VEHICLE ROUTING PROBLEM IN CROSS DOCKS WITH SHIFT-BASED TIME CONSTRAINTS ON PRODUCTS

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

BY

MENEKŞE KOÇAK

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

DECEMBER 2011

Approval of the thesis:

VEHICLE ROUTING PROBLEM IN CROSS DOCKS WITH SHIFT-BASED TIME CONSTRAINTS ON PRODUCTS

submitted by **MENEKŞE KOÇAK** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ______________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Sinan Kayalıgil ______________
Head of Department, **Industrial Engineering**

Assoc. Prof. Dr. Canan Sepil ______________
Supervisor, **Industrial Engineering Dept.**, **METU**

Examining Committee Members:

Prof. Dr. Gülser Köksal ______________
Industrial Engineering Dept., METU

Assoc. Prof. Dr. Canan Sepil ______________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Seçil Savaşaneril ______________
Industrial Engineering Dept., METU

Assist. Prof. Dr. İsmail Serdar Bakal ______________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Ferda Can Çetinkaya ______________
Industrial Engineering Dept., Çankaya University

Date: 09.12.2011

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

Name, Last name : MENEKŞE KOÇAK

Signature :

ABSTRACT

VEHICLE ROUTING PROBLEM IN CROSS DOCKS
WITH SHIFT-BASED TIME CONSTRAINTS ON PRODUCTS

Koçak, Menekşe

M.Sc., Department of Industrial Engineering
Supervisor: Assoc. Prof. Dr. Canan Sepil

December 2011, 106 pages

In this study, the capacitated vehicle routing problem with shift based time constraints is taken into consideration. The study stemmed from an application in a cross dock. The considered cross dock is assumed to feed directly the production lines of its customer. The customer has a just-in-time production system that requires producing only in necessary quantities at the necessary times. This necessitates the arrival of the parts/products collected from different suppliers at the customer at the beginning of each shift of production. The shift times constitute deadlines for the products to be collected from the suppliers and used in each shift. The collection problem then can be seen as the capacitated vehicle routing problem with shift based time constraints. The objective of the collection problem is to minimize the routing costs. For the accomplishment of this objective it is required to decide on products of which shift(s) should be taken from a supplier when a vehicle arrives at that supplier. For the solution of the problem a mathematical model is formulated. Since the dealt problem is NP-Hard, meta-heuristic solution approaches based on variable neighborhood search and simulated annealing are proposed. Computational experimentation is conducted on the test problems which are tailored from the capacitated vehicle routing instances from the literature.

Keywords: Vehicle routing problem, shift-based time constraints on products, cross docks, variable neighborhood search, simulated annealing

ÖZ

ÜRÜNLER ÜZERİNDEKİ VARDİYA BAZLI ZAMAN KISITLARI İLE ÇAPRAZ SEVKİYAT DEPOLARINDA ARAÇ ROTALAMA PROBLEMİ

Koçak, Menekşe

Yüksek Lisans, Endüstri Mühendisliği
Tez Yöneticisi: Doç. Dr. Canan Sepil

Aralık 2011, 106 sayfa

Bu çalışmada, kapasiteli araçlarla vardiya bazlı zaman kısıtları ile araç rotalama problemi değerlendirilmiştir. Çalışma çapraz sevkiyat deposundaki bir uygulamadan gelmektedir. İncelenen çapraz sevkiyat deposunun, müşterisinin üretim hattını direkt olarak beslediği varsayılmıştır. Müşteri sadece gerekli zamanlarda gerekli miktarları üretmeyi gerektiren tam zamanında üretim sistemine sahiptir. Bu, farklı tedarikçilerden toplanan parçaların/ürünlerin, müşteriye her bir üretim vardiyasının başlangıcında gelmesini gerektirmektedir. Vardiya zamanları, tedarikçilerden toplanacak ve her bir vardiyada kullanılacak ürünler için son teslim tarihlerini oluşturmaktadır. Bu durumda, toplama problemi vardiya bazlı zaman kısıtları ile kapasiteli araç rotalama problemi olarak görülebilir. Toplama probleminin amacı rotalama maliyetlerini en aza indirmektir. Bu amacın başarılmasında, bir araç tedarikçiye ulaştığında hangi vardiya(ların) ürünlerinin alınması gerektiğine karar verilmesi gerekmektedir. Problemin çözümü için bir matematiksel model formüle edilmiştir. Ele alınan problem NP-Zor olduğu için, değişken komşu arama ve tavlama benzetimine dayanan meta-sezgisel çözüm yöntemleri de önerilmiştir. Sayısal deneyler literatürdeki kapasiteli araç rotalama örneklerinden uyarlanan problem seti üzerinde yürütülmüştür.

Anahtar Kelimeler: Araç rotalama problemi, ürünler üzerinde vardiya bazlı zaman kısıtları, çapraz sevkiyat depoları, değişken komşu arama, tavlama benzetimi

To My Family

ACKNOWLEDGEMENTS

I would like to express my deepest and sincerest gratitude to my supervisor Assoc. Prof. Dr. Canan Sepil for her continuous guidance, advice, support and motivation throughout the preparation of the thesis. I am very grateful to her for constracting the frames of the thesis and helping to complete it with a great sacrifice.

I would like to express my thanks to Prof. Dr. Gülser Köksal, Assist. Prof. Dr. Seçil Savaşaneril, Assist. Prof. Dr. İsmail Serdar Bakal and Assist. Prof. Dr. Ferda Can Çetinkaya for their valuable comments and advices that really assist in forming the thesis.

I am very grateful to my parents, Yaşar and Suna Koçak and my sister, Zeynep Koçak, who are always supporting and motivating me. It is a inexpressible pleasure to thank them who made this thesis possible for me with their patience and encouragement. They always believed in me and strengthen me in my hard times with their great love, compassion and leading advices.

Finally, I would like to express my profound thanks to my husband, Çağrı Düzce, who is always the greatest supporter to me in each and every moment. I am wholeheartedly very grateful to him who always ease the life for me with his continuous support, ideas, patience, encouragement and motivation. It would not be possible to carry out this study without his endless love.

TABLE OF CONTENTS

ABSTRACT....iv
ÖZ....v
ACKNOWLEDGEMENTS.... vii
TABLE OF CONTENTS.... viii
LIST OF TABLES....x
LIST OF FIGURES....xiv
CHAPTERS
1. INTRODUCTION....1
2. LITERATURE REVIEW....6
2.1 Literature Review on Cross Docks....6
2.2 Literature Review on Vehicle Routing....14
2.3 Comparison....23
3. MATHEMATICAL MODELLING....25
3.1 Assumptions....25
3.2 Sets, Parameters and Decision Variables....26
3.3 Mathematical Model....28
4. META-HEURISTIC SOLUTION APPROACHES....32
4.1 Variable Neighborhood Search....32
4.2 Simulated Annealing....36
4.3 Proposed Variable Neighborhood Search Algorithm....38
4.4 Proposed Simulated Annealing Algorithm....44

5. COMPUTATIONAL EXPERIMENTS..50
5.1 Problem Sets...50
5.2 Parameter Setting...52
5.2.1 Parameter Setting for Variable Neighborhood Search 53
5.2.2 Parameter Setting for Simulated Annealing 56
5.3 Computational Results..59
5.3.1 Results of the Test Runs 59
5.3.2 Detailed Analysis on the Results 66
6. CONCLUSIONS 72
REFERENCES...75
APPENDICES
A. COMPUTATIONAL RESULTS FOR THE FIRST SCENARIO...............83
B. COMPUTATIONAL RESULTS FOR THE SECOND SCENARIO..........89
C. COMPUTATIONAL RESULTS FOR THE THIRD SCENARIO 95
D. COMPUTATIONAL RESULTS FOR THE 2-opt* APPROACH USING SECOND SCENARIO 101

LIST OF TABLES

TABLES

Table 1. Literature Review for Cross-Dock Operations 9

Table 2. Taxonomy of the VRP Literature (Ekşioğlu et al. (2009)) 15

Table 3. CVRP Literature Instances...51

Table 4. Latest Arrival Time Scenarios for Products.......................................52

Table 5. List of k_{max} and Step Values Assigned for Parameter Setting 53

Table 6. Evaluation of algorithm as per each k_{max} and step combination using instance 2..55

Table 7. CPU Time and Best Solutions for Different α Values........................58

Table 8. Results for scenario 1, where the shift deadlines are determined by the longest tour time..61

Table 9. Results for scenario 2, where the shift deadlines are determined by 1.2×the longest tour time 63

Table 10. Results for scenario 3, where the shift deadlines are determined by 1.4×the longest tour time 65

Table 11. Results for scenario 2, where the shift deadlines are determined by 1.2×the longest tour time using 2-opt* approach in local search......................70

Table A - 1. Results for Christophides and Eilon (n22 - v4) Problem for the First Scenario 84

Table A - 2. Results for Christophides and Eilon (n23 - v3) Problem for the First Scenario 84

Table A - 3. Results for Christophides and Eilon (n30 - v3) Problem for the First Scenario 85

Table A - 4 . Results for Augerat, et al. (n37 - v5) Problem for the First Scenario 85

Table A - 5. Results for Augerat, et al. (n40 - v5) Problem for the First Scenario 86

Table A - 6. Results for Augerat, et al. (n45 - v7) Problem for the First Scenario 86

Table A - 7. Results for Augerat, et al. (n65 - v10) Problem for the First Scenario 87

Table A - 8. Results for Augerat, et al. (n78 - v10) Problem for the First Scenario 87

Table A - 9. Results for Christophides and Eilon (n101 - v8) Problem for the First Scenario 88

Table B - 1. Results for Christophides and Eilon (n22 - v4) Problem for the Second Scenario 90

Table B - 2. Results for Christophides and Eilon (n23 - v3) Problem for the Second Scenario 90

Table B - 3. Results for Christophides and Eilon (n30 - v3) Problem for the Second Scenario 91

Table B - 4. Results for Augerat, et al. (n37 - v5) Problem for the Second Scenario 91

Table B - 5. Results for Augerat, et al. (n40 - v5) Problem for the Second Scenario 92

Table B - 6. Results for Augerat, et al. (n45 - v7) Problem for the Second Scenario 92

Table B - 7. Results for Augerat, et al. (n65 - v10) Problem for the Second

Scenario....93

Table B - 8. Results for Augerat, et al. (n78 - v10) Problem for the Second Scenario....93

Table B - 9. Results for Christophides and Eilon (n101 - v8) Problem for the Second Scenario....94

Table C - 1. Results for Christophides and Eilon (n22 - v4) Problem for the Third Scenario....96

Table C - 2. Results for Christophides and Eilon (n23 - v3) Problem for the Third Scenario....96

Table C - 3. Results for Christophides and Eilon (n30 - v3) Problem for the Third Scenario....97

Table C - 4. Results for Augerat, et al. (n37 - v5) Problem for the Third Scenario....97

Table C - 5. Results for Augerat, et al. (n40 - v5) Problem for the Third Scenario....98

Table C - 6. Results for Augerat, et al. (n45 - v7) Problem for the Third Scenario....98

Table C - 7. Results for Augerat, et al. (n65 - v10) Problem for the Third Scenario....99

Table C - 8. Results for Augerat, et al. (n78 - v10) Problem for the Third Scenario....99

Table C - 9. Results for Christophides and Eilon (n101 - v8) Problem for the Third Scenario....100

Table D - 1. Results for Christophides and Eilon (n22 - v4) Problem for the Second Scenario....102

Table D - 2. Results for Christophides and Eilon (n23 - v3) Problem for the

Second Scenario 102

Table D - 3. Results for Christophides and Eilon (n30 - v3) Problem for the Second Scenario 103

Table D - 4. Results for Augerat, et al. (n37 - v5) Problem for the Second Scenario 103

Table D - 5. Results for Augerat, et al. (n40 - v5) Problem for the Second Scenario 104

Table D - 6. Results for Augerat, et al. (n45 - v7) Problem for the Second Scenario 104

Table D - 7. Results for Augerat, et al. (n65 - v10) Problem for the Second Scenario 105

Table D - 8. Results for Augerat, et al. (n78 - v10) Problem for the Second Scenario 105

Table D - 9. Results for Christophides and Eilon (n101 - v8) Problem for the Second Scenario 106

LIST OF FIGURES

FIGURES

Figure 1. Schematic Representation of cross-dock operations 1

Figure 2. Parts Flow in the Cross-Dock System 3

Figure 3. Basic Steps of VNS 34

Figure 4. Steps of Local Search Algorithm..35

Figure 5. Swap Approach Used in the Shaking Step 39

Figure 6. Product Exchange Approach for Shaking Step 40

Figure 7. Cheapest Insertion Approach for Local Search 41

Figure 8. Pseudo Code for Variable Neighborhood Search..............................42

Figure 9. Insert Approach Used in Simulated Annealing 45

Figure 10. Swap Route Approach for Simulated Annealing............................46

Figure 11. Random Route Assignment Approach for Simulated Annealing....47

Figure 12. Pseudo Code for Simulated Annealing...48

Figure 13. Results for proposed k_{max} and step values using instance 2.............54

Figure 14. 2-opt* Approach for Local Search 69

CHAPTER 1

INTRODUCTION

In this study, we aim to develop a model in order to assist logistics decisions in cross-docking systems. Cross-docking is a branch of logistics consisting of unloading materials from an incoming truck and loading these materials directly into outbound trucks with little or no storage in between. Cross-docking operations are conducted to sort materials for different destinations, or to combine materials from different origins into transport vehicles with the same, or similar destination, in generally less than 24 hours time. Timely delivery of products is an important issue in cross docking systems in order to reduce lead times for customer orders, inventory management costs, warehouse space requirements and labor costs. The overall concept of cross-docking operations is represented in Figure 1.

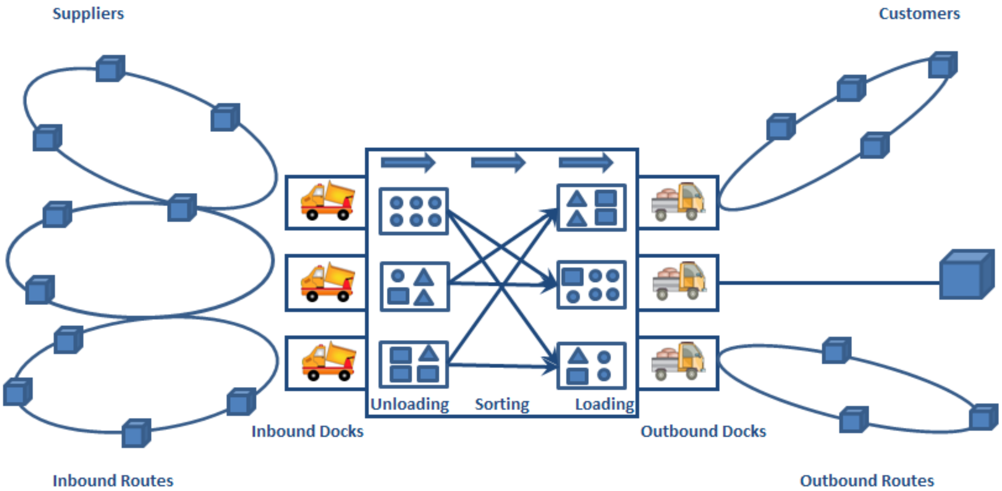

Figure 1. Schematic Representation of cross-dock operations

Materials are collected from suppliers using incoming routes. In order to increase the efficiency of incoming routes vehicles can visit more than one supplier regarding the time and capacity constraints. As soon as a vehicle arrives at the cross-dock, it is scheduled to an available dock for unloading. At this point, synchronization of vehicle arrival times with dock availability is very important to eliminate unnecessary waiting which will result in longer lead times. When the vehicles are unloaded, materials are sorted in accordance with their destination and loaded to the outbound vehicles at the outbound docks. An outbound vehicle can directly go to one customer or can visit more than one customer within its route.

