, 8, 1, and 8 instances are solved to optimality within 3 hours for the settings L-L, L-G, M-L, M-G, and H-L, respectively. For the 10-aisle instances with $n = 20$ items, on the other hand, 9, 3, 8, 1, and 7 instances are solved to optimality within 3 hours for the settings L-L, L-G, M-L, M-G, and H-L, respectively. In general, the instances with setting H for masses, i.e., high masses, take more time to solve. Besides, the increase in the height differences of the items makes the problem slightly harder to solve.

In Table 4.11, the performance of the DP algorithm is presented and compared with that of the MIP formulation. Here, for the instances with different number of items,

Table 4.10: Average solution times and relative MIP gaps when the 10-aisle instances with different number of items (n) and height-mass settings are solved using the MIP formulation

height-mass setting	Average solution time in s (Gap%)				
	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$
L-L	0.4	6.4	2711.4 (0.1)	9942.5 (4.4)	10800 (9.0)
L-H	1.0	344.3	10800 (10.3)	10800 (17.5)	10800 (22.7)
L-G	0.6	49.8	8351.7 (3.4)	10800 (11.4)	10800 (15.8)
M-L	0.3	9.8	2785.5 (0.6)	10800 (6.0)	10800 (11.3)
M-H	0.9	1555.8	10800 (13.0)	10800 (20.0)	10800 (26.2)
M-G	0.6	109.0	9839.0 (5.6)	10800 (13.6)	10800 (19.1)
H-L	0.4	12.7	5954.4 (0.9)	10800 (6.7)	10800 (12.4)
H-H	1.6	2655.7	10800 (14.9)	10800 (20.7)	10800 (27.1)
H-G	0.8	169.9	10800 (7.4)	10800 (14.2)	10800 (19.4)

the average (over 90 instances) solution time of the DP (in seconds) is given in the column titled “Time”. For each instance, the column titled as “%Dev. of Time” gives the average of the percent deviations of the computational times of the DP from that of the MIP formulation (CPLEX). The columns titled as “%Dev. of OFV” show the average percent deviation of the objective function values of the DP solutions from the best solutions of the MIP formulation returned by CPLEX in 3 hours.

According to Table 4.11, all the 5-aisle and 10-aisle instances with 10, 15, 20 and 25 items can be solved optimally by the DP algorithm in 3 hours. On the other hand, the DP algorithm cannot solve any instance with $n = 30$ items in 3 hours. Even though the DP solution times increase very quickly as n increases, they are still much better than the solution times of the MIP formulation for all instance sizes. For this time advantage of the DP over the MIP formulation, the DP algorithm is used as the optimization routine within the proposed DP-Const and DP-Imp algorithms.

Among the 61 5-aisle instances with 20 items that cannot be solved to optimality within 3 hours by the MIP formulation, the DP algorithm gives a better solution in

Table 4.11: Solution time of the DP, its percent deviation from the CPLEX time, and the percent deviation of the objective function values of the DP solutions from those of the CPLEX solution all on average for the 5- and 10- aisle instances with different n values

n	5 aisles			10 aisles		
	Time	%Dev. of Time	%Dev. of OFV	Time	%Dev. of Time	%Dev. of OFV
10	0.0	-98.7	0.00	0.0	-98.5	0.00
15	0.3	-97.1	0.00	0.3	-96.7	0.00
20	20.4	-98.5	-0.01	20.4	-97.7	-0.01
25	1175.8	-88.7	-0.12	1167.9	-88.7	-0.13

only 5 instances. This means that for the remaining 56 instances, the best solution returned after 3 hours by CPLEX is indeed an optimal solution. Similarly, among the 89 5-aisle instances with $n = 25$ items that cannot be solved to optimality within 3 hours by the MIP formulation, the DP returns a better solution in 22 instances. Therefore, for the remaining 67 instances, the best integer solution returned by CPLEX after 3 hours is the optimal solution. For the 10-aisle instances with 20 and 25 items, the DP algorithm provides better solutions than the MIP formulation for 3 out of 62 and 30 out of 89 instances that cannot be solved to optimality within 3 hours by the MIP formulation, respectively.

Now, we evaluate the quality of the solutions returned by our proposed solution approach. For this purpose, for each instance, we determine the best solution over 11 solutions provided by the proposed approach. For each such best solution, we compute the percent deviation of its objective function value from that of the MIP formulation and the DP algorithm. Statistics about these percent deviations are provided in Tables 4.12 and 4.13 for 5-aisle and 10-aisle instances, respectively. The values in these tables show the average (over 10 instances) and the maximum (in parentheses) of these percent deviations for different instance sizes and height-mass settings. Note that the proposed solution approach always finds the optimal solution for instances

Table 4.12: Solution quality of the proposed solution approach compared to exact approaches in 5-aisle instances with different height-mass settings and different n .

Setting	Av. %Dev. of OFV from exact solution approach (max %Dev.)				
	$n = 20$		$n = 25$		$n = 30$
	DP	CPLEX	DP	CPLEX	CPLEX
L-L	*	*	*	-0.05 (0.0)	-0.13 (0.0)
L-H	*	-0.03 (0.0)	0.04 (0.4)	-0.10 (0.4)	-0.30 (0.0)
L-G	*	*	*	*	-0.17 (0.4)
M-L	*	*	*	*	-0.09 (0.0)
M-H	*	-0.01 (0.0)	*	-0.25 (0.0)	-0.48 (0.0)
M-G	*	*	0.05 (0.5)	-0.13 (0.5)	-0.02 (0.8)
H-L	*	*	0.11 (0.7)	*	-0.24 (0.0)
H-H	*	-0.01 (0.0)	*	-0.19 (0.0)	-0.30 (0.1)
H-G	*	-0.03 (0.0)	*	-0.15 (0.0)	-0.24 (0.0)

* = 0.00(0.0)

with n less than or equal to 18. This is because a DP with 18 items is solved during the DP-Imp algorithm. For this reason, the results are only provided for instances with n greater than 18. In these tables, negative values show that the solutions found by the proposed solution approach are better than the corresponding exact solution approach on average.

Tables 4.12 and 4.13 show that for all the 5- and 10-aisle instances with 20 items, the proposed solution approach is able to find the optimal solution. For the majority of the instances with 25 items, the proposed approach finds the optimal solution and most of the solutions found are better than those returned by CPLEX in 3 hours. Finally, for most of the instances with 30 items, the proposed solution approach finds better solutions than the MIP formulation, while the DP cannot return a solution in 3 hours. In summary, for small size instances, the solutions found by the proposed solution approach are of high quality. We suggest the user to use the DP approach for instances with up to 20 items and the proposed solution approach for larger instances.

Table 4.13: Solution quality of the proposed solution approach compared to exact approaches in 10-aisle instances with different height-mass settings and different n .