Cross docks act as consolidation points in logistic systems. Based on the suppliers and customers profile, they have different operating characteristics. Our study is motivated by a real life example in Company X, which is a company in the automotive sector in Turkey. In X, the amount of parts required for the daily production is too high to keep inventory within the production area. A cross dock is used to feed the production line of the company directly. The company requires timely delivery of parts to the company from the suppliers through the cross dock. Working in different shifts, the company requires all parts used in that shift to arrive at the production area at the beginning of the shift.

In a three shift example with shifts at [08:00-12:00], [13:00-17:00] and [18:00-22:00], the first, second and third outbound vehicles should leave the cross dock so that they will arrive at the customer no later than 08:00, 13:00 and 18:00, respectively. When the vehicles arrive at the company, they act as a stock area, and the unloaded parts directly go to the production lines.

In correspondence to the outbound routes, incoming routes should be constructed. This time we have restriction on the arrival of parts at the cross dock. The parts used in each shift should arrive at the cross dock no later than a deadline, which is determined from the schedule of the outbound vehicles.

The depicted problem can be summarized as in Figure 2. In area numbered as (1), parts are picked up from the suppliers and delivered to the cross-dock via incoming routes. Regarding the capacity, a vehicle can pick up from a supplier only parts needed at the nearest (in time) shift, or can pick up parts needed in a number of subsequent shifts. In area (2), parts are separated into lanes in the cross dock depending on the shift they are needed. In area (3), the parts are loaded to outbound vehicles and delivered to the company based on known demands in each shift. In area (4), loaded parts are carried to the company via outbound vehicles. And finally, in area (5), parts are unloaded and directly transferred to the production lines.

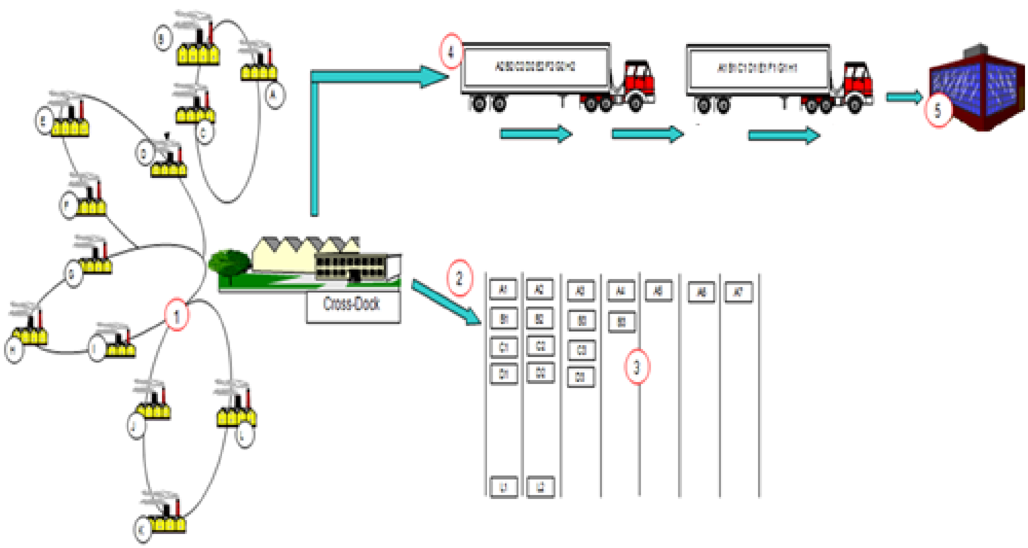

Figure 2. Parts Flow in the Cross-Dock System

Motivated by this application we have defined a vehicle routing problem (VRP). In our problem setting, vehicles are assumed to be capacitated, which makes the problem capacitated VRP, (CVRP). Additionally, as described above, the customer defines time deadlines for shifts in which predetermined amount of products are to be provided. Therefore, in addition to the vehicle capacities, we have shift-based time restrictions, which make the problem capacitated vehicle routing problem with shift-based time constraints (CVRP-STC).

(CVRP-STC) is concerned with the determination of the routes for a fleet of homogeneous capacitated vehicles, based at one depot, to serve a set of suppliers, and there exist shift-based time restrictions for products. At this point, the depot we are dealing with is a cross dock, but, our approach is applicable to any kind of warehouse or production unit. The objective is to minimize the routing costs or total travel time needed to collect all the products.

In (CVRP-STC), the following basic decisions are involved:

- which vehicles visit which suppliers,

- in what order a vehicle visits the suppliers,

- while a vehicle is visiting a supplier, products of which shift are collected.

Regarding the capacity, a vehicle can pick up from a supplier only parts needed at the nearest (in time) shift, or can pick up parts needed in a number of subsequent shifts.

In these basic decisions, a trade-off occurs such that a vehicle can pick up from a supplier parts needed in a number of subsequent shifts, (to minimize the total travel time), however the capacities of the vehicles and the shift-based time constraints could limit the parts collected from that supplier, resulting in multiple visits to the supplier. To the best of our knowledge, the problem (CVRP-STC) has not been defined as such in the vehicle routing or the cross-dock literature.

The rest of the study is organized as follows; In Chapter 2 the literature review on cross docks and also on VRPs is given. In Chapter 3, the details of CVRP-STC are provided and mathematical formulation of the problem and its assumptions are given. In Chapter 4, meta-heuristic solution approaches; Variable Neighborhood Search (VNS) and Simulated Annealing (SA) algorithms are described in detail. Chapter 5 demonstrates our computational experimentation. Finally Chapter 6 presents the conclusions.

CHAPTER 2

LITERATURE REVIEW

In this chapter we provide related literature review on cross docks and vehicle routing. Then, we identify how our problem differs from existing studies in the literature.

2.1 Literature Review on Cross Docks

There are a number of operations within the context of cross-docks that require decision making. Boysen and Fliedner (2009) list the decision problems to be solved during the life cycle of a cross docking terminal from strategic to operational as follows:

- Location of cross docking terminal(s)
- Layout of the terminal
- Assignment of vehicles to dock doors
- Vehicle routing
- Truck scheduling
- Resource scheduling within the terminal

Internal cross-dock operations related decision problems include assignment of trucks to dock doors and truck scheduling. There is a considerable amount of research conducted in this area. Table 1 lists these studies based on the parameters below;

1. Objective Function
    a. Minimize makespan
    b. Minimize earliness and tardiness
    c. Minimize total cost
    d. Minimize the total distance
    e. Maximize the direct flow
2. Operations Inside the Cross-Dock
    a. Considered
    b. Not considered
3. Temporary Storage
    a. Allowed
    b. Not allowed
4. Dock Return Visit
    a. Allowed
    b. Not allowed
5. Truck Change-Over Time
    a. Same for all trucks
    b. Different for different trucks
6. Availability of Trucks at Time Zero
    a. Available
    b. Not available
7. Long Term Storage
    a. Allowed
    b. Not allowed
8. Number of Docks
    a. Single
    b. Multiple
9. Number of Product Types
    a. Single
    b. Multiple

10. Inbound and Outbound Truck Ingredients/Needs
    a. Known as a Priori
    b. Not Known as a Priori
11. Nature of Demand
    a. Deterministic
    b. Stochastic
12. Number of Facilities
    a. Single
    b. Multiple
13. Facility Capacity
    a. Capacitated
    b. Uncapacitated
14. Solution Method
    a. Exact
    b. Heuristic

The majority of the articles listed in Table 1 aim to minimize the makespan. Nearly all of them do not deal in detail with the operations conducted inside the dock; instead, they use a fixed time interval. Most of them assume to have a temporary storage for the products that are unloaded and waiting to be loaded. Dock return visit is not allowed in nearly all of the studies which mean a vehicle is not allowed to return to the dock once it is loaded / unloaded. Truck changeover times are assumed to be known for most of the problems.

Table 1. Literature Review for Cross-Dock Operations

Article	1					2		3		4		5		6		7		8		9		10		11		12		13		14	
	a	b	c	d	e	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b
Yu and Egbelu (2008)	√						√	√			√	√		√			√	√			√	√		√		√			√	√	√
Arabani et al. (2009a)	√						√	√			√	√		√			√	√			√	√		√		√			√		√
Arabani et al. (2009b)		√					√	√			√	√		√			√	√			√	√		√		√			√		√
Sharabiani (2009)	√						√	√			√	√		√			√		√		√	√		√		√			√	√	√
Vahdani et al. (2010)	√						√		√	√		√		√			√	√			√	√		√		√			√		√
Miao et al. (2009)			√				√	√			√		√		√		√		√		√	√		√		√		√		√	√
Boysen (2010)	√	√					√		√		√	√		√			√		√		√	√		√		√			√	√	√
Lee et al. (2006)			√				√		√		√	√		√			√		√		√	√		√		√			√		√
Boysen et al. (2008)	√						√	√			√	√		√			√	√			√	√		√		√			√	√	√

Table 1. Literature Review for Cross-Dock Operations (continued)

Article	1					2		3		4		5		6		7		8		9		10		11		12		13		14	
	a	b	c	d	e	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b
Vahdani and Zandieh (2010)	√						√	√			√	√		√			√	√			√	√		√		√			√		√
Soltani and Sadjadi (2010)	√						√		√	√		√		√			√	√			√	√		√		√			√		√
Sadykov (2009)			√				√	√			√	√			√		√	√			√	√		√		√		√		√	
McWilliams et al. (2005)	√					√			√		√	√		√			√		√		√	√		√		√		√			√
Chen and Song (2009)	√						√		√		√		√	√			√				√	√		√		√				√	√
Chen and Lee (2009)	√						√		√		√		√	√			√	√		√		√		√		√				√	√
Ley and Elfayoumy (2007)	√						√		√		√			√			√		√		√	√		√		√					√
Li et al. (2009)	√						√	√			√		√	√			√		√		√	√		√		√			√	√	√

Table 1. Literature Review for Cross-Dock Operations (continued)

Article	1					2		3		4		5		6		7		8		9		10		11		12		13		14	
	a	b	c	d	e	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b
Li et al. (2004)		√					√	√			√		√				√		√		√	√		√		√			√	√	√
Larbi et al. (2009)			√				√	√		√			√		√		√		√		√	√		√		√			√		√
Song and Chen (2007)	√						√		√		√	√					√		√		√	√		√		√			√	√	√
Lim et al. (2006)				√			√	√			√				√		√		√		√	√		√		√		√		√	√
Briskorn et al. (2009)	√	√					√		√		√						√	√			√	√		√		√					√
Wang and Regan (2008)	√						√		√		√		√		√		√				√	√		√		√			√		√
Ou et al. (2010)			√				√	√			√		√		√		√		√		√	√		√		√			√	√	√

In the majority of the articles, all trucks are assumed to be available at time zero. Long term storage is not allowed in any study since it would conflict with the concept of cross-docking. Number of docks and number of product types are assumed to be multiple in most of them. In all of the studies inbound and outbound truck ingredients/needs are assumed to be known as a priori and nature of demand assumed to be deterministic and number of facilities (cross-dock) assumed to be single. Most of the studies use uncapacitated facilities. Nearly half of the studies use heuristic procedures for the solution of problems since the nature of the operations is NP-Hard. Only one of the studies tries purely exact algorithms and the rest of the articles deal with both exact and heuristic methods.

As stated above, the objective function of most of the studies is minimizing the makespan. Minimizing the makespan in a cross docking system is highly dependent on the availability of the products whenever they are needed. Products are carried to the cross-docks via incoming routes. At this point, for the vehicle scheduling and routing problem it gets important to minimize the makespan. A vehicle route should visit the required suppliers and should arrive at the cross-dock in a timely manner to avoid unnecessary waiting times. Instead of considering the vehicle scheduling aspect of the problem, most of the studies assume that all vehicles are available at time zero. But this assumption will result in larger lead times and higher costs, since the vehicles and the products will wait unnecessarily. Furthermore, all of the studies assume that inbound and outbound truck ingredients/needs are known a priori. To satisfy that assumption, vehicle routes should have been determined.

Liao et al. (2010) work on a model that integrates cross-docking with the vehicle routing problem (VRP). In the study, within the planning horizon, predefined numbers of identical vehicles are used to transport goods from supplies to retailers through a cross-dock.

It is stated that each supplier and retailer can be visited only once while the capacity of the vehicles are respected. The objective of the problem is to determine the number of vehicles and the best route to minimize the sum of the operational and the transportation costs of vehicles. For the solution of the problem, they propose a new tabu search (TS) algorithm. Based on their computational experiments, they express that the proposed TS algorithm can achieve better performance than the existing TS algorithms while using much less computation time.

Dondo et al. (2011) study the multi-echelon VRP with cross docking. They deal with the operational management of hybrid multi-echelon multi-item distribution networks with the objective of satisfying customer demands at minimum total transportation cost. They considered a transportation infrastructure which allows direct shipping or shipping via DC/regional warehouses, including cross-docking or a hybrid strategy that is a combination of both types of shipments. They propose a monolithic optimization framework based on a mixed-integer linear mathematical formulation. They also report computational results for several problem instances.

Lee et al. (2006) work on vehicle routing scheduling where a distribution network with a cross-dock is considered. In their study, an integrated model considering both cross-docking and vehicle routing scheduling is taken into consideration. In the problem setting, split deliveries are not allowed. In order to minimize the transportation costs, the aim of the study is to determine the best routing, number of vehicles required and the arrival time of each vehicle at a cross-dock. For the solution of the problem, firstly they represent a mathematical formulation. Then, since the problem is known to be NP-hard, a tabu search based algorithm is developed. Based on the computational study, they report that proposed algorithm produces good solutions.

2.2 Literature Review on Vehicle Routing

Laporte (1992) describes the VRP as the problem of designing optimal delivery or collection routes from one or several depots to a number of geographically scattered cities or customers, subject to side constraints and states that the VRP plays a central role in the fields of physical distribution and logistics.

Marinakis and Migdalas (2007) express that the vehicle routing problem and its variants have very important applications in the area of distribution management, as a consequence they have become some of the most studied problems in the combinatorial optimization and a large number of papers dealing with the numerous procedures have been proposed to solve them. In their study they provide a bibliography in VRP.

Desrochers et al. (1990) explain the interest on developing optimization and approximation algorithms for vehicle routing problems as the practical importance of effective and efficient methods for handling physical distribution situations. They express that it is very hard for a distribution manager regardless of his experience to decide on a method that is well suited for his specific situation because of the large number of existing algorithms. Therefore, in order to facilitate this decision process they provide a classification system to support modeling problem situations and suggesting algorithms.

Ekşioğlu et al. (2009) present a taxonomic framework for defining and integrating the domain of the extant VRP literature in terms that are operationally meaningful. They review the existing classifications and the growth in vehicle routing literature, after that they state the need for a taxonomy and represent their taxonomy for VRP. They use five main headings, and divide them in to small characteristics as given in Table 2.

Table 2. Taxonomy of the VRP Literature (Ekşioğlu et al. (2009))

1. Type of Study
    - 1.1. Theory
    - 1.2. Applied methods
        - 1.2.1. Exact methods
        - 1.2.2. Heuristics
        - 1.2.3. Simulation
        - 1.2.4. Real time solution methods
    - 1.3. Implementation documented
    - 1.4. Survey, review or meta-research
2. Scenario Characteristics
    - 2.1. Number of stops on route
        - 2.1.1. Known (deterministic)
        - 2.1.2. Partially known, partially probabilistic
    - 2.2. Load splitting constraint
        - 2.2.1. Splitting alowed
        - 2.2.2. Splitting not alowed
    - 2.3. Customer service demand quantity
        - 2.3.1. Deterministic
        - 2.3.2. Stochastic
        - 2.3.3. Unknown
    - 2.4. Request times of new customers
        - 2.4.1. Deterministic
        - 2.4.2. Stochastic
        - 2.4.3. Unknown
    - 2.5. On site service/waiting times
        - 2.5.1. Deterministic
        - 2.5.2. Time dependent
        - 2.5.3. Vehicle type dependent
        - 2.5.4. Stochastic
        - 2.5.5. Unknown
    - 2.6. Time window structure
        - 2.6.1. Soft time windows
        - 2.6.2. Strict time windows
        - 2.6.3. Mix of both
    - 2.7. Time horizon
        - 2.7.1. Single period
        - 2.7.2. Multi period
    - 2.8. Backhauls
        - 2.8.1. Nodes request simultaneous pick ups and deliveries
        - 2.8.2. Nodes request either linehaul or backhaul service, but not both
    - 2.9. Node/Arc covering constraints
        - 2.9.1. Precedence and coupling constraints
        - 2.9.2. Subset covering constraints
        - 2.9.3. Recourse allowed
3. Problem Physical Characteristics
    - 3.1. Transportation network design
        - 3.1.1. Directed network
        - 3.1.2. Undirected network
    - 3.2. Location of addresses (customers)
        - 3.2.1. Customers on nodes
        - 3.2.2. Arc routing instances
    - 3.3. Geographical location of customers
        - 3.3.1. Urban (scattered with a pattern)
        - 3.3.2. Rural (randomly scattered)
        - 3.3.3. Mixed
    - 3.4. Number of points of origin
        - 3.4.1. Single origin
        - 3.4.2. Multiple origins
    - 3.5. Number of points of loading/unloading facilities (depot)
        - 3.5.1. Single depot
        - 3.5.2. Multiple depots
    - 3.6. Time window type
        - 3.6.1. Restriction on customers
        - 3.6.2. Restriction on roads
        - 3.6.3. Restriction on depot/hubs
        - 3.6.4. Restriction on drivers/vehicle
    - 3.7. Number of vehicles
        - 3.7.1. Exactly n vehicles *(TSP in this segment)*
        - 3.7.2. Up to n vehicles
        - 3.7.3. Unlimited number of vehicles
    - 3.8. Capacity consideration
        - 3.8.1. Capacitated vehicles
        - 3.8.2. Uncapacitated vehicles
    - 3.9. Vehicle homogeneity (Capacity)
        - 3.9.1. Similar vehicles
        - 3.9.2. Load-specific vehicles
        - 3.9.3. Heterogeneous vehicles
        - 3.9.4. Customer-specific vehicles
    - 3.10. Travel time
        - 3.10.1. Deterministic
        - 3.10.2. Function dependent (a function of current time)
        - 3.10.3. Stochastic
        - 3.10.4. Unknown
    - 3.11. Transportation Cost
        - 3.11.1. Travel time dependent
        - 3.11.2. Distance dependent
        - 3.11.3. Vehicle dependent
        - 3.11.4. Operation dependent
        - 3.11.5. Function of lateness
        - 3.11.6. Implied hazard/risk related
4. Information Characteristics
    - 4.1. Evaluation of information
        - 4.1.1. Static
        - 4.1.2. Partially dynamic
    - 4.2. Quality of information
        - 4.2.1. Known (deterministic)
        - 4.2.2. Stochastic
        - 4.2.3. Forecast
        - 4.2.4. Unknown (Real-time)
    - 4.3. Availability of information
        - 4.3.1. Local
        - 4.3.2. Global
    - 4.4. Processing of information
        - 4.4.1. Centralized
        - 4.4.2. Decentralized
5. Data Characteristics
    - 5.1. Data Used
        - 5.1.1. Real world data
        - 5.1.2. Synthetic data
        - 5.1.3. Both real and synthetic data
    - 5.2. No data used

Based on this classification scheme, they also provide an extensive literature review in their study.

In the literature, there are multiple variants of VRP that researchers work on. Main variants of vehicle routing problems can be listed as follows.

- Capacitated Vehicle Routing Problems (CVRP)
- Multi–depot VRP (MDVRP)
- VRP with Time Windows (VRPTW)
- Stochastic VRP (SVRP)
- Periodic VRP
- Split Delivery VRP
- VRP with Backhauls (VRPB)
- VRP with Pick-Ups and Deliveries (VRPPD)
- VRP with Compartments

In that follows we provide a review on a sample of studies on VRP and the extended problems, since there exist a vast number of such studies.

Ekşioğlu et al. (2009) state that first incorporation of more than one vehicle in the problem formulation which can be considered as being first in the VRP literature was in 1964. After that mainly based on the improvements in computer technology VRP research accelerated during the 1990s. They express that the literature growth is almost perfectly exponential with a 6.09% annual growth rate which demonstrates VRP's vitality. At this point, we try to give a survey on progressive VRP literature based on the described VRP variants.

Lin et al. (2009) deal with CVRP. In their problem setting, each vehicle has the same capacity and starts from the same delivery depot and then routes through customers. The loading and traveling distance of each vehicle cannot exceed the loading capacity and the maximum traveling distance of vehicle where the objective is to minimize the traveling cost. To solve the problem they used a hybrid algorithm of simulated annealing and tabu search. They run simulations and according to the simulation results they report that the proposed algorithm is competitive with other existing algorithms for solving CVRP.

Toth and Vigo (2002) study on the branch and bound algorithms for the CVRP with both symmetric and asymmetric cost matrices. In their study they present the performance of different relaxations and algorithms on a set of benchmark instances through the comparison of computational results.

Ho et al. (2008) focus on the multi-depot VRP. In their problem setting, the number and locations of the depots and customers are predetermined. Each depot is large enough to store all the products ordered by the customers and the demand of each customer known a priori. Vehicles with limited capacity are used to transport the products from depots to customers and each vehicle starts and finishes at the same depot visiting each customer exactly once. They stated that in multi-depot VRP problems, three decision areas exist. First is the grouping problem, clustering the set of customers to be served by the same depot. Second is the routing problem, assigning customer groups to routes. And the last one is the scheduling problem where the sequences of routes are determined. For the solution of the problem, they propose two hybrid genetic algorithms. In the first one, the initial solution is generated randomly, and in the second one the Clarke and Wright savings method and the nearest neighbor heuristic are incorporated for the initialization procedure. They conduct a computational study with different problem sizes and report that the performance of the second algorithm is better regarding the total delivery time.

Hong (2012) studies the dynamic vehicle routing problem with hard time windows. For the solution, they decompose the problem to a series of static subsets of VRP with time windows. In order to solve the decomposition problem, they propose an event-trigger mechanism through which they obtain a series of system delay snapshots that is regarded as static VRP with time windows. The new request arrival during the stable operation is taken in to account as the trigger event. They propose an improved large neighborhood search (LNS) algorithm to solve the static problem. They work on test problems from Solomons static benchmarks and Lacker's dynamic data sets and based on the computational results they state that their method is superior to other methods in most instances.

Hashimoto et al. (2008) work on the VRP with time windows, VRPTW, where traveling times and traveling costs are time-dependent. In order to determine the routes of the vehicles, they use local search. They incorporate dynamic programming in the local search algorithm to compute an optimal time schedule for each route in neighborhood solution. For the solution of the problem, they develop an iterated local search algorithm. They express the effectiveness of the proposed generalization based on the computational results of the iterated local search algorithm compared against existing methods.

Li et al. (2010) focus on stochastic vehicle routing problems. In their problem setting travel and service times are stochastic, and each customer has a specific time window constraint. They state that the problem they are dealing with is originally formulated as a chance constrained programming model and a stochastic programming model with recourse in terms of different optimization criteria. For the solution of the problem, they use tabu search algorithm. They also report their computational test results.

Novoa and Storer (2009) work on dynamic programming algorithms for the single-vehicle routing problem with stochastic demands. In the problem setting, a single vehicle with fixed capacity departs from a depot to perform only deliveries (or only pick-ups) at different customer locations. For the solution of the problem, a two-step look ahead rollout started with a stochastic base sequence is utilized which is relatively better than the one-step rollout algorithm started with a deterministic sequence. In the study, they also consider computing the cost-to-go with Monte Carlo simulation which reduces the computation time pretty much with little or no loss in solution quality. They also report computational results for sampled vehicle routing problems.

Hemmelmayr et al. (2009) study the periodic vehicle routing problem without time windows. In the study planning horizon of several days is taken into account where each customer requires a certain number of visits. There also exists some flexibility on the exact days of the visits which corresponds to a decision about the choice of the visit days for each customer and to solve a VRP for each day. For the solution, they use a variable neighborhood search algorithm. Based on the computational results, they state that their approach is competitive and outperforms existing solution procedures proposed in the literature. They also indicate that if only a single vehicle is considered which will result in periodic traveling salesman problem, their VNS procedure approach is again competitive with slight changes.

Francis and Smilowitz (2006) work on a continuous approximation model for the period vehicle routing problem where service choices exist. In that problem, the visit frequency to nodes is a decision of the model which corresponds to more efficient vehicle tours. For problem resolution, they use a continuous approximation model to facilitate strategic and tactical planning of periodic distribution systems while evaluating the value of service choice.

They examine their approach using a test instance from the literature. They also state that results of the proposed model can both help distribution service providers design valuable service options and be used to guide discrete solutions in exact vehicle routes determination.

Aleman et al. (2010) study the split delivery VRP. In their study, they provided a detailed literature survey. For the solution of the split deliver VRP with the minimum fleet size, they present three local heuristic search algorithms. First, they represent a new constructive algorithm based on a novel concept called the route angle control measure. Second, via adaptive memory concepts, they enhance this constructive approach to an iterative approach. As per the third one, they include a variable neighborhood descent process. Using benchmark problem sets, they evaluate their approaches against exact and heuristic approaches.

Gulczynski et al. (2010) work on split delivery VRP. They state in their study that although split delivery philosophy is cost effective considering travel costs, it is a disadvantage for a customer who prefers single visit. At that point, in their problem setting they allow split deliveries only if a minimum fraction of a customer’s demand is serviced by a vehicle which is a new problem called the split delivery VRP with minimum delivery amounts. For the solution of the problem they present a heuristic method that combines an endpoint mixed integer program with an enhanced record-to-record travel algorithm. They report computational results on a wide range of problem sets and state that their approach is competitive with the best heuristics in the literature. They also produce a set of 21 new test problems with minimum delivery amounts to be used in future studies as benchmarks.

Zhong and Cole (2005) study on VRP with backhauls and time windows. In their study, they take into account the case of existence of customer precedence which requires that all line-haul customers be visited before any backhaul customer. Their solution approach is two phased. In the first phase, proposed heuristic uses guided local search to improve routes allowing time and capacity violations. In the second phase, a new technique, called section planning, takes place, which eliminates time and capacity violations by assigning problematic customers to newly created routes. They report their computational results stating that the proposed heuristic produces better results than the best solutions in the literature.

Wade and Salhi (2002) work on a vehicle routing problem with backhauls. In their problem setting, they state that there is no need to visit all line-haul customers before serving backhaul customers which is new to the literature. In order to solve the problem, they use an insertion-type heuristic. Based on the relaxation in the restriction of the mix of line-haul and backhaul customers, improvements in route costs are obtained.

Deng et al. (2009) study on VRP with pick-up and delivery. In their study, they take into account soft time windows, fixed vehicle costs and a coefficient for vehicle full-load. For the solution of the problem, they use simulated annealing algorithm. In order to increase the efficiency of the algorithm, they improve it such that the proposed approach searches for a larger solution space within a certain period of time. While that search, to speed up the algorithm, a memory device is included to ensure that the output result at the end of algorithm is the optimal solution for that run. Additionally, as per the termination criteria of the algorithm, they present a new mixed-termination rule, that is, the algorithm terminates when the temperature is below a predefined value, or the memory array is same with no changes after a certain number of steps. Finally, they test the algorithm and report that the proposed approach is stable and efficient.

Zachariadis et al. (2009) work on VRP with simultaneous pick-up and delivery, i.e. transported products are bi-directionally, from the central depot to the customers, and from the customers back to the central depot. It is aimed to satisfy delivery and pick-up demand of the customer population in the problem. In order to solve the problem, they propose a hybrid solution approach consisting tabu search and guided local search. In that approach, to keep a balance between the intensification and diversification, the algorithm searches vast areas of the solution space mainly the most promising portions. Based on the computational experiments ranging from 50 to 400 customers, it is stated that the proposed approach is capable of producing high quality solutions.

Derigs et al. (2010) study on the VRP with compartments, which is a problem relevant to several industries that requires inhomogeneous distribution of products in the food and petrol sector. For the solution of the problem, they prefer to work with different heuristic components. Their approach consists of alternative approaches for construction like greedy insertion, sweep and savings; for local search like 2-opt, Or-opt, 2-opt* and for large neighborhood search like order and vehicle based removal operators, greedy and regret based insertion and in addition diverse meta-heuristics like simulated annealing, record-to-record travel and tabu search. In their study, they also provide a benchmark suite of 200 instances.

Fallahi et al. (2008), worked on multi-compartment VRP. In their problem setting, vehicles are identical with several compartments each dedicated to one product. For each customer the demand for a single product must be entirely delivered by one single vehicle, whereas the demand for different products allowed to be delivered by several vehicles. In order to solve the problem, they propose three algorithms. First a constructed heuristic, second a memetic algorithm (MA) with a path relinking method and third a tabu search (TS) algorithm. It is stated that TS provides slightly better solutions than MA.

2.3 Comparison

Comparing our problem with the variants of VRP, the following observations can be made. First of all we have capacity constraints for vehicles, which make our problem a CVRP. We have only one cross-dock which makes our problem a single-depot problem. We define shift-based time constraints which makes our problem different from the VRPTW. Best to our knowledge, shift-based time constraints for the products is new to the literature. We have a single period, a day, in our problem. Our problem shows some similarities with the split delivery VRP, which removes the restriction that each supplier has to be visited exactly once. But the reason behind multiple visits to a supplier is not just exceeding vehicle capacity with the amount of supply, but also meeting shift-based time restrictions for the products. There are no backhauls in our problem and only pick-ups from suppliers and delivery to the cross dock take place. In VRP with compartments, transportation of inhomogeneous products is allowed in different compartments. In that problem setting a vehicle arrived at a supplier can take only a specific type of product which is similar to our problem. The difference here is that; in vehicle routing problems with compartments, products are assigned to compartments based on their characteristics and there exist no time restrictions Moreover, each compartment may have different capacities. Additionally, while some products cannot be carried together in a vehicle in VRP with compartments setting, we have no such restriction on products.

As stated in literature review part, Ekşioğlu et al. (2009) described a taxonomy related with VRP. Based on their taxonomy, our problem is applied methods as we provide exact and meta-heuristic solution approaches to our problem. Scenario characteristics of our problem are; multi visits to a supplier is allowed, the parameters are deterministic, strict time windows for products are defined, single period is assumed and no backhauls are defined.

Regarding the physical characteristics, transportation network is undirected, suppliers are on nodes, geographical locations of suppliers are rural, and only a single depot is defined. We have homogeneous capacitated vehicles and transportation costs are vehicle independent. Regarding the information characteristics; evolution of information is static, quality of information is deterministic, availability of information is global and processing of information is centralized. Finally, data used is synthetic data.