Setting	Av. %Dev. of OFV from exact solution approach (max %Dev.)				
	$n = 20$		$n = 25$		$n = 30$
	DP	CPLEX	DP	CPLEX	CPLEX
L-L	*	*	*	-0.11 (0.0)	-0.15 (0.0)
L-H	*	*	*	-0.09 (0.0)	-0.10 (0.0)
L-G	*	*	0.02 (0.2)	-0.04 (0.2)	-0.25 (0.3)
M-L	*	*	0.01 (0.1)	-0.02 (0.1)	-0.17 (0.2)
M-H	*	*	0.02 (0.1)	-0.28 (0.0)	-0.73 (0.0)
M-G	*	-0.05 (0.0)	*	-0.14 (0.0)	-0.36 (0.0)
H-L	*	*	*	-0.05 (0.0)	-0.43 (0.0)
H-H	*	-0.08 (0.0)	*	-0.13 (0.0)	-0.47 (0.0)
H-G	*	*	*	-0.21 (0.0)	-0.50 (0.0)

* = 0.00(0.0)

For instances with 25 items, the solution time of the DP is about 20 minutes. On the other hand, the proposed approach finds very good quality (mostly optimal) solutions for these instances in less than 1.5 minutes.

In this part of the experimental results, we provide a set of analysis on the construction algorithms that are used within the proposed solution approach. Remember that, 3 different construction algorithms are utilized within the proposed approach to generate 11 initial solutions: 9 at random, 1 by the nearest neighbor algorithm, and 1 by the DP-Const algorithm. Each initial solution constructed is then improved by the improvement algorithms (first by the 2&3-opt algorithm and then by the DP-Imp algorithm) to obtain the final solution. Here, we want to find out how often the best final solution of each instance is obtained when different construction algorithms are used. For this purpose, for each construction algorithm, we report in the first three columns of Tables 4.14 and 4.15 the proportion of the times the initial solution constructed results in the best solution (over 11 solutions) after the improvement algorithms are applied for the 5- and 10-aisle instances, respectively, averaged over 90 instances. The last column titled “Random, *in total*”, reports the proportion of the times any of the 9 random initial solutions constructed results in the best solution (over 11 solutions), averaged again over 90 instances. Moreover, we evaluate the time it takes for our proposed approach to construct the initial solution and to bring it to the final solution when different construction algorithms are used (see the values in parentheses in the tables). The values in the parentheses in the last column refer to the total solution time of our proposed solution approach generating 9 random solutions and improving all of them to obtain 9 final solutions. The average values of the columns are given in the last rows of the tables.

When Tables 4.14 and 4.15 are examined, it can be seen that the solutions constructed using the nearest neighbor algorithm and DP-Const algorithm result in the best solution with similar proportions on average (0.30 and 0.32 in the 5-aisle instances, and 0.32 and 0.32 in the 10-aisle instances, respectively). The proposed algorithm has a higher chance of finding the best solution when the DP-Const algorithm is used to generate the initial solutions for small size instances. On the other hand, for the larger size instances, it has a higher chance of finding the best solution when the nearest neighbor algorithm is used as the tour construction algorithm. Moreover, a solution

Table 4.14: Average proportion of the times the initial solution constructed results in the best solution and average solution times when different construction algorithms are utilized for 5-aisle instances with different n .

n	Av. proportion resulting in best sol. (Av. Sol. Time)							
	Nearest neighbor		DP-Const		Random, <i>on average</i>		Random, <i>in total</i>	
20	0.71	(3.9)	0.71	(4.2)	0.65	(3.9)	0.98	(35.1)
25	0.44	(6.6)	0.60	(6.3)	0.51	(6.3)	1.00	(56.5)
30	0.33	(8.5)	0.37	(9.4)	0.32	(8.5)	0.91	(76.2)
40	0.27	(10.5)	0.22	(12.2)	0.18	(11.4)	0.90	(102.7)
50	0.13	(14.9)	0.14	(17.6)	0.12	(16.5)	0.86	(148.3)
75	0.11	(29.4)	0.09	(37.8)	0.09	(38.4)	0.81	(346.0)
100	0.13	(63.9)	0.11	(92.0)	0.09	(106.4)	0.76	(957.4)
Av.	0.30	(19.7)	0.32	(25.6)	0.28	(27.3)	0.89	(246.0)

generated by the nearest neighbor algorithm or DP-Const algorithm is more likely to turn into the best solution than a solution generated at random. On the other hand, when 9 random initial solutions are taken, they have a larger chance to get to the best solution than a single nearest neighbor or the DP-Const solution (see the last column of the tables). This observation signifies the importance of number of replications.

When the computational time of the proposed solution approach is examined (values in the parentheses), it can be seen that the approach spends the least time when the nearest neighbor algorithm is used. Even though the construction time of the DP-Const is larger than that of the random construction, the proposed solution approach does not spend more time (including the construction time) when the DP-Const is used. This is because more effort is likely to be needed to improve a random initial solution. Considering the solution quality and the computational time of the final solutions obtained by using different construction algorithms, we suggest the user to run the proposed approach first with 1 solution constructed with the nearest neighbor algorithm. If the user has still some time, s/he can run the approach again by 1

Table 4.15: Average proportion of the times the initial solution constructed results in the best solution and average solution times when different constuction algorithms are utilized for 10-aisle instances with different n .

n	Av. proportion resulting in best sol. (Av. Sol. Time)				
	Nearest neighbor	DP-Const	Random, <i>on average</i>	Random, <i>in total</i>	
20	0.82 (4.1)	0.88 (4.3)	0.75 (4.1)	1.00 (36.9)	
25	0.42 (6.3)	0.50 (7.8)	0.51 (6.5)	0.99 (58.2)	
30	0.36 (8.0)	0.42 (9.6)	0.29 (8.3)	0.92 (75.1)	
40	0.21 (11.4)	0.24 (12.7)	0.17 (10.9)	0.89 (97.9)	
50	0.17 (16.0)	0.13 (19.5)	0.10 (16.8)	0.81 (151.4)	
75	0.09 (32.3)	0.02 (40.6)	0.10 (38.9)	0.89 (350.4)	
100	0.20 (65.9)	0.04 (95.2)	0.09 (106.7)	0.76 (960.2)	
Av.	0.32 (20.6)	0.32 (27.1)	0.29 (27.5)	0.89 (247.2)	

solution constructed with the DP-Const algorithm. Having additional time, the user can continue with replications by generating random initial solutions.

Now, we provide a comprehensive comparison between the solutions of the EMFRP and the HVDMFRP. In Tables 4.16 and 4.17, we present percent energy savings obtained by using the EMFRP instead of the HVDMFRP for the 5- and 10-aisle instances, respectively. The EMFRP instances with $n = 30$ are solved by the proposed solution approach, while all the other instances of the the EMFRP and all the instances of the HVDMFRP are solved to optimality by exact solution methods. The titles of the columns give the number of items while those of the rows give the height-mass settings of the instances. The values in the tables give the average (over 10 instances) percent energy saving, and the least (among 10 instances) percent energy saving in parentheses, when we use the solutions of the EMFRP instead of the HVDMFRP.

Tables 4.16 and 4.17 show that the solutions of the EMFRP can result in up to %30 energy savings compared to the solutions obtained by the HVDMFRP. Generally, we can state that the larger the size of an instance the more energy savings we obtain by

Table 4.16: Average percent energy savings when EMFRP is used instead of HVM-FRP on 5-aisle instances with different n and height-mass settings

Setting	Average % energy saving (min % energy saving among 10 instances)				
	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$
L-L	2.3 (0.0)	3.0 (0.0)	3.6 (1.7)	3.9 (1.1)	4.0 (1.4)
L-H	3.8 (0.0)	6.0 (0.3)	7.2 (4.8)	8.1 (3.3)	9.0 (3.7)
L-G	3.4 (0.0)	4.6 (2.3)	6.2 (3.4)	6.6 (1.3)	7.4 (4.0)
M-L	3.7 (1.5)	6.4 (3.2)	8.3 (4.0)	7.6 (5.0)	11.7 (7.7)
M-H	6.4 (2.0)	11.3 (6.2)	14.1 (8.6)	15.0 (9.2)	22.2 (16.5)
M-G	5.8 (1.2)	9.3 (3.3)	11.9 (6.2)	12.1 (5.1)	18.2 (13.3)
H-L	5.3 (1.7)	7.4 (4.2)	12.8 (5.9)	13.3 (7.2)	18.0 (8.9)
H-H	9.9 (4.4)	12.4 (7.5)	21.4 (14.3)	23.5 (15.4)	30.2 (16.6)
H-G	8.7 (3.4)	11.0 (5.8)	18.2 (11.1)	19.7 (10.5)	26.0 (16.4)

Table 4.17: Average percent energy savings when EMFRP is used instead of HVM-FRP on 10-aisle instances with different n and height-mass settings

Setting	Average % energy saving (min % energy saving among 10 instances)				
	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$
L-L	2.4 (0.0)	3.1 (0.0)	2.9 (1.3)	4.6 (2.4)	4.7 (0.9)
L-H	3.8 (0.2)	6.0 (0.8)	5.8 (3.9)	8.2 (4.8)	9.4 (4.4)
L-G	3.2 (0.0)	5.2 (1.1)	4.6 (1.3)	6.9 (4.0)	7.3 (2.8)
M-L	4.4 (0.0)	5.1 (2.1)	6.0 (1.6)	7.3 (3.5)	8.9 (5.0)
M-H	7.3 (1.7)	9.1 (5.2)	10.7 (3.7)	13.4 (7.4)	17.0 (10.9)
M-G	6.2 (1.7)	7.9 (4.1)	8.7 (3.3)	11.3 (3.8)	12.7 (5.7)
H-L	3.1 (0.0)	6.2 (2.5)	10.6 (4.3)	12.2 (5.4)	14.3 (10.3)
H-H	6.0 (0.0)	11.0 (4.8)	17.8 (8.6)	21.3 (12.0)	25.4 (19.4)
H-G	4.9 (0.0)	9.4 (4.9)	14.6 (5.9)	17.7 (8.8)	20.6 (14.6)

the EMFRP. For all the instances with $n = 15$ (except the ones with setting L-L) and more items, there is always some energy saving when the EMFRP is used.

In addition to the average percent energy saving values presented in the last two tables, we now show the average (over 10 instances) energy consumption of the solutions of the EMFRP and HVDMFRP for each height-mass setting and for different number of items in Figure 4.6. In every part of this figure, which refer to 9 height-mass settings of the instances, the vertical axes show the amount of energy consumption in kilo joules, while the horizontal axes give the number of items in the instances.

Comparing different parts of Figure 4.6, we observe a higher difference in the energy consumption of the solutions of the EMFRP and HVDMFRP as the height difference of the items, the masses of the items, and the number of items increase. For example, the difference in the energy consumption of the solutions of the EMFRP and HVDMFRP is higher when the height difference setting is medium compared to low, and when it is high compared to medium, independent of the other settings. When the mass setting of the items is high, the energy consumption of the solutions of both problems are higher compared to the general and low settings, independent of the other settings. Finally, independent of the height-mass setting of the instances, the difference in the energy consumption obtained by two problems increases by the size of the instances.

The next comparison between the solutions of the HVDMFRP and EMFRP we provide, is in terms of the total (sum of horizontal and vertical) travel distances (in meters), total tour times (in seconds) (which ignores the operator's picking time), and the energy consumptions of the forklift (in joules). In Figure 4.7, we present the average percent deviation of the total travel distances, tour times, and energy consumption values of the solutions of the EMFRP from those of the HVDMFRP. Similar to Figure 4.6, there are 9 subfigures for 9 height-mass settings of the instances with the number of items in their horizontal axes. The vertical axes in these subfigures give the corresponding average percent deviation values for the total travel distances (D), tour times (T), and energy consumptions (E). The negative values signify that the corresponding results of the solutions of the EMFRP are lower than those of the HVDMFRP.

By Figure 4.7, we can observe that the total travel distances of the solutions of the

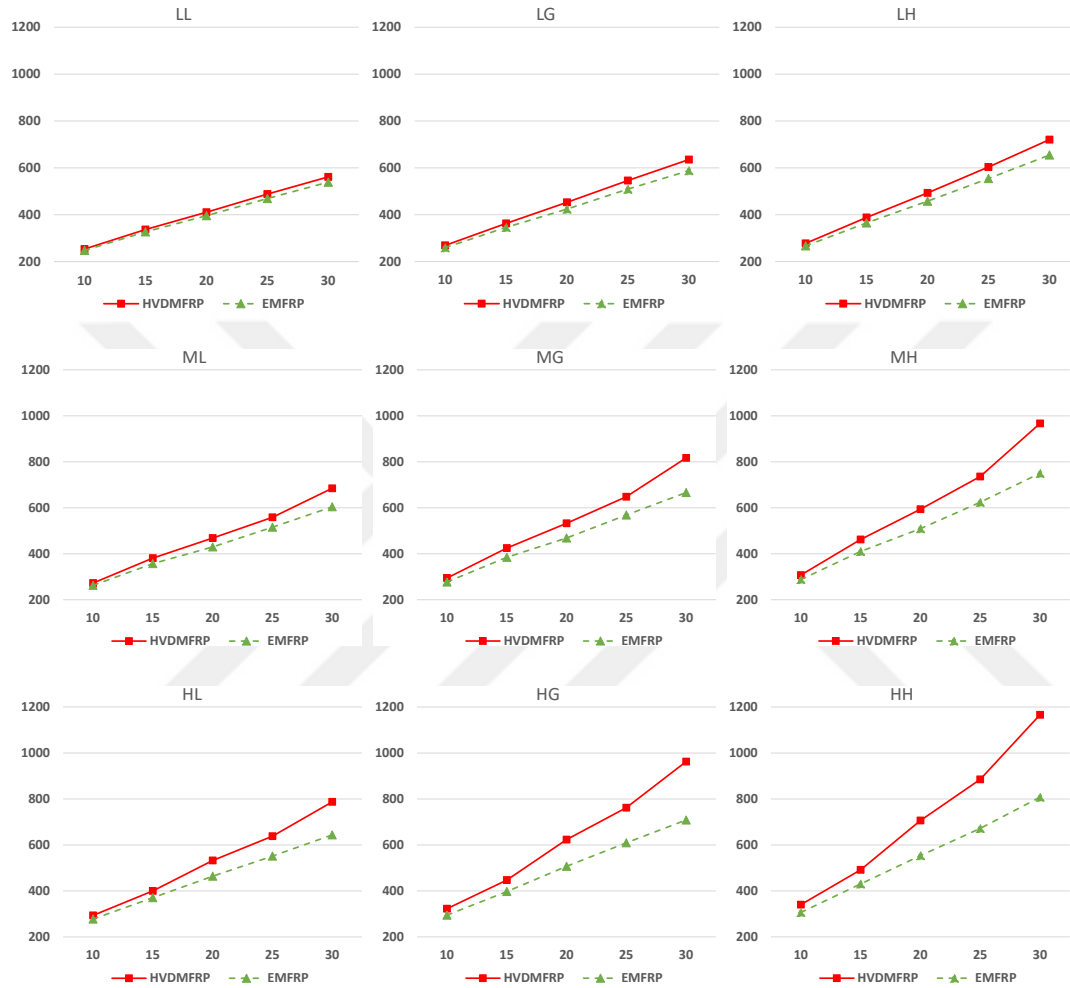


Figure 4.6: Average energy consumption in kilo joule (in vertical axes) when the HVMFRP and EMFRP are used to solve the 5-aisle instances with different height-mass settings and different number of items (in horizontal axes)