CHAPTER 3

MATHEMATICAL MODELLING

In this chapter we first present the assumptions of our problem, then provide a mathematical model for (CVRP-STC).

3.1 Assumptions

Assumptions of our problem, CVRP-STC, are listed below;

- ✓ Supplier and the cross dock locations are known.
- ✓ The matrix of shortest distances is assumed to be known and all travel occurs on the shortest paths at constant speeds.
- ✓ A supplier can provide different type of products.
- ✓ Product supplies are assumed to be fixed and known.
- ✓ At the supplier, the products are available at the beginning of the planning period.
- ✓ There are *m* different shifts, thus *m* different deadlines for the products.
- ✓ The lengths of the shifts are assumed to be equal.
- ✓ The batch of product, which is prepared for a specific shift, cannot be split.
- ✓ The inbound vehicle fleet is composed by a limited number of homogeneous capacitated vehicles.

- ✓ Each vehicle is assumed to start its route from the cross dock and return to that point after collecting required products from the suppliers on its route.

- ✓ Tour length of each vehicle is assumed to be less than or equal to the shift length, that is each vehicle is assumed to finish its route within one shift interval.

- ✓ Upon finishing a tour, a vehicle can start its subsequent tour. A vehicle can have at most *m* tours.

- ✓ The fixed costs of the vehicles are not considered.

3.2 Sets, Parameters and Decision Variables

Sets used in the problem are as follows;

N': Set of all nodes (Suppliers + Cross Dock)

N: Set of suppliers

P_i: Set of products supplied by supplier i, $i \in N$

$P = \bigcup_{i \in N} P_i$

K: Set of shifts

V_k: Set of vehicles used in the k^{th} shift, $k \in K$.

$V = \bigcup_{k=1,\dots,m} V_k$

Parameters required to handle the problem are as follows;

C : Capacity of vehicles.

m: Number of shifts.

TI_{ij} : Travel time between nodes i and j, which includes the service time at node i, $i,j \in N$.

TL_k : Latest time that cross dock can receive supplies of product batches for shift k, $k \in K$.

TM = Maximum $_{k \in K}$ $\{TL_k\}$

a_{ipk}: Vehicle capacity usage of the batch of product p for shift k from supplier i, $i \in N, p \in P_i, k \in K$.

s: Constant number such as $s \in \{1,2 \ldots ,m\text{-}1\}$.

n: Number of vehicles.

Decision variables used in the problem are as follows;

tp_{ipk} : Arrival time of the batch of product p for shift k from supplier i to the cross dock, $i \in N, p \in P_i, k \in K$.

tv_{iv} : Arrival time of vehicle v to node i, $i \in N, v \in V$.

b_{ijv} : 1, if vehicle v travels from node i to j;
0 otherwise, $i,j \in N, v \in V$.

x_{ipkv} : 1, if the batch of product p for shift k from supplier i is delivered by vehicle v; 0 otherwise, $i \in N, p \in P_i, k \in K, v \in V$.

u_{iv}: Order of node i for the route of vehicle v, $v \in V$.

r_v: Ready time of vehicle v at the cross dock, $v \in V$.

3.3 Mathematical Model

Mathematical model developed to solve the problem of capacitated vehicle routing with shift-based time constraints is given below;

Min $\sum_{i \in N'} \sum_{j \in N'} \sum_{v \in V} TI_{ij}\, b_{ijv}$

subject to

$\sum_{j \in N} b_{0jv} \leq 1$ $\quad v \in V$ (1)

$\sum_{i \in N'} b_{ilv} = \sum_{j \in N'} b_{ljv}$ $\quad v \in V,\ l \in N'$ (2)

$u_{0v} = 1$ $\quad v \in V$ (3)

$u_{iv} - u_{jv} + 1 \leq |N|(1 - b_{ijv})$ $\quad v \in V,\ i \in N', j \in N$ (4)

$\sum_{i \in N} \sum_{p \in Pi} \sum_{k \in K} a_{ipk} x_{ipkv} \leq C$ $\quad v \in V$ (5)

$\sum_{v \in V} x_{ipkv} = 1$ $\quad i \in N,\ p \in P_i,\ k \in K$ (6)

$\sum_{k \in K} \sum_{p \in Pi} x_{jpkv} \leq m \times |N| \times |P| \times \sum_{i \in N'} b_{ijv}$ $\quad v \in V,\ j \in N$ (7)

$\sum_{j \in N} b_{0jv+(s-1)n} \geq \sum_{j \in N} b_{0jv+sn}$ $\quad v \in V_1,\ s \in \{1,2 \ldots ,m-1\}$ (8)

$tp_{ipk} \leq TL_k$ $\quad i \in N,\ p \in P_i,\ k \in K$ (9)

$r_v = 0$ $\quad v \in V_1$ (10)

$r_{v+sn} - tv_{0v+(s-1)n} \leq TM(1 - \sum_{j \in N} b_{0jv+(s-1)n})$ $\quad v \in V_1, s \in \{1,2 \ldots ,m-1\}$ (11)

$tv_{0v+(s-1)n} - tv_{0v+sn} \leq TM(1 - \sum_{j \in N} b_{0jv+sn})$ $\quad v \in V_1,\ s \in \{1,2 \ldots ,m-1\}$ (12)

$r_{v+sn} \geq TM(1 - tv_{0v+(s-1)n})$ $\quad v \in V_1,\ s \in \{1,2 \ldots ,m-1\}$ (13)

$tv_{iv} + TI_{ij}b_{ijv} - TM(1 - b_{ijv}) \leq tv_{jv}$ $\quad v \in V,\ i,j \in N,\ i \neq j$ (14)

$tv_{iv} + TI_{i0}b_{i0v} - TM(1 - b_{i0v}) \leq tv_{0v}$ $\quad v \in V,\ i \in N$ (15)

$r_v + TI_{0j}b_{0jv} - TM(1 - b_{0jv}) \leq tv_{jv}$ $\quad v \in V,\ j \in N$ (16)

$TI_{0j}b_{0jv} \leq tv_{jv}$ $\quad v \in V,\ j \in N$ (17)

$tv_{0v} - TM(1 - x_{ipkv}) \leq tp_{ipk}$ $\quad i \in N,\ p \in P_i,\ k \in K,\ v \in V.$ (18)

$tp_{ipk} - TM(1 - x_{ipkv}) \leq tv_{0v}$ $\quad i \in N,\ p \in P_i,\ k \in K,\ v \in V$ (19)

$b_{ijv} \in \{0,1\}$ $\quad v \in V,\ i,j \in N'$ (20)

$x_{ipkv} \in \{0,1\}$ $\quad i \in N,\ p \in P_i,\ k \in K,\ v \in V$ (21)

$u_{iv} \geq 0$ $\quad i \in N',\ v \in V$ (22)

$tp_{ipk} \geq 0$ $\quad i \in N,\ p \in P_i,\ k \in K$ (23)

$tv_{iv} \geq 0$ $\quad i \in N',\ v \in V$ (24)

$r_v \geq 0$ $\quad v \in V$ (25)

In the objective function, it is aimed to minimize the routing times. Constraints (1) indicate that each vehicle can go to at most one supplier from the cross-dock. Constraints (2) satisfy that if a vehicle arrives at a supplier, it must also leave it. Constraints (3) state that each vehicle starts its route from the cross-dock. Constraints (4) are the MTZ constraints (Miller, Tucker and Zemlin (1960)) used to eliminate subtours. Constraints (5) are related with vehicle capacities. Constraints (6) ensure that all the batches of all the products supplied by the suppliers must be collected; it also indicates that each batch of each product is assigned to one vehicle. Constraints (7) state that if any of the product batches supplied by the supplier is assigned to a vehicle, this vehicle must visit that supplier. Constraints (8) ensure that if a vehicle used for the s^{th} interval, then it must also have been used for the 1,…,*s*-1 intervals.

Following sets of constraints are the time related constraints. Constraints (9) state that arrival time of the batch of product *p* for shift *k* from supplier *i* at the cross dock must not exceed the latest time that cross dock can receive supplies of products for shift *k*. Constraints (10) ensure that vehicles used for the first time are ready at cross dock at time zero. Constraints (11) state that ready time of a vehicle for a usage is equal to arrival time of the vehicle to the cross dock from the previous usage. Constraints (12) indicate that arrival time of a vehicle to the cross dock for the s^{th} interval, is greater than the arrival time of the vehicle to the cross dock for the previous interval. Constraints (13) ensure that ready time of a vehicle at the cross dock for the s^{th} interval is linked to the arrival time of the vehicle to the cross dock from the $(s\text{-}1)^{th}$ interval. Constraints (14) – (15) express that if a vehicle *v* travels from node *i* to node *j*, than arrival time to node *j* will be the sum of arrival time of the vehicle *v* to node *i* and travel time between nodes *i* and *j*, including the service time at node *i*. Constraints (16) state that if a vehicle *v* travels from cross dock to node *j*, than arrival time to node *j* will be the sum of ready time of the vehicle *v* at the cross dock and travel time between cross dock and node *j*. Similarly, constraints (17) state the arrival time of vehicle *v* to the cross dock, if the vehicle *v* is used. Constraints (18) – (19) set the relationship between the arrival times of vehicles and product batches to the cross dock. Finally, the remaining constraints numbered (20) - (25) define the decision variables.

The VRP is proved to be NP-hard and we additionally have shift-based time constraints, which make it harder for a mathematical model to find optimal solutions in reasonable time frames. Using proposed mathematical model for the solution of a problem with ten nodes, three products from each supplier, two vehicles and three shifts, after a ten hours solution time, it is seen that the relative gap between the best possible solution and the obtained solution is about 30%. This result also demonstrates the complexity of the proposed mathematical model. Related to that issue, Marinakis and Migdalas (2007) mentioned that VRP is an NP-hard problem; therefore it is not expected and is in general believed impossible to develop exact solution methods that can solve a VRP instance in reasonable amount of running time. At this point, for the solution of the described problem, we developed meta-heuristic approaches based on variable neighborhood search and simulated annealing in order to obtain high quality solutions in reasonable time frames.

CHAPTER 4

META-HEURISTIC SOLUTION APPROACHES

Since it is not possible to use developed mathematical model to obtain solutions for (CVRP-STC) for most real-world problems in reasonable times, we propose meta-heuristic approaches based on variable neighborhood search and simulated annealing to obtain good solutions fast. Proposed meta-heuristic approaches will be described in detail in the following sections.

4.1 Variable Neighborhood Search

Variable neighborhood search (VNS) is a recently developed meta-heuristic which produces promising solutions to vehicle routing problems. Hansen and Miladovic (2001) described the VNS as a simple and effective meta-heuristic for combinatorial and global optimization, which takes advantage of systematic change of neighborhood within a possibly randomized local search algorithm. Zhao et al. (2008) and Polacek et al. (2004) stated that VNS produces better solutions in vehicle routing problems they have dealt with, comparing with the existing proposed meta-heuristics.

VNS as a basic idea while searching for a better solution, changes the neighborhoods in the search. A descent method is used in VNS in order to move to a local minimum and then either systematically or at random, it precedes increasingly distant neighborhoods of this solution. Whenever the algorithm moves to a new neighborhood, one or several points within that neighborhood are used as starting points for a local descent.

Change of the current solution depends on the improvement that means the value of the current solution changes to a new one if and only if a better solution has been found. Different from the meta-heuristics like Simulated Annealing or Tabu Search, VNS is not a trajectory following method and does not specify forbidden moves.

Kytöjoki et al. (2007) studied on CVRP with the objective of designing minimum cost routes for a fleet of homogeneous capacitated vehicles in order to service geographically scattered customers with known demands. For the solution of the problem they proposed an efficient two-phase variable neighborhood search heuristic. In the first phase, an initial solution is created with a hybrid cheapest insertion heuristic applying seven improvement heuristics according to a variable neighborhood search scheme. In the second phase, an attempt is made to improve the initial solution with the same VNS approach, but using a different strategy. Based on the computational experiments, they stated that their proposed algorithm produces high-quality solutions within reasonable CPU times.

Imran et al. (2009) worked on the heterogeneous fleet VRP. For the solution of the problem, they proposed an adaptation of the basic VNS algorithm. In their approach, the initial solution is obtained by Dijkstra's algorithm based on a cost network constructed by the sweep algorithm and the 2-opt. They also used a diversification procedure with a number of local search methods. In order to solve the problem, they proposed two VNS variants, which differ in the order the diversification and Dijkstra's algorithm. They stated that both of the approaches are competitive.

Hansen and Miladovic (2001) quite extensively worked on VNS. They stated the basic rules, principles and applications of the algorithm in detail. They also described the basic steps of VNS as in Figure 3.

Initialization

(1) Select the set of neighborhood structures N_k, for k=1,...,k_{max}

(2) Find an initial solution x;

(3) Choose a stopping condition;

Repeat the following until stopping condition is met.

Main Step:

(1) Set $k \leftarrow 1$;

(2) Until $k=k_{max}$, Repeat the following steps:

(a) *Shaking*: Generate a point x' at random from the k^{th} neighborhood of x ($x' \in N_k(x)$);

(b) *Local Search*: Apply local search method with x' as initial solution;

Denote x'' as the obtained local optimum

(c) *Move or not*: If this local optimum is better than the incumbent, move there ($x \leftarrow x''$) with $k \leftarrow 1$; Otherwise, set $k \leftarrow k+1$.

Figure 3. Basic Steps of VNS

In the initialization part, the neighborhood structure for k is determined which will affect the diversification of the algorithm. And then an initial solution is generated, x, so as to produce promising solutions. The quality of the initial solution is very critical for high quality solutions. With better start points, it is more possible for an algorithm to obtain better results. Finally, in the initialization step, the stopping condition of the algorithm is determined.

In the main step of the algorithm, firstly, the value of k is set to the smallest value namely k is set to 1. And then, until the algorithm reaches to the predefined k_{max} value, a number of steps executed in order to improve the results of the problems. Within this concept, the first step is shaking, according to the value of k, a new solution is produced, x', from the k^{th} neighborhood of x. Then taking the new generated solution x' as the initial solution local search algorithm applied in order to find the local minimum. The steps of local search can be explained as in Figure 4.

Set $x'' \leftarrow x'$;
improvement← yes;
Until improvement= no repeat the following;
Calculate objective function value, $f(x''')$, for all x''' from the neighborhood of x'' ($x''' \in N_1(x'')$);
Set $x''' \leftarrow x'''$ with smallest objective function;
If $f(x'') > f(x''')$
Set $x'' \leftarrow x'''$;
If $f(x'') \leq f(x''')$
improvement← no;
Denote x'' as the obtained local optimum

Figure 4. Steps of Local Search Algorithm

After local search step, the result obtained from the local search is compared with the incumbent solution at hand. For the minimization problem if the local search result is smaller than the incumbent, the value of k is set to 1, and algorithm goes to step 1.

Otherwise the value of the *k* is increased with the predefined step value, in that case the algorithm continues until *k* is equals to the defined k_{max} value.