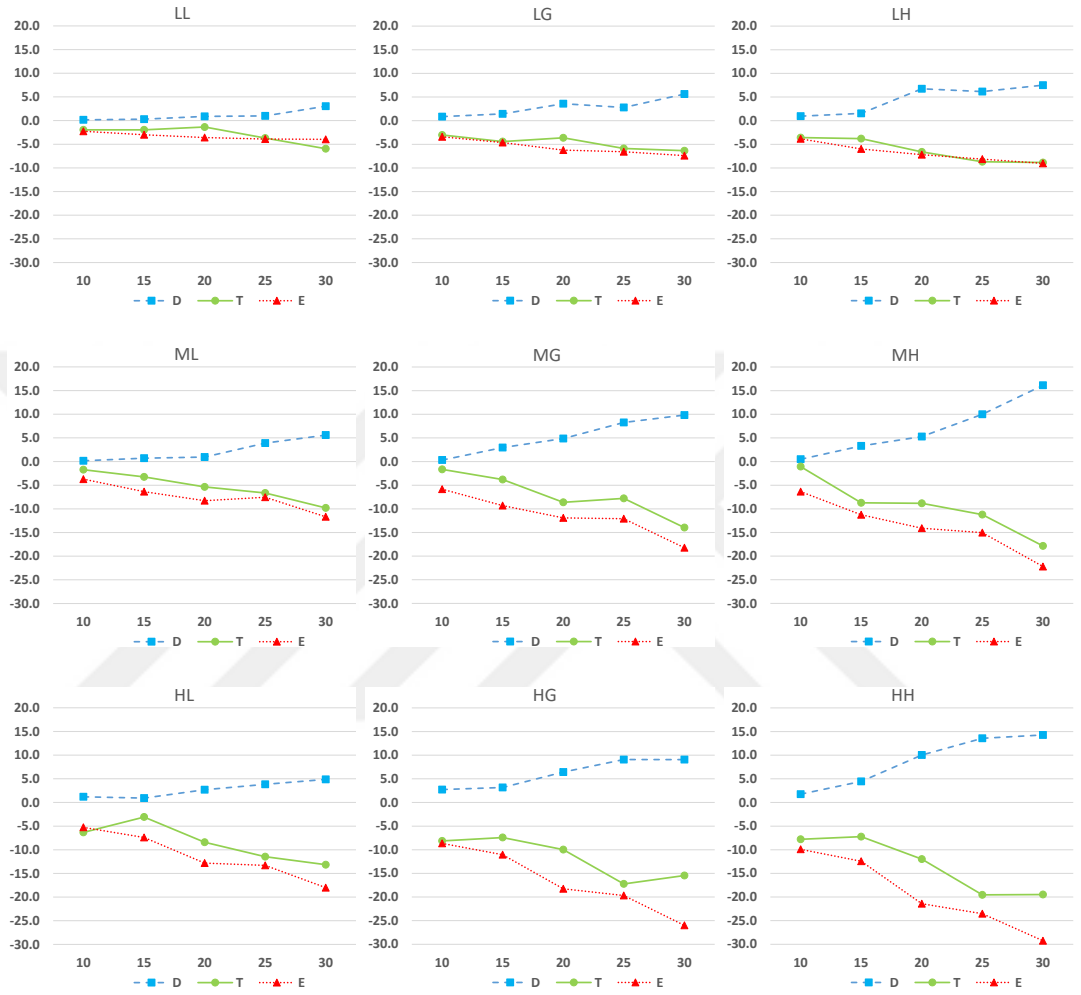


Figure 4.7: Average percent change (in vertical axes) in total horizontal and vertical distances (D), total travel times (T), and total energy consumptions (E) of the forklift when 5-aisle instances with different height-mass settings and different number of items (in horizontal axes) are solved with the EMFRP instead of the HVMFRP

EMFRP are higher compared to those of the HVDMFRP. On the other hand, the average tour times of the solutions of the EMFRP fall lower than those of the HVDMFRP along with the average energy consumption values. These differences become more remarkable, in general, by increase in the height difference, masses, and number of items. In other words, applying the EMFRP instead of the HVDMFRP, results not only in energy savings but also in time savings.

Finally, we present in Figure 4.8 the percent deviations of the horizontal distances (H), vertical distances (V), and total distances (D) traveled by the forklift when the solutions of the EMFRP are used instead of those of the HVDMFRP. Even though the total travel distances and horizontal distances of the solutions of the EMFRP are higher, the EMFRP solutions result in lower energy consumption by reducing the amount of vertical moves significantly. This figure therefore shows us the source of the energy savings obtained by the solutions of the EMFRP. In summary, the EMFRP, when compared with the HVDMFRP, provides energy-efficient order picker forklift tour by trading off some horizontal travel distance against vertical travel distance.

4.6 Concluding Remarks

Routing an order picker forklift to pick the ordered items belongs to the operational decision making level and is done in high frequency resulting in high energy consumption daily. Finding an energy-efficient route for an order picker forklift can yield significant savings in the energy consumption in warehouses. This chapter studies the EMFRP with MIP formulation that seeks the best tour of an order picker forklift, in terms of energy consumption, which is routed to pick a list of items and gather them in depot. In this study, the energy consumption of the forklift in its horizontal and vertical moves is calculated in detail with the friction forces and the load on the forklift as well as its acceleration and deceleration. The MIP formulation of the EMFRP has some characteristics of the TSP while its computational time is highly dependent on the instance settings other than the number of items (instance size). Similar to the MIP formulation, the adapted TSP-based dynamic programming algorithm also can only solve small instances in reasonable time. Therefore, some TSP-based heuristics, i.e., nearest neighbor algorithm, 2-opt and 3-opt algorithms, are adapted for the EMFRP.

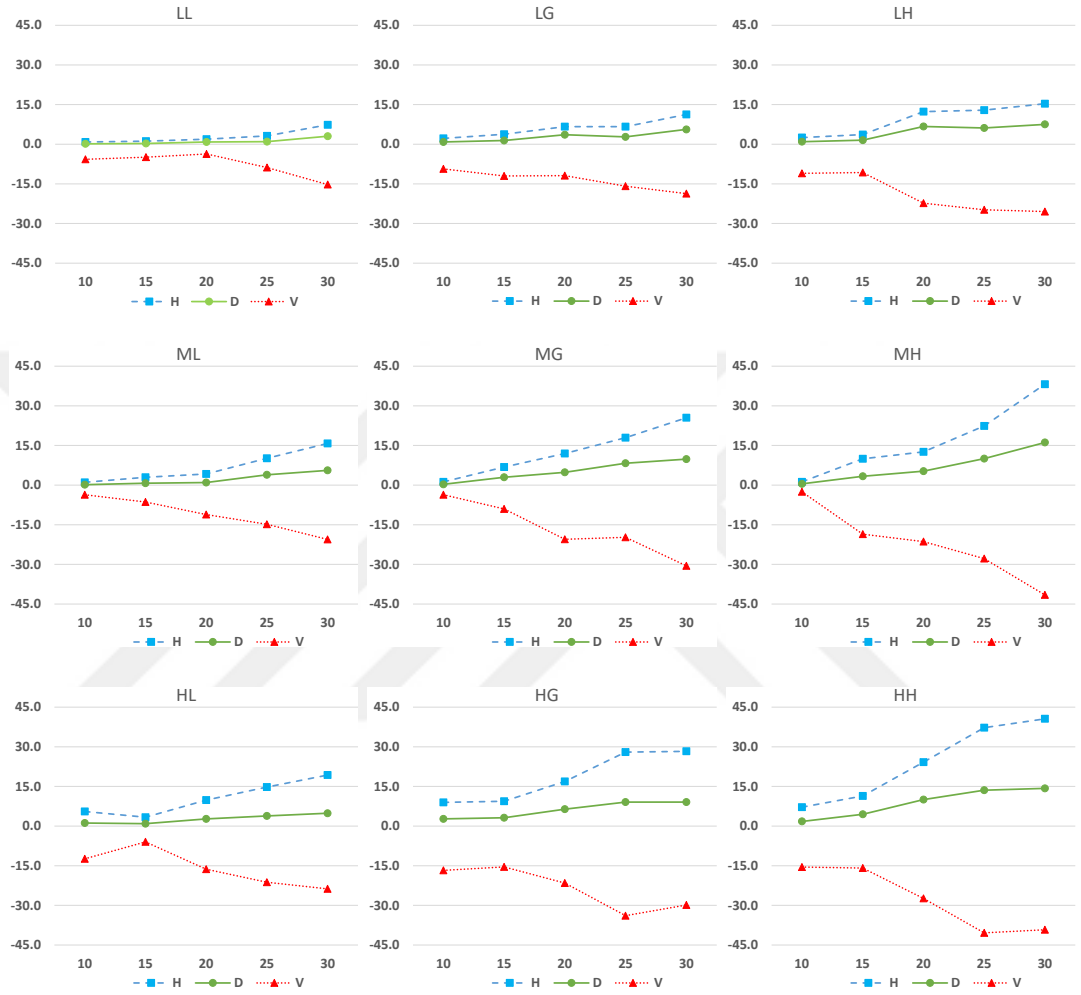


Figure 4.8: Average percent change (in vertical axes) in horizontal distances (H), vertical distances (V), and total horizontal and vertical distances (D) of the forklift when the instances with different height-mass settings and different number of items (in horizontal axes) are solved with the EMFRP instead of the HVMFRP

Moreover, two problem based algorithms, one for constructing an initial tour, other for improving a given tour, are also developed to be integrated to the TSP-based algorithms in order to come up with a single solution approach. The experimental results show that the dynamic programming approach beats the MIP formulation in general and the proposed heuristic approach provides near optimal results. They also point out that for problem instances with at least 25 items, the proposed solution approach can be preferred over the exact solution methods. According to the experiments that compare the solutions of the EMFRP and total distance minimization problem, it can be said that the solutions of the EMFRP does not only provide energy savings but also time savings in the order picker forklifts' tours.



CHAPTER 5

CONCLUSION

In this thesis, Chapter 2 introduces extensions of the classical Weber problem (WP) with CO₂ emission concerns. First we provide the green Weber problem (GWP) as an extension of the single-facility WP. The GWP determines the location of a single facility in the plane and the speeds of the vehicles serving the customers that result in the minimum total amount of CO₂ emission in the distribution system. Here, the customers are assumed to have hard one-sided time windows (deadlines) and the vehicles serving the customers should finish their service on or before the time limits. When there is no deadline or all deadlines are relaxed enough, the optimal facility location of the GWP corresponds to that of the single-facility WP. To the knowledge of us, all the studies in the literature that take the vehicle speeds into account assume that the location of the facility is given a priori. Furthermore, this is one of the few studies in the literature that optimizes the vehicle speeds without using any discretization. Also, all exact solution approaches in the literature are able to solve small to medium size problems. Due to the second order cone programming formulation of the GWP, instances with 1000 customers are solved within a couple of seconds in our study. A worst case complexity of the GWP, due to the SOCP formulation, is $O(n^{3.5})$ Lobo et al. (1998), where n is the number of customers. In the second problem considered, namely the time-dependent green Weber problem (TD-GWP), time-dependent traffic congestion on roads is introduced which limits the vehicle speeds in different time periods. In the TD-GWP, the aim is again the minimization of the total amount of CO₂ emission in the distribution system. The vehicles are allowed to wait in certain time periods in order to reduce the fuel-emission, but are not allowed to violate the customers' deadlines. This problem is again shown to be polynomial-time solvable by formulating it as a second order cone programming problem.

Chapter 3, studies the multi-facility green Weber problem (MF-GWP), which is an extension of the classical multi-facility Weber problem (MF-WP) with environmental considerations. The MF-GWP is also a multi-facility variant of the GWP provided in the second chapter. In the MF-GWP there are a number of customers with pre-determined service deadlines that are to be satisfied by the vehicles (whose speeds are to be optimized) sent from a number of facilities to be located with minimal CO₂ emission objective. We first propose a mixed integer second order cone programming (MISOCP) formulation for the MF-GWP and investigate the effects of the symmetry breaking constraints and the deadlines of the customers on the solution times. Due to the weakness of the formulation, we then develop heuristic solution methods to solve larger problem instances in reasonable time. We propose a local search heuristic, and modified versions of the transfer and decomposition heuristics as improvement stages of the solution method. The MF-GWP and its extensions or modifications may find uses in several different real life applications and therefore their solutions may be used in reducing the CO₂ emissions or energy consumption and enhancing the sustainability and green logistics applications in practice. In this study, we give applications from different sectors including aviation and robotics, where the developed solution methods can be employed by integration of problem specific modifications. Moreover, an illustrative example is considered in an assembly line system where the stations are fed by dedicated rail-guided vehicles (RGVs). For this problem, we modify the MISOCP formulation of the MF-GWP where we also utilize speed limit, capacity, and location constraints along with constraints preventing intersecting rails within the formulation.

The manual or automated material handling systems used in warehouses involve important technology-driven operations such as the usage of high-tech order picker forklifts. Order picker forklifts with their ability to allow order picking operations in narrow aisles and from high-level racks make them widely used in warehouses. Moreover, a single operator does the riding and the picking tasks resulting in no need for additional equipment or operator. The ability to ride the forklift while the fork is lifted provides energy efficiency when picking items of similar heights. Routing the order picker forklift to pick ordered items belongs to the operational decision making level and is done in high frequency resulting in high energy consumption daily.