4.2 Simulated Annealing

Simulated annealing was firstly introduced by Kirkpatrick et al. (1983), for the solution of combinatorial optimization problems. Tang (2004) states that the idea of simulated annealing algorithm is motivated by the annealing process in solids. In order to accomplish this situation of growing silicon in the form of highly ordered, defect-free crystals, the material is annealed. Firstly, it is heated to a temperature that permits many molecules to move freely with respect to each other, and then it is cooled carefully, slowly, until the material freezes into a crystal, which is completely ordered, and thus the system is at the state of minimum energy. In order to optimize the objective function, simulated annealing techniques prefer an analogous cooling operation for transforming a poor, unordered solution into an ordered, desirable solution. The cooling schedule is a very important key point for the annealing process. In order to achieve a low energy state, slow cooling is a must. In gradient-based minimization algorithms such as the Newton–Raphson algorithm, only downhill moves are allowed where downhill moves realized as fast as possible. The advantage of the annealing algorithm is that, it takes not only downhill moves, but also permits uphill moves with an assigned probability depending on the temperature of the current state. Çakır et al. (2011) states that using a probability function, simulated annealing is able to escape local optima. The probability of accepting uphill move is made via Boltzman distribution $e^{-\Delta E/Tc}$; where ΔE indicates the change of energy and T_c corresponds to the current temperature. Tang (2004) also states that other than the probability, an annealing algorithm includes the four main components as described below.

The first component is the configuration which corresponds to the possible problem solutions that the search for optimal value conducted. The decision variables in general are multidimensional, discrete and have upper and lower bounds. The second is the cost function which is an objective function to measure how well the system performs within a certain configuration. The third one is the move set which is a generator of random changes, namely the neighbors in the configuration. And finally, the fourth one is the cooling schedule that is the cooling speed to anneal the problem from a random solution to a good, frozen one. Within the cooling schedule, a starting temperature, a stopping temperature together with the rules to determine when and how much the temperature should be reduced should be provided.

Lin et al. (2011) studied on the truck and trailer routing problem with time windows. In their study, an extension of the truck and trailer routing problem, that is the truck and trailer routing problem with time windows is described, in order to bring this type of routing problem closer to the reality. For the solution of the problem a simulated annealing algorithm is proposed. They performed two computational experiments, one with a set of six Solomon's benchmark problems, and the other, 54 instances converted from Solomon's benchmark problems. Based on the solution results, they stated that the proposed simulated annealing heuristic is a producing promising solution.

Tavakkoli-Moghaddam et al. (2007), worked on a capacitated vehicle routing problem. In their problem setting, split services are allowed by assuming heterogeneous fixed fleet. They also assumed that the cost of the fleet linked to the number of vehicles used and total unused capacity. They firstly proposed a mix integer linear model. Then, they generated a simulated annealing method. Based on the computational experiments, they stated that the proposed algorithm is capable of producing high quality solutions.

4.3 Proposed Variable Neighborhood Search Algorithm

In order to customize the variable neighborhood approach for CVRP-STC, we utilized the roadmap described by Hansen and Miladovic (2001).

We have defined two neighborhood structures of a solution $\boldsymbol{x}$**.** In the first neighborhood structure, the k^{th} neighborhood of a solution $\boldsymbol{x}$**,** $N_k^1(\boldsymbol{x})$, is composed of all solutions that differ from $\boldsymbol{x}$ in routes in terms of the positions of k suppliers**.** In the second neighborhood structure, the k^{th} neighborhood of a solution $\boldsymbol{x}$**,** $N_k^2(\boldsymbol{x})$, is composed of all solutions that differ from $\boldsymbol{x}$ in terms of the shifts of products collected from the suppliers.

In the initialization step, neighborhood structures $N_k^i(\boldsymbol{x})$ for i = 1,2 and k=1,...,$kmax$ are defined. The initial solutions are obtained by assuming that the vehicles first collect only the products of the first shift, and this is repeated for all the shifts. Thus the initial solution is the solution of the CVRP problem that repeats m times. At this point, for the solution of the CVRP problem, the best solutions reported for the literature examples are utilized so as to obtain initial solutions and it is assumed that the routes are same for each shift at the beginning which results in three visits for all suppliers within the day. The stopping condition is also determined via parameter setting and described in detail in following sections.

In the shaking part, a neighborhood structure $N_k^1(\boldsymbol{x})$ or $N_k^2(\boldsymbol{x})$ is selected randomly. For $N_k^1(\boldsymbol{x})$, the algorithm randomly selects two suppliers from different routes and changes them with each other and that repeats k times. Schematic representation of this shaking is given in Figure 5.

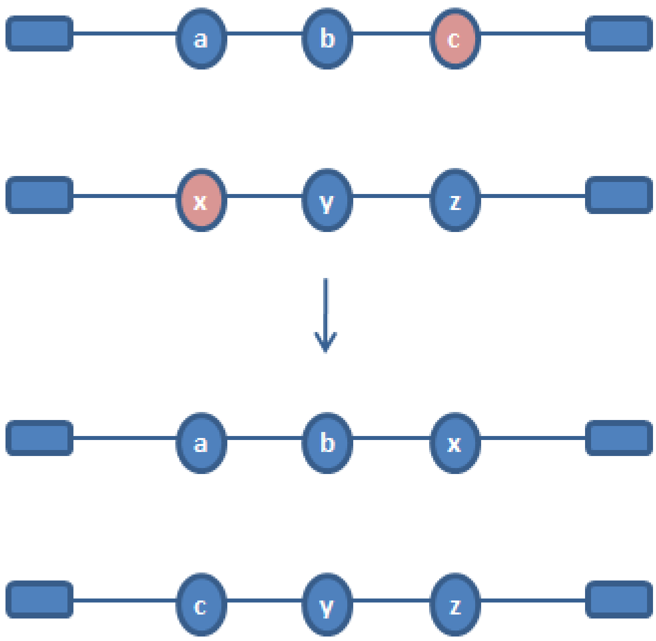

Figure 5. Swap Approach Used in the Shaking Step

For $N_k^2(\boldsymbol{x})$, the shaking is done by considering shift (of products) exchanges, i.e. the algorithm searches for the possibility of taking more or less products from the randomly selected supplier. If in the route supplier i belongs to, for a product supplied by i, only the batches of ℓ shifts are collected, a change is made on the value of ℓ. This change of shifts is repeated k times. Schematic representation of the second shaking is given in Figure 6.

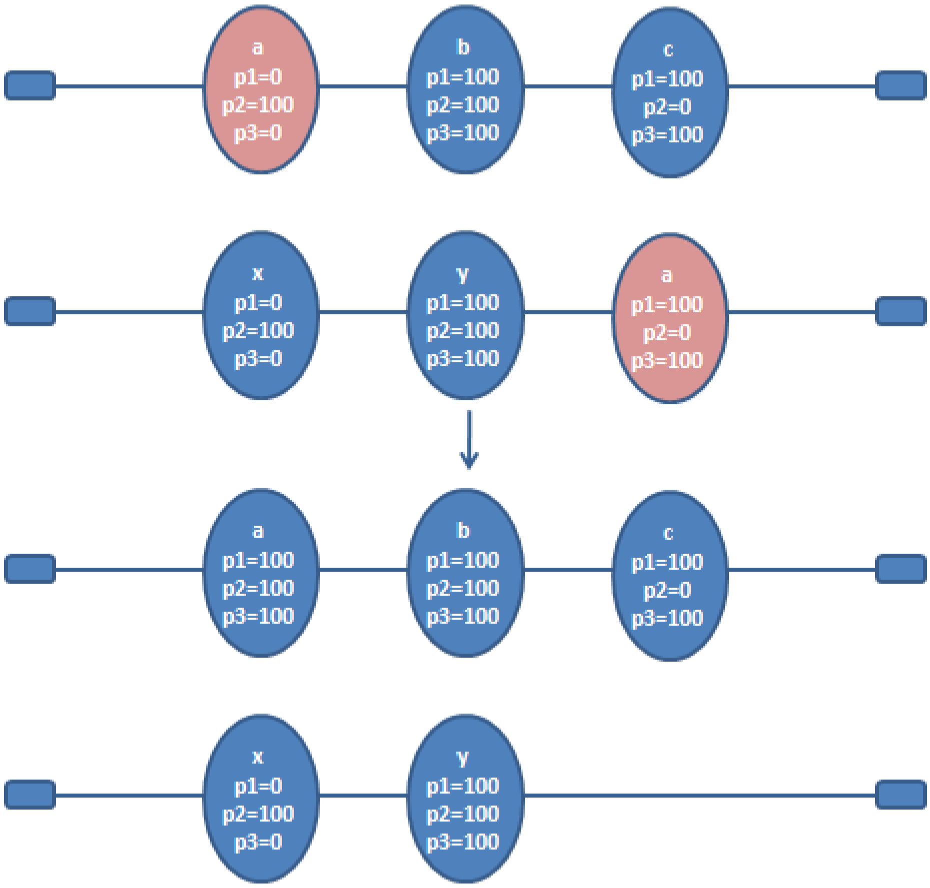

Figure 6. Product Exchange Approach for Shaking Step

Local search is applied to a solution obtained in the shaking step. Polacek et al. (2004) applied local search only on a route basis to increase the effectiveness of the algorithm. That is, after each shaking only the routes that have been changed need to be re-optimized. In the local search part, we used the same approach as proposed by Polacek et al. (2004) in order to save time and increase effectiveness of the algorithm. For the first type of shaking, we utilize the cheapest insertion method. Within the concept of cheapest insertion, changed suppliers are inserted to each position in the changed routes which are obtained during the shaking step.

As a result, position with the smallest cost is chosen as the new place of the supplier within the route. Such moves lead to relatively small neighborhoods. Therefore, it is expected to be computationally efficient. Cheapest insertion approach is represented in Figure 7.

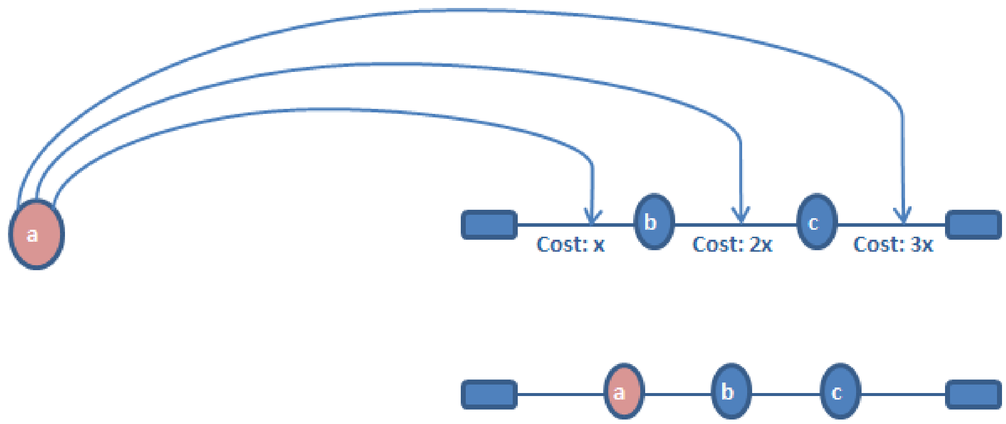

Figure 7. Cheapest Insertion Approach for Local Search

While implementing the cheapest insertion approach, whenever the algorithm generates a better route for supplier i, a search on $N_k^2(x)$ is carried out. Among all solutions generated the one with smallest objective function value is selected.

After the algorithm reaches to the local minimum, it compares the obtained result with the incumbent, if it is better than the incumbent, the value of k again set as 1 and the algorithm goes back to the shaking step. If the solution is worse than the incumbent, then the value of k is increased by the amount of step value. This situation continues until the stopping condition is met. Pseudo code of the proposed variable neighborhood search is described in Figure 8.

Initialization

Select the set of neighborhood structures $N_k^i(\boldsymbol{x})$, for i = 1,2, k=1,...,k_{max}

Generate the initial solution, x, by solving CVRP that repeats m times.

Return x: Initial solution

(1) Iterationnum:=0, Calculate $f(x)$.

Repeat Main Step until Iterationnum>Iterationnummax

Main Step:

(1) k=1;

(2) Repeat the following until $k=k_{max}$

(a) Shaking

For k=1 to k=$kmax$

Repeat the following until a feasible solution obtained

Randomly select $N_k^i(\boldsymbol{x})$, i = 1,2

If $N_k^1(\boldsymbol{x})$ is selected

i=randomly selected number from x

j=randomly selected number from x

Swap supplier i and supplier j

Else

i=randomly selected number from x

Check product exchange for supplier$i$

End If

Generate x' from the k^{th} neighborhood of x

Solve the VRP corresponding to x'

End for

Figure 8. Pseudo Code for Variable Neighborhood Search

```
(b) Local Search
    Repeat the following until improvement=no
        improvement=yes
        x″=x′
        For all x‴ ∈ N1(x″) that belongs to the routes
        changed in shaking step do
            Solve the VRP corresponding to each
            x‴ using cheapest insertion. Also per
            each x‴ with best possible solution
            after cheapest insertion, check
            product-shift exchanges for supplier i.
            Calculate f(x‴)
            x‴=min(x‴)
            If f(x″)>f(x‴) then
                x″=x‴
            Else
                improvement=no
            End if
        End for
    Localmin=x″
(c) Move or not
    If f(x″) <= f(x) then
        x=x″
        k=1
        Iterationnum=0
        goto step1
    Else
        k=k+1
        If(k<= kmax)
            goto step2
        Else
            Iterationnum= Iterationnum+1
            If(Iterationnum<=Iterationnummax)
                goto step1
            End If
        End If
    End if
```

Figure 8. Pseudo Code for Variable Neighborhood Search (continued)

4.4 Proposed Simulated Annealing Algorithm

In the construction of the simulated annealing approach, in order to decide on the cooling schedule, we firstly determined the initial and final temperatures. Based on the approach proposed by Altıparmak et al. (2009), we set the value of the initial temperature to 665 in which an inferior solution (inferior by 70% relative to current solution) is accepted with a probability of 0.90. And likely, we set the final temperature to 0.15 such that a solution which is inferior by %1 relative to current solution is accepted with a probability of 0.1%. The other component of the cooling schedule that is α, determined via parameter setting as described in the following sections.

The initial solution is generated in the same way as in variable neighborhood search algorithm.

In the neighborhood search, several approaches are used simultaneously. These are Swap, Insert, Swap Route and Random Route Assignment. In addition to the use of listed neighborhood approaches, the algorithm also searches for the possibility of taking more or less batches of a product (corresponding to different shifts) from the visited supplier i whenever a new neighborhood generated. After producing solutions using the described neighborhood search approaches, the solution with the smallest value between feasible solutions is taken as the new solution of the algorithm.

Swap approach means that randomly selected two suppliers are changed with each other in order to produce a neighbor solution, also algorithm checks for possible product exchanges as described above. Swap approach used in simulated annealing is same with the one used in the shaking step of proposed variable neighborhood search.

Insert is that, a randomly selected supplier is inserted in a randomly selected position within the solution space. The approach can be represented in Figure 9. In addition to that, algorithm searches for possible product exchanges for supplier "a" as explained above.

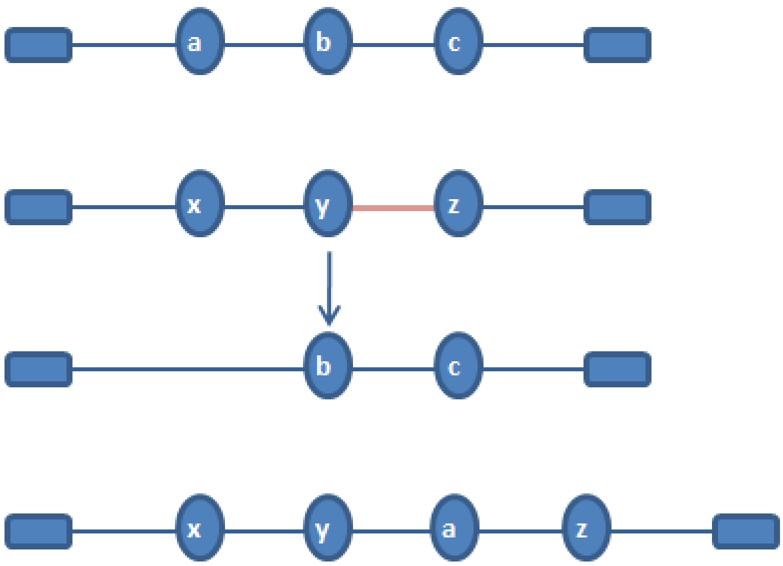

Figure 9. Insert Approach Used in Simulated Annealing

Swap route means that the route assignment of two suppliers is changed as represented in Figure 10. Also, product exchanges controlled for supplier "a" as described above.