Therefore, finding an energy-efficient route for an order picker forklift can yield significant savings in the energy consumption in warehouses. For this reason, in this thesis, we studied the energy-minimizing order picker forklift routing problem. To our knowledge, this is the first study that considers the order picker forklifts in the context of the order picking problem. One of the contributions of this study to the literature is the detailed calculation of the forklift's energy consumption in its horizontal and vertical moves taking the friction forces, the load on the forklift, acceleration and deceleration of the forklift into account. An MIP formulation is proposed for the EMFRP which is not able to solve large size problem instances in reasonable time. Moreover, we adapted a TSP-based dynamic programming algorithm for the EMFRP, which solves instances with up to 25 items in less than 20 minutes. Some TSP-based construction and improvement heuristics, i.e., nearest neighbor algorithm, 2-opt and 3-opt algorithms, are adapted for the EMFRP. Two problem based algorithms, one for constructing an initial tour, other for improving a given tour, are also developed both using the dynamic programming approach as the optimization tool in each step. Finally, all construction and improvement heuristics are integrated to develop a single solution approach. Several sets of experiments are performed with the MIP formulation, dynamic programming approach, and proposed solution approach.

5.1 Major Findings

In our computational experiments in Chapter 2, it is shown that locating the facility based on the solution of the single-facility WP (without taking the time limits into account) instead of the GWP results in a higher total amount of CO₂ emission in the distribution system. Also, as expected, it is seen that the higher the traffic congestion, the higher the total-fuel emission cost in the distribution system.

The computational results in Chapter 3 show the effect of each improvement stage of the proposed solution approach on improving the solution quality. Moreover, the results indicate that within a fixed computational time, even though the location-allocation heuristic is able to make more replications, the improvement heuristics considered, i.e., transfer or transfer followed by decomposition, usually find better solutions using less number of replications. We also investigate how the total amount

of CO₂ emitted by the distribution vehicles changes with respect to the number of facilities located. Such considerations are helpful when the number of facilities is not fixed a priori. A major limitation of this study on MF-GWP is the applicability of the solution methods to the planar facility location problems involving direct shipments. Moreover, we are only able to solve small size instances of the MF-GWP to optimality. The results of the illustrative example in the same chapter show that the RGVs prefer higher speeds when they are empty and lower speeds when they are loaded to minimize their energy consumption.

The experimental results in Chapter 4 show that the dynamic programming approach beats the MIP formulation in general and the proposed heuristic solution approach provides near optimal results. They also point out that for problem instances with at least 25 items, the proposed heuristic solution approach can be preferred over the exact solution methods. According to the experiments that compare the solutions of the EMFRP and total distance minimization problem, it can be said that the solutions of the EMFRP does not only provide energy savings but also time savings in the order picker forklifts' tours.

5.2 Future Research Directions

As a future research direction in context of the GWP and TD-GWP, one can work on finding iterative solution approaches (like an extension of the Weiszfeld procedure) for these problems. Furthermore, location and routing decisions can be considered at the same time in a distribution system.

For the MF-GWP, one can involve routing decisions along with the multi-facility location decisions. This extension makes the problem even more complicated and is left as a future research direction. In designing distribution networks, one cannot ignore the economic aspects of the system. As in the bi-objective pollution routing problem Demir et al. (2014a), the cost of the distribution system including the fixed and variable costs of the facilities and the transportation cost can be considered as another objective function in addition to the total amount of CO₂ emitted in the distribution system. Such multi-objective considerations are crucial and can be addressed

in future studies.

As possible future research directions for the EMFRP, the problems of order batching and order sequencing can be integrated to the EMFRP to achieve energy efficiency. Along with these extensions, the variable speed of the forklift in its horizontal movements as well as the variable lowering and lifting speeds of the fork can be considered in cases where the due times exist. Consideration of a bi-objective version of the EMFRP, where the travel time or the customer waiting time (or some other objective) and energy consumption minimization are simultaneously considered, would be another future research direction.





REFERENCES

- Aktürk, M. S., Atamtürk, A., & Gürel, S. (2014). Aircraft rescheduling with cruise speed control. *Operations Research*, 62, 829–845.
- Alizadeh, F., & Goldfarb, D. (2003). Second-order cone programming. *Mathematical programming*, 95, 3–51.
- Anand, V., Lee, S., & Prabhu, V. (2014). Energy-aware models for warehousing operations. In *Advances in Production Management Systems IFIP Advances in Information and Communication Technology* (pp. 390–397). Springer New York LLC. doi:10.1007/978-3-662-44736-9_48 iFIP WG 5.7 International Conference on Advances in Production Management Systems, APMS 2014 ; Conference date: 20-09-2014 Through 24-09-2014.
- Atashi Khoei, A., Süral, H., & Tural, M. K. (2017). Time-dependent green Weber problem. *Computers & Operations Research*, 88, 316–323. doi:<https://doi.org/10.1016/j.cor.2017.04.010>.
- Atashi Khoei, A., Süral, H., & Tural, M. K. (2020). Multi-facility green weber problem. *Computers & Operations Research*, 113, 104780. doi:<https://doi.org/10.1016/j.cor.2019.104780>.
- Barth, M., Younglove, T., & Scora, G. (2005). *Development of a Heavy-Duty Diesel Modal Emissions and Fuel Consumption Model*. Technical Report UC Berkeley: California Partners for Advanced Transit and Highways, San Francisco.
- Bartolini, M., Bottani, E., & Grosse, E. H. (2019). Green warehousing: Systematic literature review and bibliometric analysis. *Journal of Cleaner Production*, 226, 242–258. doi:<https://doi.org/10.1016/j.jclepro.2019.04.055>.
- Bektaş, T., Ehmke, J. F., Psaraftis, H. N., & Puchinger, J. (2019). The role of operational research in green freight transportation. *European Journal of Op-*

- erational Research*, 274, 807 – 823. doi:<https://doi.org/10.1016/j.ejor.2018.06.001>.
- Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, 45, 1232 – 1250. doi:<http://dx.doi.org/10.1016/j.trb.2011.02.004>.
- Bellman, R. (1962). Dynamic programming treatment of the travelling salesman problem. *J. ACM*, 9, 61–63. doi:<https://doi.org/10.1145/321105.321111>.
- Bongartz, I., Calamai, P. H., & Conn, A. R. (1994). A projection method for ℓ_p norm location-allocation problems. *Mathematical Programming*, 66, 283–312.
- Brimberg, J., & Drezner, Z. (2013). A new heuristic for solving the p -median problem in the plane. *Computers & Operations Research*, 40, 427–437. doi:<https://doi.org/10.1016/j.cor.2012.07.012>.
- Brimberg, J., Hansen, P., Mladenović, N., & Salhi, S. (2008). A survey of solution methods for the continuous location allocation problem. *International Journal of Operations Research*, 5, 1–12.
- Brimberg, J., Hansen, P., Mladenović, N., & Taillard, E. D. (2000). Improvements and comparison of heuristics for solving the uncapacitated multisource Weber problem. *Operations Research*, 48, 444–460. doi:<https://doi.org/10.1287/opre.48.3.444.12431>.
- Çelik, M., & Süral, H. (2019). Order picking in parallel-aisle warehouses with multiple blocks: complexity and a graph theory-based heuristic. *International Journal of Production Research*, 57, 888–906. doi:<https://doi.org/10.1080/00207543.2018.1489154>.
- Chandrasekaran, R., & Tamir, A. (1989). Open questions concerning Weiszfeld's algorithm for the Fermat-Weber location problem. *Mathematical Programming*, 44, 293–295. doi:10.1007/BF01587094.
- Charkhgard, H., & Savelsbergh, M. (2015). Efficient algorithms for travelling salesman problems arising in warehouse order picking. *The ANZIAM Journal*, 57, 166 – 174. doi:10.1017/S1446181115000140.

- Chen, C., Qiu, R., & Hu, X. (2018). The location-routing problem with full truck-loads in low-carbon supply chain network designing. *Mathematical Problems in Engineering*, 2018, 1–13.
- Cooper, L. (1964). Heuristic methods for location-allocation problems. *SIAM Review*, 6, 37–53. doi:10.1137/1006005.
- Cortés, P., Gómez-Montoya, R. A., Muñuzuri, J., & Correa-Espinal, A. (2017). A tabu search approach to solving the picking routing problem for large- and medium-size distribution centres considering the availability of inventory and k heterogeneous material handling equipment. *Applied Soft Computing*, 53, 61 – 73. doi:https://doi.org/10.1016/j.asoc.2016.12.026.
- De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182, 481 – 501. doi:https://doi.org/10.1016/j.ejor.2006.07.009.
- De Koster, R., & Ven der Poort, E. S. (1998). Routing order pickers in a warehouse: a comparison between optimal and heuristic solutions. *IIE Transactions*, 30, 469 – 480. doi:10.1023/A:1007599307171.
- Demir, E., Bektaş, T., & Laporte, G. (2011). A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16, 347 – 357. doi:http://dx.doi.org/10.1016/j.trd.2011.01.011.
- Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223, 346 – 359. doi:http://dx.doi.org/10.1016/j.ejor.2012.06.044.
- Demir, E., Bektaş, T., & Laporte, G. (2014a). The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232, 464 – 478. doi:http://dx.doi.org/10.1016/j.ejor.2013.08.002.
- Demir, E., Bektaş, T., & Laporte, G. (2014b). A review of recent research on green

- road freight transportation. *European Journal of Operational Research*, 237, 775–793. doi:<http://dx.doi.org/10.1016/j.ejor.2013.12.033>.
- Drezner, Z., Brimberg, J., Mladenović, N., & Salhi, S. (2015). New heuristic algorithms for solving the planar p -median problem. *Computers & Operations Research*, 62, 296–304. doi:<https://doi.org/10.1016/j.cor.2014.05.010>.
- Drezner, Z., Brimberg, J., Mladenović, N., & Salhi, S. (2016). New local searches for solving the multi-source Weber problem. *Annals of Operations Research*, 246, 181–203. doi:[10.1007/s10479-015-1797-5](https://doi.org/10.1007/s10479-015-1797-5).
- Drezner, Z., & Hamacher, H. (2002). *Facility Location: Applications and Theory*. Berlin; New York: Springer.
- Drezner, Z., & Salhi, S. (2017). Incorporating neighborhood reduction for the solution of the planar p -median problem. *Annals of Operations Research*, 258, 639–654. doi:[10.1007/s10479-015-1961-y](https://doi.org/10.1007/s10479-015-1961-y).
- Dukkanci, O., Bektaş, T., & Kara, B. Y. (2019). Chapter 7 - Green network design problems. In J. Faulin, S. E. Grasman, A. A. Juan, & P. Hirsch (Eds.), *Sustainable Transportation and Smart Logistics* (pp. 169 – 206). Elsevier. doi:<https://doi.org/10.1016/B978-0-12-814242-4.00007-7>.
- Ene, S., Küçükoğlu, İ., Aksoy, A., & Öztürk, N. (2016). A genetic algorithm for minimizing energy consumption in warehouses. *Energy*, 114, 973 – 980. doi:<https://doi.org/10.1016/j.energy.2016.08.045>.
- Facchini, F., Boenzi, F., Digiesi, S., Mossa, G., & Mummolo, G. (2015). Greening activities in warehouses: A model for identifying sustainable strategies in material handling. (pp. 0980–0988). doi:[10.2507/26th.daaam.proceedings.138](https://doi.org/10.2507/26th.daaam.proceedings.138).
- Facchini, F., Boenzi, F., Mummolo, G., Mossa, G., Digiesi, S., & Verriello, R. (2016). Minimizing the carbon footprint of material handling equipment: comparison of electric and lpg forklifts. *Journal of Industrial Engineering and Management*, 9, 1035–1046. doi:<http://dx.doi.org/10.3926/jiem.2082>.

- Fichtinger, J., Ries, J. M., Grosse, E. H., & Baker, P. (2015). Assessing the environmental impact of integrated inventory and warehouse management. *International Journal of Production Economics*, 170, 717 – 729. doi:<https://doi.org/10.1016/j.ijpe.2015.06.025>.
- Freis, J., Vohlidka, P., & Günthner, W. A. (2016). Low-Carbon Warehousing: Examining Impacts of Building and Intra-Logistics Design Options on Energy Demand and the CO₂ Emissions of Logistics Centers. *Sustainability*, 8, 1–36. doi:<https://doi.org/10.3390/su8050448>.
- Gürel, S., Gultekin, H., & Akhlaghi, V. E. (2019). Energy conscious scheduling of a material handling robot in a manufacturing cell. *Robotics and Computer-Integrated Manufacturing*, 58, 97 – 108.
- Harris, I., Mumford, C. L., & Naim, M. M. (2014). A hybrid multi-objective approach to capacitated facility location with flexible store allocation for green logistics modeling. *Transportation Research Part E: Logistics and Transportation Review*, 66, 1–22. doi:<https://doi.org/10.1016/j.tre.2014.01.010>.
- Held, M., & Karp, R. M. (1962). A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10, 196–210. doi:<https://doi.org/10.1137/0110015>.
- Henn, S., & Wäscher, G. (2012). Tabu search heuristics for the order batching problem in manual order picking systems. *European Journal of Operational Research*, 222, 484 – 494. doi:<https://doi.org/10.1016/j.ejor.2012.05.049>.
- IBM (2012). Cplex optimizer (11.0). URL: <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>.
- Kara, İ., Kara, B. Y., & Yetis, M. K. (2007). Energy minimizing vehicle routing problem. In *Combinatorial Optimization and Applications* (pp. 62–71). Berlin, Heidelberg: Springer.
- Katz, I. N. (1974). Local convergence in Fermat’s problem. *Mathematical Programming*, 6, 89–104. doi:10.1007/BF01580224.

- Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). The impact of depot location, fleet composition and routing on emissions in city logistics. *Transportation Research Part B: Methodological*, 84, 81–102. doi:<https://doi.org/10.1016/j.trb.2015.12.010>.
- Kramer, R., Subramanian, A., Vidal, T., & dos Anjos F. Cabral, L. (2015). A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, 243, 523 – 539. doi:<http://dx.doi.org/10.1016/j.ejor.2014.12.009>.
- Kuhn, H. W. (1973). A note on Fermat's problem. *Mathematical Programming*, 4, 98–107. doi:[10.1007/BF01584648](https://doi.org/10.1007/BF01584648).
- Lobo, M., Vandenberghe, L., Boyd, S., & Lebrete, H. (1998). Applications of second-order cone programming. *Linear Algebra and its Applications*, 284, 193–228. doi:[10.1016/S0024-3795\(98\)10032-0](https://doi.org/10.1016/S0024-3795(98)10032-0).
- Lozano, S., Guerrero, F., Onieva, L., & Larrañeta, J. (1998). Kohonen maps for solving a class of location-allocation problems. *European Journal of Operational Research*, 108, 106–117. doi:[https://doi.org/10.1016/S0377-2217\(97\)00046-5](https://doi.org/10.1016/S0377-2217(97)00046-5).
- Makris, P., Makri, A., & Provatidis, C. (2006). Energy-saving methodology for material handling applications. *Applied Energy*, 83, 1116 – 1124. doi:<https://doi.org/10.1016/j.apenergy.2005.11.005>.
- Makris, P. A., & Giakoumakis, I. G. (2003). k-interchange heuristic as an optimization procedure for material handling applications. *Applied Mathematical Modelling*, 27, 345 – 358. doi:[https://doi.org/10.1016/S0307-904X\(02\)00137-3](https://doi.org/10.1016/S0307-904X(02)00137-3).
- Masae, M., Glock, C. H., & Grosse, E. H. (2020). Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*, 224, 107564. doi:<https://doi.org/10.1016/j.ijpe.2019.107564>.
- Megiddo, N., & Supowit, K. J. (1984). On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13, 182–196.

- Meriam, J., & Kraige, L. (2012). *Engineering Mechanics: Dynamics, 7th edition*. Hoboken, NJ, USA: John Wiley & Sons, Inc.
- Nilsson, C. (2003). *Heuristics for the traveling salesman problem*. Technical Report Linköping University.
- OECD (2010) (). Reducing transport greenhouse gas emissions: Trends and data. The International Transport Forum, Paris, <https://www.itf-oecd.org/sites/default/files/docs/10ghgtrends.pdf>.
- OR Library (). Multi-source Weber problems. <http://mistic.heig-vd.ch/taillard/problemes.dir/location.html>, [Online; accessed 13-March-2019].
- Piecyk, M., Browne, M., Whiteing, A., & McKinnon, A. (2015). *Green Logistics: Improving the Environmental Sustainability of Logistics*. Kogan Page.
- Ratliff, H., & Rosenthal, A. (1983). Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research*, 31, 507–521. doi:10.1287/opre.31.3.507.
- Rojanapitoon, T., & Teeravaraprug, J. (2018). A computer simulation for economical order picker routing when considering travel distance and vehicle energy consumption. *International Journal of Engineering & Technology*, 7, 33–37. doi:10.14419/ijet.v7i2.28.12878.
- Roodbergen, K. J. (2001). *Layout and Routing Methods for Warehouses*. Ph.D. thesis Erasmus University Rotterdam. <Http://hdl.handle.net/1765/861>.
- Rüdiger, D., Schön, A., & Dobers, K. (2016). Managing greenhouse gas emissions from warehousing and transshipment with environmental performance indicators. *Transportation Research Procedia*, 14, 886 – 895. doi:<https://doi.org/10.1016/j.trpro.2016.05.083>. Transport Research Arena TRA2016.
- Saka, O. C. (2013). *Local search heuristics for pollution-routing problem with multiple vehicle types and deadlines*. Master's thesis at Middle East Technical University, Turkey.
- Santis, R. D., Montanari, R., Vignali, G., & Bottani, E. (2018). An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in

- manual warehouses. *European Journal of Operational Research*, 267, 120 – 137. doi:<https://doi.org/10.1016/j.ejor.2017.11.017>.
- Scholz, A., Henn, S., Stuhlmann, M., & Wäscher, G. (2016). A new mathematical programming formulation for the single-picker routing problem. *European Journal of Operational Research*, 253, 68 – 84. doi:<https://doi.org/10.1016/j.ejor.2016.02.018>.
- Senzig, D., & Cumper, J. (2013). *Fuel consumption of ADS-B and non-ADS-B helicopter operations in the Gulf of Mexico*. Technical Report John A. Volpe National Transportation Systems Center (U.S.).
- Tachibana, T., & Adachi, M. (2014). Method for solving optimization problems using algorithm switching by chaotic neural networks. *Proc. of NOLTA 2014*, (pp. 329–332).
- The Population Division of the Department of Economic and Social Affairs of the United Nations (2014). World urbanization prospects: 2014 revision. Available at <http://esa.un.org/unpd/wup/>, accessed: January 25th, 2021.
- Theys, C., Bräysy, O., Dullaert, W., & Raa, B. (2010). Using a tsp heuristic for routing order pickers in warehouses. *European Journal of Operational Research*, 200, 755 – 763. doi:<https://doi.org/10.1016/j.ejor.2009.01.036>.
- Tokekar, P., Karnad, N., & Isler, V. (2011). Energy-optimal velocity profiles for car-like robots. *2011 IEEE International Conference on Robotics and Automation*, (pp. 1457–1462).
- Toro, E. M., Franco, J. F., Echeverri, M. G., & Guimarães, F. G. (2017). A multi-objective model for the green capacitated location-routing problem considering environmental impact. *Computers & Industrial Engineering*, 110, 114–125. doi:<https://doi.org/10.1016/j.cie.2017.05.013>.
- Vardi, Y., & Zhang, C.-H. (2001). A modified Weiszfeld algorithm for the Fermat-Weber location problem. *Mathematical Programming*, 90, 559–566.
- Waltho, C., Elhedhli, S., & Gzara, F. (2019). Green supply chain network design: A review focused on policy adoption and emission quantification. *International*

- Journal of Production Economics*, 208, 305 – 318. doi:<https://doi.org/10.1016/j.ijpe.2018.12.003>.
- Weiszfeld, E., & Plastria, F. (2009). On the point for which the sum of the distances to n given points is minimum. *Annals of Operations Research*, 167, 7–41. doi:10.1007/s10479-008-0352-z.
- World Economic Forum (2009) (). Supply chain decarbonization. World Economic Forum Report, Geneva, <http://reports.weforum.org/supply-chain-decarbonization-info/>.
- World Economic Forum (2016) (). World economic forum white paper digital transformation of industries: In collaboration with accenture. <Http://reports.weforum.org/digital-transformation/wp-content/blogs.dir/94/mp/files/pages/files/wef-dti-logisticswhitepaper-final-january-2016.pdf>, Accessed date: 28 November 2020.
- Xiao, Y., Zhao, Q., Kaku, I., & Xu, Y. (2012). Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research*, 39, 1419–1431. doi:<https://doi.org/10.1016/j.cor.2011.08.013>.
- Xifeng, T., Ji, Z., & Peng, X. (2013). A multi-objective optimization model for sustainable logistics facility location. *Transportation Research Part D: Transport and Environment*, 22, 45–48. doi:<https://doi.org/10.1016/j.trd.2013.03.003>.
- Zebrowski, P., Litus, Y., & Vaughan, R. T. (2007). Energy efficient robot rendezvous. In *Fourth Canadian Conference on Computer and Robot Vision (CRV '07)* (pp. 139–148).