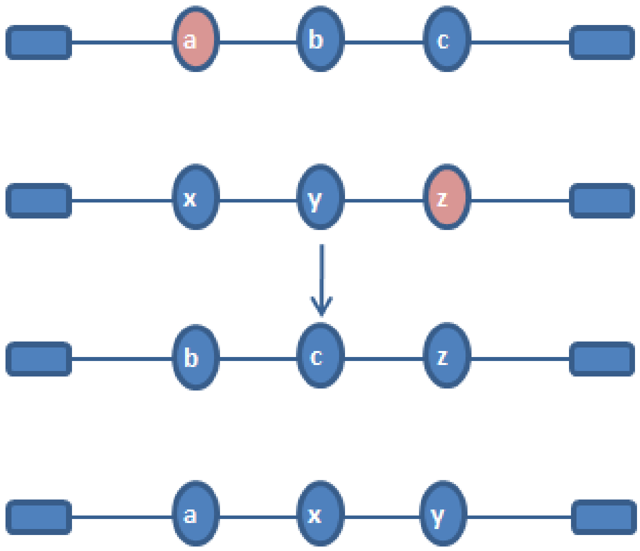

Figure 10. Swap Route Approach for Simulated Annealing

And finally, random route assignment is that the route assignment of a randomly selected supplier is changed with a randomly selected new route as shown in Figure 11. Here again, product exchanges for supplier “a” is checked for possible improvements as described above.

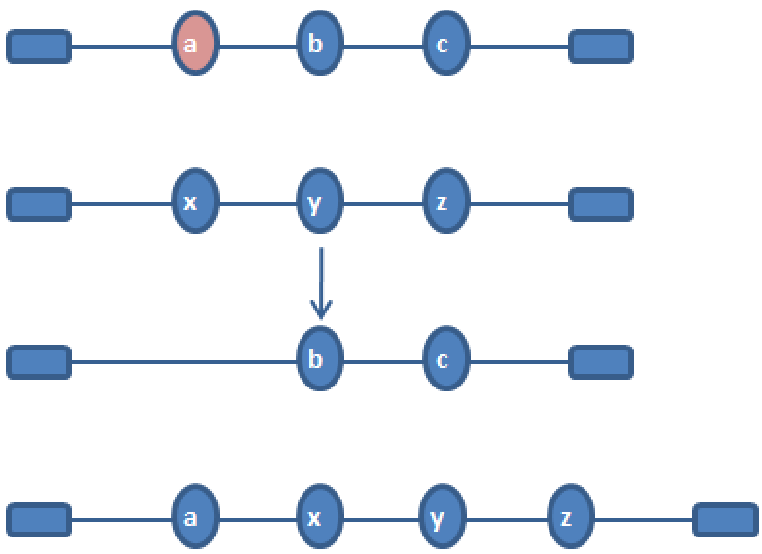

Figure 11. Random Route Assignment Approach for Simulated Annealing

Here a defined step of the algorithm finishes where iteration number is increased by one and the temperature is reduced based on the cooling approach. This continues until the stopping condition that is the final temperature is met.

Pseudo code of the proposed simulated annealing approach is described in Figure 12.

(0) Determine the initial parameters;

$T_0 \leftarrow 665$; $T_k \leftarrow T_0$; $T_f \leftarrow 0.15$; $\alpha \leftarrow 0{,}0001$; $k \leftarrow 0$

(1) Find Initial Solution:

(2) Generate the initial solution, *x*, by solving CVRP that repeats *m* times.

Return *x*: Initial solution

$x = S_0$;

$f(x) = f(S_0)$;

S_m=Current Solution;

$S_m = S_o$;

S_{best}=Best Solution;

$S_{best} = S_0$;

(3) *Neighborhood structure k=Swap, Insert, Swap Route, Random Route Assignment including possible product exchanges for i*

For all *k* do Solve the VRP corresponding to *x'*

according to $x' \in N_k(x)$

Calculate $f(x')$

$S_{new} = min(x')$

(4) If $S_{new} <= S_m$ then $S_m = S_{new}$

Goto *Step 4*

Else

$$\Delta = \frac{f(Snew) - f(Sm)}{f(Sm)} * 100$$

Generate a random number u using

uniform distribution u~U(0,1)

If $u \leq e^{-(\Delta/T)}$ then

$S_m = S_{new}$

Goto *Step 5*

End if

End if

Figure 12. Pseudo Code for Simulated Annealing

(5) If $f(S_m) < f(S_{best})$ then

$S_{best}=S_m$,

$f(S_{best})=f(S_m)$

End if

(6) $k=k+1$

$T_k= T_{k-1} / (1+ \alpha \sqrt{T_{k-1}})$

If $T_k>=T_f$ then

Goto *Step 2*

Else

Goto *Step 6*

End if

(7) Report S_{best}, $f(S_{best})$

Figure 12. Pseudo Code for Simulated Annealing (continued)

CHAPTER 5

COMPUTATIONAL EXPERIMENTS

In this section, in order to evaluate the performance of the proposed meta-heuristics, runs on test problems are conducted and results are reported. Firstly, the test problem sets used for runs are explained in detail. Secondly, parameter settings for the two approaches, variable neighborhood search and simulated annealing, are provided. At the end, computational results as per the conducted runs are reported.

5.1 Problem Sets

For the evaluation of the performance of proposed solution approaches vehicle routing problem instances taken from the CVRP literature are utilized.

The reason behind selecting the CVRP instances is that, we have capacitated homogeneous vehicles in our problem. In addition to the capacity constraints, we have shift based time restrictions which are different from the regular time window constraints. Therefore, we did not utilize VRPTW benchmark instances. All the instances are available in VRP web site. List of instances that we utilized is given in Table 3.

In all the instances, we assumed m=3, and each supplier provides only one product. Additionally, the amounts of supply for each supplier per each shift assumed to be same that is; each supplier supplies the same size of product batch for each shift. For all instances, locations of suppliers and the depot (cross dock for our problem) are provided. With the help of location coordinates, we generated our symmetric distance matrix.

In the demand section of the instances, demands of customers are provided; however, since we consider that there exist three shifts within a day in our problem setting, we multiply the given product demands by three, that is, one batch for each shift. The vehicle capacities are directly taken from literature instances and assumed to be the same for all vehicles. Total number of vehicles are again taken from instances but here each vehicle used at most *m* times, so as to provide limited number of vehicles per each shift.

Table 3. CVRP Literature Instances

Instance	**Article**	**# of Nodes**	**# of Vehicles**	**Vehicle Capacity**
(1)	Augerat, et al.(1995)	37	5	100
(2)	Augerat, et al.(1995)	40	5	140
(3)	Augerat, et al.(1995)	45	7	100
(4)	Augerat et al.(1995)	65	10	130
(5)	Augerat et al.(1995)	78	10	100
(6)	Christophides and Eilon(1969)	22	4	6000
(7)	Christophides and Eilon(1969)	23	3	4500
(8)	Christophides and Eilon(1969)	30	3	4500
(9)	Christophides and Eilon(1969)	101	8	200

In these instances, no information related with shift-based time constraints exist. Therefore, the latest arrival times of product batches to the cross dock are defined as different scenarios, by giving deadlines derived based on the longest tour length of the best solutions reported in the literature.

For each instance, we considered three scenarios, resulting with shift-based constraints ranging from tight to loose. Latest arrival times for each shift in these scenarios are listed in Table 4.

Table 4. Latest Arrival Time Scenarios for Products

	Latest Arrival Time for *shift1*	Latest Arrival Time for *shift2*	Latest Arrival Time for *shift3*
Scenario 1	Longest Tour Length	2×Longest Tour Length	3×Longest Tour Length
Scenario 2	1.2×Longest Tour Length	2.4×Longest Tour Length	3.6×Longest Tour Length
Scenario 3	1.4×Longest Tour Length	2.8×Longest Tour Length	4.2×Longest Tour Length

5.2 Parameter Setting

For solving CVRP-STC, we have proposed two meta-heuristic approaches, variable neighborhood search and simulated annealing. The initial solutions used in both approaches require the solutions of CVRP that are repeated m times. These solutions for CVRP are taken as the best reported solutions in the literature. Both approaches use some parameters during their process which needs to be determined before computational experiments. The parameter settings for these meta-heuristic approaches are explained in detail in the following sections.

5.2.1 Parameter Setting for Variable Neighborhood Search

The parameters that require setting in the variable neighborhood search are the stopping condition and the k_{max} value.

In order to determine the k_{max} value, we made test runs using the Augerat et al.(1995) VRP instance for 40 nodes and 5 vehicles. We assigned value of 4, 5, 7, 10, 15 and 20 for k_{max}. The values assigned to k_{max} and steps are represented in Table 5 below.

Table 5. List of k_{max} and Step Values Assigned for Parameter Setting

k_{max}	Step
4	1
5	1
7	1
10	2
15	3
20	4

We conducted test runs so as to let the algorithm to converge for each k_{max} and step value pair. The results of the algorithm for the solution of the Augerat et al. (1995) VRP instance for 40 nodes and 5 vehicles using different k_{max} values are represented in Figure 13.

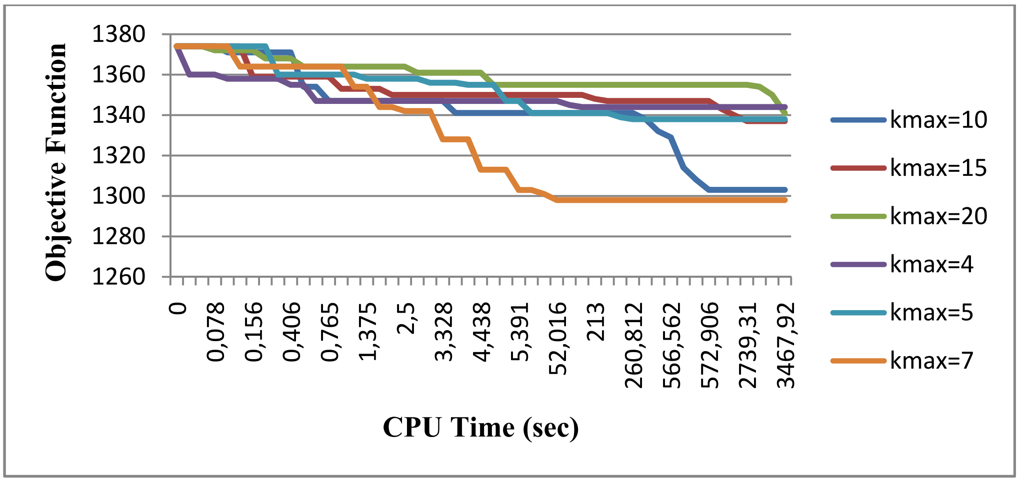

Figure 13. Results for proposed k_{max} and step values using instance 2

When the value of k_{max} is 7, the algorithm reaches to the best solution. k_{max}=7 is better than the case 4 and 5, since it makes larger perturbations to the current solution which result in better results. The performance of the algorithm was not promising when k_{max} value is greater than 7. Since the initial solution that the algorithm uses is the best solution reported so far in the literature, larger changes to the solution are either gives worse or infeasible solutions. Table 6 indicates the evaluation of results as per the run for each k_{max} and step combination given above. In the table, the CPU time in seconds at which each combination reaches the best solution are reported where the highlighted CPU shows the k_{max} value that reaches the best solution in the shortest time and the highlighted objective function shows the overall best solution.

Table 6. Evaluation of algorithm as per each k_{max} and step combination using instance 2

Augerat et al. (1995) – 40 Nodes, 5 Vehicles			
k_{max} **Value**	**Step Value**	**Best Result**	**CPU Time(sec)**
4	1	1344	71,359
5	1	1338	260,812
7	1	1298	52,016
10	2	1303	572,906
15	3	1337	2739,31
20	4	1341	3467,92

The CPU time of the algorithm when k_{max}=7 is the best, this condition also reaches the overall best solution. The difference of the best solutions between k_{max}=7 and the k_{max}=10 is close to each other but when k_{max}=7 algorithm reaches a better solution about 9 minutes earlier. In the remaining instances objective function values are close to each other and worse than the case of k_{max}=7. Thus we set the value of k_{max} as 7.

As the stopping condition time limit or iteration limit is preferred in general. In our problem, since time limit can stop the algorithm when there is a possibility of producing more promising solutions, we choose to use the iteration limit as the stopping condition. We define the iteration number as the number of iterations without any improvement on the best solution. The iteration limit is set as 5000, a promising limit based on the results reported above.

5.2.2 Parameter Setting for Simulated Annealing

In our simulated annealing approach, we firstly generate an initial solution, S_0, then in the neighborhood of the original solution, within the defined approach a new solution is produced, S_{new}. At this point, the amount of change in the objective function value is calculated using $\Delta = f(S_{new}) - f(S)$. If the obtained Δ value is $\Delta \leq 0$, then since the problem is a minimization problem, a move to the new solution is accepted. In case of $\Delta > 0$, the move to the new solution is accepted with a specified probability, denoted by $\exp^{-|\Delta|/T}$. T is a control parameter which corresponds to temperature which is reduced during the search. In the cooling strategy, there is a single iteration at each temperature in which the temperature determined by $T_k = (T_k - 1)/(1 + \alpha\sqrt{T_k - 1})$, where T_k corresponds to the temperature at the k^{th} iteration and α is the cooling ratio. In the algorithm, the temperature in the step 0 is equals to the initial temperature that is $T_k \leftarrow T_0$ where k is equals to zero at the beginning ($k \leftarrow 0$). The neighborhood search structures that will be used within the context of simulated annealing have already been determined as described in the above sections.

Remaining parameters that require settings in simulated annealing are initial temperature, T_0, cooling ratio, α, stopping condition namely final temperature, T_f.

The initial temperature that is used in the proposed simulated annealing approach is set to 665 in which an inferior solution (inferior by 70% relative to current solution) is accepted with a probability of 0.90. The final temperature is assumed as 0.15 such that a solution which is inferior by %1 relative to current solution is accepted with a probability of 0.1% as described by Altıparmak et al. (2009).

In our algorithm, we assigned α values of 0.0001, 0.0005, 0.001, 0.005, 0.01 and 0.02. We tried to find the best value of α via experimental search using the instances of Christophides and Eilon(1969) with 30 nodes and 3 vehicles, and the instances of Augerat, et al. (1995) with 37 nodes and 5 vehicles, with 40 nodes and 5 vehicles, with 45 nodes and 5 vehicles. Results of the conducted experimental search are listed in Table 7 which shows the best solution and CPU time in seconds at which each alternative reaches the best solution within the defined cooling approach. In the table, highlighted values indicate α value that reaches the best solution.

In each trial best solution of the algorithm based on predefined cooling strategy is found when α value is 0.0001. In contrast to that, higher CPU times also belong to α=0.0001. At this point, since the represented CPU values are in seconds and the longest duration when Augerat, et al. (1995) 45 nodes and 5 vehicles test problem used is about 970 seconds, they can be regarded as tolerable. Therefore, α value is assigned as 0.0001 for the computational experiments.

Table 7. CPU Time and Best Solutions for Different α Values

	α											
	0.0001		**0.0005**		**0.001**		**0.005**		**0.01**		**0.02**	
Instance	**Best Result**	**CPU Time(sec)**	**Best Result**	**CPU Time(sec)**	**Best Result**	**CPU Time(sec)**	**Best Result**	**CPU Time(sec)**	**Best Result**	**CPU Time (sec)**	**Best Result**	**CPU Time(sec)**
1	1449	619.90	1511	103.39	1644	61.82	1801	7.79	1932	4.64	1971	2.06
2	1266	627.21	1349	104.65	1359	50.45	1360	9.65	1373	3.81	1374	1.68
3	3216	970.37	3234	183.23	3338	77.25	3378	12.45	3389	6.98	3428	2.62
8	1107	298.96	1220	59.96	1270	31,47	1287	4,86	1380	2,42	1577	0,97

5.3 Computational Results

In this section, the performance of the proposed meta-heuristic solution approaches are reported based on the runs on the problems selected from the CVRP literature. We also provide a detailed analysis on the solution results.

5.3.1 Results of the Test Runs

The computational results for VNS and SA approaches are discussed in detail in that section. The proposed approaches are coded in C++ programming language. The computational experiments are conducted using the defined problem sets, gathered from CVRP literature. The runs are conducted on the computers with Intel Core 2 Duo 3.00 GHz CPU processor and 3.49 GB of RAM.

Test runs are executed for each instance for three shift-deadline scenarios for both VNS and SA algorithms. Each test problem is run for both algorithms and the average results are reported. For the performance evaluation of the proposed approaches, the performance measures of CPU time, percent deviation from the best value, percent deviation from the initial solution result and percentage of the products collected in earlier shifts are used. Percentages of the test results are calculated with the given formulation;

$$\%Dev_{best} = [(f_{heuristic} - f_{best})/\ f_{best}] \times 100$$

$$\%Dev_{initial} = [(f_{initial} - f_{heuristic})/\ f_{initial}] \times 100$$

$$\%EC = S_{early}/\ S_{total} \times 100$$

At these formulations, $\%Dev_{best}$ indicates the percent deviation from the best solution produced, $\%Dev_{initial}$ states the percent deviation from the initial solution while $f_{initial}$ and $f_{heuristic}$ correspond to objective function value of the initial solution, and the solution generated by a specific heuristic, respectively. f_{best} is the best objective function value generated by either of the meta-heuristic procedures. $\%EC$ corresponds to the ratio of the products collected in earlier shifts than they required, S_{total} indicates the total amount of products defined in the problem and S_{early} is the amount of products collected earlier than the required shifts.

Firstly, we report in Table 8 the results of the test runs for scenario 1, where the shift deadlines are determined by the longest tour time of the best solution in the literature.

Table 8. Results for scenario 1, where the shift deadlines are determined by the longest tour time

Instance	|N|	|V|	% Deviation from Best		CPU (min)		% Deviation from Initial Solution		% Early Collected	
			VNS	SA	VNS	SA	VNS	SA	VNS	SA
6	22	4	2.72	0.00	2.54	3.53	6.60	9.07	8.83	14.43
7	23	3	40.42	0.00	5.35	3.27	28.48	49.07	21.22	50.75
8	30	3	12.35	0.00	6.00	4.10	9.99	19.89	11.62	23.10
1	37	5	16.96	0.00	9.79	9.16	8.13	21.45	10.12	29.30
2	40	5	5.70	0.00	8.54	13.78	2.61	7.86	5.39	10.02
3	45	7	0.27	0.00	11.09	15.19	5.45	5.70	9.20	14.58
4	65	10	0.83	0.00	18.04	56.23	2.69	3.48	5.51	8.42
5	78	10	2.14	0.00	55.03	39.16	2.44	4.49	3.87	22.80
9	101	8	0.00	0.07	56.95	83.10	3.12	3.06	4.79	14.73
Minimum			**0.00**	**0.00**	**2.54**	**3.27**	**2.44**	**3.06**	**3.87**	**8.42**
Average			**9.04**	**0.01**	**19.26**	**25.28**	**7.72**	**13.79**	**8.95**	**20.90**
Maximum			**40.42**	**0.07**	**56.95**	**83.10**	**28.48**	**49.07**	**21.22**	**50.75**

In Table 8 the reported values are the averages of five runs. Results per each run for both algorithms are listed in Appendix A. As seen from the results SA produces better solutions in all instances except instance 8. For all of the test runs it can be said that on the average SA produces about 9% better solutions than VNS. However, from the perspective of the CPU time it is clear that SA is not always the best. While in some cases SA works faster, it is seen that in more than half of the instances VNS works faster than SA. On the average VNS gives the result in about 360 seconds that is like 6 minutes earlier than SA. Both of the algorithms provide promising results compared to the initial solution. Regarding the initial solutions again SA acts better, that is on the average SA makes about 6% more improvement than VNS which is a total of about 14% improvement compared to the initial solution. Considering the ratio of the products collected from suppliers in earlier shifts, SA is about 12% greater than VNS on the average. That is using SA on the average about 21% of products are collected earlier than they required.

In Table 9, we provide the results for scenario 2, where the shift deadlines are determined by multiplying the longest tour time by 1.2. Here we have not so strict time deadlines. The listed values are again the averages of five test runs. Results per each run for both algorithms are listed in Appendix B. It is clear from the results that SA produces better solutions for all of the instances. Regarding the results per each instance, it can be said that on the average SA produces about 8% better solutions than VNS.

Table 9. Results for scenario 2, where the shift deadlines are determined by 1.2×the longest tour time

Instance	|N|	|V|	% Deviation from Best		CPU (min)		% Deviation from Initial Solution		% Early Collected	
			VNS	SA	VNS	SA	VNS	SA	VNS	SA
6	22	4	9.33	0.00	3.74	2.63	7.72	15.59	10.97	20.91
7	23	3	16.89	0.00	11.55	2.96	42.93	51.18	39.11	47.84
8	30	3	14.90	0.00	12.27	4.63	29.25	38.42	22.12	30.58
1	37	5	21.71	0.00	18.32	7.78	8.99	25.22	10.89	32.93
2	40	5	5.38	0.00	6.80	8.17	3.68	8.60	4.62	15.10
3	45	7	3.42	0.00	19.48	14.26	5.35	8.48	8.62	23.91
4	65	10	0.78	0.00	24.45	34.82	3.75	4.50	7.16	12.48
5	78	10	0.98	0.00	129.06	35.20	3.59	4.53	8.79	26.49
9	101	8	0.28	0.00	80.68	83.77	4.61	4.88	6.52	23.05
Minimum			**0.28**	**0.00**	**3.74**	**2.63**	**3.59**	**4.50**	**4.62**	**12.48**
Average			**8.19**	**0.00**	**34.04**	**21.58**	**12.21**	**17.93**	**13.20**	**25.92**
Maximum			**21.71**	**0.00**	**129.06**	**83.77**	**42.93**	**51.18**	**39.11**	**47.84**

In contrast to the first case, in the second case in most of the instances SA works faster than VNS comparing the CPU times. On the average SA produces the result about 747 seconds that is like 12.5 minutes earlier than VNS. Similar to the first one, both of the proposed algorithms provide promising results compared to the initial solution. Regarding the initial solutions again SA acts better, that is on the average SA makes about 6% more improvement than VNS that is very close to the situation in the first case. On the average SA totally makes about 18% improvement on the initial solution. Similar to the first case, using SA, the ratio of the products collected from suppliers in earlier shifts is on the average about 12.7% greater than VNS which corresponds to the on the average about 26% early collection for SA.

In Table 10, we provide the results for scenario 3, where the shift deadlines are determined by multiplying the longest tour time by 1.4. Here we have the loosest time deadlines. Like in the other cases, in the Table 10 the reported values are the averages of the test runs. Results per each run for both algorithms are listed in Appendix C. Based on the listed results, it is clear that SA produces better solutions than VNS in all of the cases. Taking into consideration the entire test runs it can be said that on the average SA produces about 9% better solutions than VNS. Unlike to the situations in the first and second cases, from the perspective of the CPU time it is clear that SA is produces faster solutions in each of the instances. On the average SA gives the result about 1662 seconds that is like 27 minutes earlier than VNS. Like in the other cases, both of the algorithms provide promising results compared to the initial solution, When compared with the initial solutions again SA produces better results. On the average SA makes about 6% more improvement than VNS that is SA improves the initial solution by about 22%. The ratio of the products collected from suppliers in earlier shifts using SA is on the average about 14% greater than VNS which is also greater than the other cases.

Table 10. Results for scenario 3, where the shift deadlines are determined by 1.4×the longest tour time

Instance	|N|	|V|	% Deviation from Best		CPU (min)		% Deviation from Initial Solution		% Early Collected	
			VNS	SA	VNS	SA	VNS	SA	VNS	SA
6	22	4	13.09	0.00	6.70	2.62	14.76	24.62	15.50	26.33
7	23	3	13.80	0.00	24.39	2.91	45.88	52.44	41.14	51.80
8	30	3	7.65	0.00	20.61	5.02	34.67	39.32	28.35	30.82
1	37	5	25.39	0.00	34.18	7.70	17.22	33.98	18.96	38.59
2	40	5	14.19	0.00	9.63	7.99	7.57	19.05	9.26	24.78
3	45	7	3.16	0.00	33.38	11.84	7.05	9.90	12.14	27.17
4	65	10	5.00	0.00	34.23	30.69	4.96	9.49	8.59	24.04
5	78	10	1.03	0.00	174.49	32.97	3.60	4.59	9.63	30.19
9	101	8	0.62	0.00	96.53	83.01	5.15	5.73	7.81	26.25
Minimum			**0.62**	**0.00**	**6.70**	**2.62**	**3.60**	**4.59**	**7.81**	**24.04**
Average			**9.33**	**0.00**	**48.24**	**20.53**	**15.65**	**22.13**	**16.82**	**31.11**
Maximum			**25.39**	**0.00**	**174.49**	**83.01**	**45.88**	**52.44**	**41.14**	**51.80**

5.3.2 Detailed Analysis on the Results

Analyzing the results, it is seen that solution results for the proposed VNS algorithm get better when the shift deadlines get looser. That means, comparing the first and second scenarios, VNS produces worse solutions in the first scenario, similarly, when we compare second and third scenarios, it can be said that third scenario produces better solutions than the second one in nearly all of the instances. On the contrary, CPU times required to obtain the solutions get worse when the shift deadlines get looser. That is, while in the first scenario we see the shortest CPU times, they are longest in the third scenario in again nearly all of the trials. The reason behind obtaining worse solutions in shorter times in the first scenario is that, since, it is more restrictive regarding the shift lengths; it is hard for the algorithm to find better solutions. If the algorithm cannot find a better solution the iteration number will be increased and the stopping condition will be met earlier. At this point, as it is easier to find better solutions when the restrictions are relaxed, the algorithm produces better solutions. Additionally, the stopping condition in the proposed VNS algorithm is the number of iterations without any improvement on the best solution, since with relaxed constraints it is more possible to find better solutions; it takes longer time to meet the stopping condition in the third scenario.

Considering the results, it appears that both the solution results and the CPU time for the proposed SA algorithm get better when the restriction on the shift deadlines get looser. At this point, when we compare the first and second scenarios, SA produces worse solutions in the first scenario regarding both the objective functions and solution time, similarly, when we compare second and third scenarios, it is clear that third scenario produces better solutions than the second one in nearly all of the trials in the same perspective. The reason behind obtaining better solutions in shorter times in the third scenario is that, since, the restrictions are relaxed in the third scenario; it is easier for the algorithm to find better solutions. When the algorithm produces better solutions, it passes to the next iteration with a cooler temperature. That means the cooling is faster when the constraints are relaxed which results in better solutions in shorter time frames.

As mentioned in the above sections, SA algorithm provides better solutions than VNS in all cases. But the amount of deviation between the solutions obtained by each of the approaches remains nearly same around 9%. That can be interpreted as; the reaction of both of the algorithms against the restriction on the shift deadlines is in the same manner. But only in the first scenario, VNS produces faster solutions than SA, which is on the average 6 minutes. This duration can be regarded as tolerable in favor of SA considering 27 minutes difference in the third scenario. Therefore, within these settings usage of SA algorithm will be more beneficial.

Taking into consideration the results of the third scenario, deviation from the initial solution is most promising that is 45.88% for VNS and 52.44% for SA when instance 7 is studied. Also, the deviation from the initial solution is least promising that is 3.60% for VNS and 4.59% for SA when instance 5 is considered. In instance 7, longest tour length is 289 units and the closest tour length is 211 where the difference is 87 units. But for instance 5 largest tour length is 222 units and the closest tour length is 204 where the difference is 18 units. At this point, it can be said that when the difference between the tour lengths increases, and also when the largest tour length gets bigger, the flexibility of the algorithm increases and better improvements become possible. Here, flexibility means, the possibility of finding alternative routes between shifts which promise better solutions.

Considering the ratio of products collected in earlier shifts than they required, it is clear that the ratio increases for both VNS and SA approach from scenario 1 to scenario 3. Concentrating on the instances 7 and 2, we try to investigate the behavior of the algorithm against the initial solution in second scenario.

For both of the problems in the initial solution, algorithm repeats the same routes in each shift visiting each supplier three times a day. In the first instance with 23 nodes, algorithm trends to carry some of the product batches that belong to second shift in the first shift. Similarly, algorithm tends to carry some of the products batches of the third shift in the first and second shift. Like in the example of 23 nodes, in 78 nodes instance again algorithm prefers to carry some of the product batches in the shifts that are earlier than the time that they required to be at the cross dock. At this point, it can be said that when the algorithm searches for better solutions, it slides the possible product batches to the vehicles of earlier shifts if there exist enough vehicle capacity or if the length of tour of a vehicle is shorter than the shift length, the algorithm tends to use the vehicle second time in the same shift which also produce promising solutions. Therefore, as the lengths of the shifts are extending beginning from the first scenario till the third scenario, the algorithm finds better solutions with the relaxation of the time restrictions taking the advantage of early collection. In some instances even the solution results for both VNS and SA approach are close to each other, the ratios of early collection of products are higher for SA. The reason behind is that; in SA approach, construction of neighbor solutions considers more tour exchanges for suppliers comparing with VNS.

As reported the performance of the proposed VNS algorithm is worse than the SA approach. Therefore, in order to increase the effectiveness of the VNS algorithm we tried a different local search structure that is called 2-opt* which is a special inter-tour move. Within the concept of 2-opt*, two connections, one from each tour, are removed which produces four sub-tours that are then recombined. 2-opt* approach is represented in Figure 14.

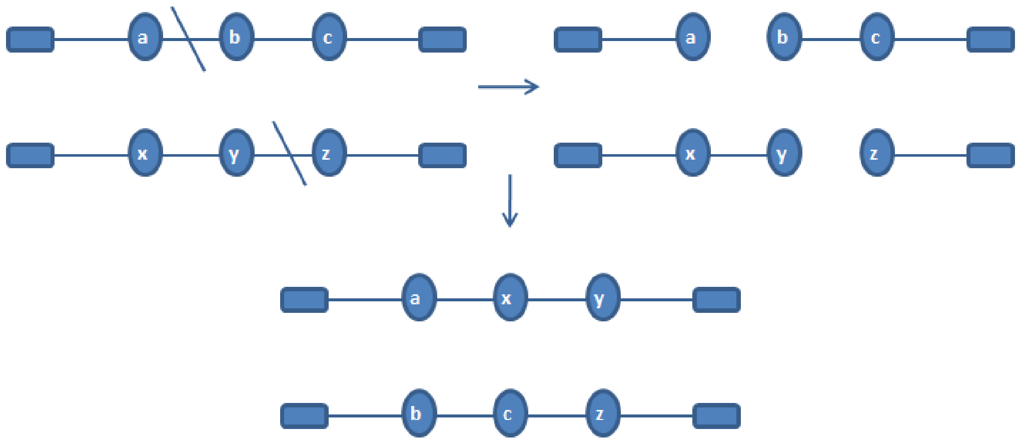

Figure 14. 2-opt* Approach for Local Search

In Table 11, we provide the average results using 2-opt* approach in local search for scenario 2, where the shift deadlines are determined by multiplying the longest tour time by 1.2. Results per each run are listed in Appendix D. When we used the method of 2-opt* in the local search phase rather than the method of cheapest insertion, we obtain better solutions but this time solution times get too long.

Table 11. Results for scenario 2, where the shift deadlines are determined by 1.2×the longest tour time using 2-opt* approach in local search

Instance	**|N|**	**|V|**	**% Improvement Over**		**CPU (min)**	**% Deviation from Initial Solution**	**% Early Collected**
			VNS(CI)	**SA**			
6	22	4	15.78	7.93	53.58	22.28	21.35
7	23	3	22.71	9.65	657.72	55.89	36.22
8	30	3	14.69	1.98	262.75	39.64	29.39
1	37	5	24.54	8.16	927.72	31.32	29.42
2	40	5	15.65	11.11	103.33	18.75	18.67
3	45	7	5.33	2.10	1142.58	10.39	19.22
4	65	10	2.89	2.13	351.75	6.54	9.65
5	78*	10	4.30	3.36	605.49	7.74	20.62
9	101*	8	3.34	3.07	606.03	7.80	12.28
Minimum			**2.89**	**1.98**	**53.58**	**6.54**	**9.65**
Average			**12.14**	**5.50**	**523.44**	**22.26**	**21.87**
Maximum			**24.54**	**11.11**	**1142.58**	**55.89**	**36.22**

* - Since the algorithm could not terminate in 7200 minutes that is 120 hours, we report the results obtained when the algorithm terminated after 10 hours.

For example regarding the second scenario, for the problem with 22 nodes best solution obtained with the 2-opt* method is 846, which is even smaller than the average result of SA algorithm for the same scenario that is 949.6. But the duration when 2-opt* used is nearly 64 minutes. This duration is on the average about 4 minutes for VNS and 3 minutes for SA approach. The difference is about 60 minutes that is 1 hour for the smallest size example. When the size of the problem increases gap is also increases. For example again regarding the second scenario, for 37 nodes instance using 2-opt* method, the objective function value is 1344, and the duration is about 848 minutes where on the average the objective function value and the duration is respectively 1826 and 18 for VNS and 1500 and 8 for SA approach. As seen again the 2-opt* method finds better solutions than all of them but the difference of the durations is about 830 minutes which corresponds to about 13.8 hours. At this point 2-opt* method can be advisable if the user have time flexibility while obtaining the solutions.

CHAPTER 6

CONCLUSIONS

In this study, we work on the capacitated vehicle routing problem in cross docks where shift based time constraints on products are taken into consideration. In our problem setting, the dealt cross dock is assumed to feed directly the production lines of its customer. At this point, the customer defines frequencies which are corresponds to shifts to get the products based on its production rate. Based on the defined number of shifts, time deadlines per each shift is also determined within the day. Therefore, in order to meet the customer's demand, products are collected from the suppliers regarding these deadlines which results in shift-based time constraints on products. In our study, the objective is to minimize the routing costs for inbound routes. At this point, it is very important to decide on the product batches that will be taken from a supplier when a vehicle arrives at the supplier. In order to solve the problem firstly a mathematical model is formulated. Since the dealt problem is NP-Hard, meta-heuristic solution approaches based on variable neighborhood search and simulated annealing are also proposed. For the evaluation of the performance of the proposed algorithms computational experimentation is conducted on the test problems which are tailored from the capacitated vehicle routing instances from the literature.

Based on the conducted computational study, it is seen that the performance of the SA algorithm is much better than the VNS approach, both regarding the solution times and objective functions. Moreover, both of the algorithms are providing promising solutions compared with the initial solutions considering that the initial solutions are taken as the best solutions reported in the literature.

The proposed algorithms react in the same manner to the changes on the shift lengths. For the experiments we defined three different shift lengths based on the longest tour lengths of the best solutions of the literature examples ranging from tight to loose. Both approaches produce better solutions when the constraints on shift end times get looser. In contrast, while the solution duration of the VNS approach gets longer, duration of the SA algorithm gets shorter with the relaxation of the shift lengths which also favors SA approach.

In order to obtain better solutions we tried a different local search structure for VNS algorithm which is 2-opt* method. This time, produced solutions are better even from SA in the objective function perspective but the solution durations extend too much to be acceptable. The 2-opt* approach can be preferred if the user have time flexibility while obtaining the solutions.

The test problems used to evaluate the performance of the proposed meta-heuristic algorithms are tailored from CVRP literature and the initial solutions are taken as the best solutions reported in the literature. In case of using the proposed algorithms for the problems rather than literature instances, there will be a need to produce initial solutions. At this point, initial solutions can be produced using existing VRP heuristics like nearest neighbor search algorithm regarding the capacity and time restrictions.

The problem of capacitated vehicle routing with shift based time constraints on products is applicable to the companies in which mass production on a shift basis is taken into consideration like in automotive or domestic appliance industries. In accordance with the preferences of the customer a cross dock can be added to the logistics flow in order to reduce the amount of inventory kept in company area and also in order to increase the effectiveness of the logistics operations with full truck load deliveries to the customer. With the described settings the required products will be delivered to the customer at the beginning of each shift which will directly feed the production lines. Since the proposed meta-heuristic approaches are constructed considering basic characteristics, they can be applicable to the real life examples with little or no change based on problem specific requirements.

There are many extensions are possible for the future works. One of them can be adding holding costs at the cross dock for early arrivals of product batches. Effects of different number of shifts can be investigated. Considering heterogeneous set of vehicles will be a harder problem. Also, taking in to consideration the capacitated cross docks can be another extension. Additionally, considering the combination of the proposed problem with the internal cross dock operations will lead to a different problem. Also, development of different metaheuristic approaches may be another extension. Moreover, splitting the delivery of a batch of product can be regarded as another approach.

REFERENCES

Aleman E.R., Zhang X., Hill R.R. (2010), "An adaptive memory algorithm for the split delivery vehicle routing problem", J Heuristics, Volume 16, Pages 441-473

Altıparmak F., Gen M., Lin L., Karaoğlan İ. (2009), "A steady-state genetic algorithm for multi-product supply chain network design", Computers & Industrial Engineering, Volume 56, Pages 521-537

Arabani B.A., Ramtin F., Rafienejad N.S. (2009a), "Applying Simulated Annealing Algorithm for Cross Docking Scheduling", Proceedings of the World Congress on Engineering and Computer Science 2009, Volume 2, WCECS 2009, October 20-22, 2009, San Francisco, USA

Arabani B.R.A., Ghomi F.T.M.S., Zandieh M. (2009b), "A multi-criteria cross-docking scheduling with just-in-time approach", Int J Adv Manuf Technol, DOI 10.1007/s00170-009-2429-5

Augerat P., Belenguer J.M., Benavent E., Corber´an A., Naddef D., Rinaldi G. (1995), "Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem" Research Report 949-M, Universit´e Joseph Fourier, Grenoble, France

Boysen N. (2010), "Truck scheduling at zero-inventory cross docking terminals", Computers & Operations Research, Volume 37, Pages 32-41

Boysen N., Fliedner M. (2009), "Cross Dock Scheduling: Classification, Literature Review and Research Agenda", Jena Research Papers in Business and Economics, 2/2009, ISSN 1864-3108

Boysen N., Fliedner M., Scholl A. (2008), "Scheduling inbound and outbound trucks at cross docking terminals", OR Spectrum, DOI 10.1007/s00291-008-0139-2

Briskorn D., Choi B-C., Lee K., Leung J., Pinedo M. (2009), "Single machine scheduling with regard to inventory constraints", February 12, 2009

Chen F., Lee C-Y. (2009), "Minimizing the makespan in a two-machine cross-docking flow shop problem", European Journal of Operational Research, Volume 193, Pages 59-72

Chen F., Song K. (2009), "Minimizing makespan in two-stage hybrid cross docking scheduling problem", Computers & Operations Research, Volume 36, Pages 2066-2073

Christofides N., Eilon S. (1969), "An Algorithm for the Vehicle Dispatching Problem", Operational Research, Quarterly 20, Pages 309–318

Çakır B., Altıparmak F., Dengiz B. (2011), "Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm", Computers & Industrial Engineering, Volume 60, Pages 376–384

Deng A., Mao C., Zhou Y. (2009), "Optimizing Research of an Improved Simulated Annealing Algorithm to Soft TimeWindows Vehicle Routing Problem with Pick-up and Delivery", SETP, Volume 29, Issue 5, Pages 186-192

Derigs U., Gottlieb J., Kalkoff J., Piesche M., Rothlauf F., Vogel U. (2010), "Vehicle routing with compartments: applications, modelling and heuristics", OR Spectrum, DOI 10.1007/s00291-010-0194-3

Desrochers M., Lenstra K.J., Savelsbergh P.W.M. (1990), "A Classigication Sheme for Vehicle Routing and Scheduling Problems", Centre for Mathematics and Computer Science, Report OS-R8721, November

Dondo R., Méndez A.C., Cerdá J. (2011) "The multi-echelon vehicle routing problem with cross docking in supply chain management", Computers and Chemical Engineering, DOI:10.1016/j.compchemeng.2011.03.028

Ekşioğlu B., Vural V.A., Reisman A. (2009), "The vehicle routing problem: A taxonomic review", Computers & Industrial Engineering, Volume 57, Pages 1472-1483

Fallahi E.A., Prins C., Calvo W.R. (2008), "A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem", Computers & Operations Research, Volume 35, Pages 1725 – 1741

Francis P., Smilowitz K. (2006), "Modeling techniques for periodic vehicle routing problems", Transportation Research, Part B, Volume 40, Pages 872–884

Gulczynski D., Golden B., Wasil E. (2010), "The split delivery vehicle routing problem with minimum delivery amounts", Transportation Research, Part E, Volume 46, Pages 612–626

Hansen P., Miladovic N. (2001), "Variable Neighborhood Search: Principles and applications", European Journal of Operational Research, Volume 130, Pages 449-467

Hashimoto H., Yagiura M., Ibaraki T. (2008), "An iterated local search algorithm for the time-dependent vehicle routing problem with time Windows", Discrete Optimization, Volume 5, Pages 434–456

Hemmelmayr C.V., Doerner F.K., Hartl F.R. (2009), "A variable neighborhood search heuristic for periodic routing problems", European Journal of Operational Research, Volume 195, Pages 791-802

Ho W., Ho S.T.G., Ji P., Lau W.C.H. (2008), "A hybrid genetic algorithm for the multi-depot vehicle routing problem", Engineering Applications of Artificial Intelligence, Volume 21, Pages 548-557

Hong L. (2012), "An improved LNS algorithm for real-time vehicle routing problem with time Windows", Computers & Operations Research, Volume 39, Pages 151-163

Imran A., Salhi S., Wassan A.N. (2009), "A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem", European Journal of Operational Research, Volume 197, Pages 509–518

Kirkpatrick S., Gelatt D.C., Vecchi P.M. (1983), "Optimization by Simulated Annealing", Science, New Series, Volume 220, Number 4598, Pages 671-680

Kytöjoki J., Nuortio T., Bräysy O., Gendreau M. (2007), "An efficient variable neighborhood search heuristic for very large scale vehicle routing problems", Computers & Operations Research, Volume 34, Pages 2743-2757

Laporte G. (1992), "The Vehicle Routing Problem: An overview of exact and approximate algorithms", European Journal of Operational Research, Volume 59, Pages 345-358

Larbi R., Alpan G., Penz B. (2009), "Scheduling Transshipment Operations In A Multiple Inbound And Outbound Door Crossdock", 978-1-4244-4136-5/09, Pages 227-232

Lee H.Y., Jung W.J., Lee M.K. (2006), "Vehicle routing scheduling for cross-docking in the supply chain", Computers & Industrial Engineering, Volume 51, Pages 247-256

Ley S., Elfayoumy S. (2007), "Cross Dock Scheduling Using Genetic Algorithms", Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation Jacksonville, FL, USA, June 20-23, 2007

Li P.Z., Low1 H.Y.M., Shakeri M., Lim G.Y. (2009), "Crossdocking planning and scheduling: Problems and algorithms", SIMTech technical reports, Volume 10, Number 3, Jul-Sep 2009

Li X., Tian P., Leung H.C.S. (2010), "Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm", Int. J. Production Economics, Volume 125, Pages 137-145

Li Y., Lim A., Rodrigues B. (2004), “Crossdocking: JIT Scheduling with Time Windows”, The Journal of the Operational Research Society, Volume 55, Number 12, Pages 1342- 1351

Liao C-J., Lin Y., Shih C.S. (2010), “Vehicle routing with cross-docking in the supply chain”, Expert Systems with Applications, Volume 37, Pages 6868–6873

Lim A., Ma H., Miao Z. (2006), “Truck Dock Assignment Problem with Time Windows and Capacity Constraint in Transshipment Network Through Crossdocks”, ICCSA 2006, LNCS 3982, Pages 688–697

Lin S-W., Lee Z-J., Ying K-C., Lee C-Y. (2009), “Applying hybrid meta-heuristics for capacitated vehicle routing problem”, Expert Systems with Applications, Volume 36, Pages 1505–1512

Lin S-W., Yu F.V., Lu C-C. (2011), “A simulated annealing heuristic for the truck and trailer routing problem with time windows”, Expert Systems with Applications, DOI:10.1016/j.eswa.2011.05.075

Marinakis Y., Migdalas A. (2007), “Annotated Bibliography in Vehicle Routing”, Operational Research. An International Journal, Volume 7, Number 1, Pages27-46

McWilliams L.D., Stanfiel M.P., Geiger D.C. (2005), “The parcel hub scheduling problem: A simulation-based solution approach”, Computers & Industrial Engineering, Volume 49, Pages 393–412

Miao Z., Lim A., Ma H. (2009), “Truck dock assignment problem with operational time constraint within crossdocks”, European Journal of Operational Research, Volume 192, Pages 105-115

Miller C.E., Tucker A.W., Zemlin R.A. (1960), “Integer programming formulation of traveling salesman problems”, Journal of the ACM (JACM), Volume 7, Issue 4, Pages 326 – 329

Novoa C., Storer R. (2009), “An approximate dynamic programming approach for the vehicle routing problem with stochastic demands”, European Journal of Operational Research, Volume 196, Pages 509–515

Ou J., Hsu N.R., Li C-L. (2010), “Scheduling Truck Arrivals at an Air Cargo Terminal”, Volume 19, Number 1, Pages 83-97

Polacek M., Hartl F.R., Doerner K. (2004), “A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with TimeWindows”, Journal of Heuristics, Volume 10, Pages 613–627

Sadykov R. (2009), “A polynomial algorithm for a simple scheduling problem at cross docking terminals”, INRIA Research Report, inria-00412519, Version 1-2, September 1, 2009

Sharabiani N.S. (2009), “Obtaining optimal and approximate solutions to the problem of scheduling inbound and outbound trucks in cross docking operations”, University of Boras, School of Engineering, Master Thesis, Nr 12/2009

Soltani R., Sadjadi J.S. (2010), “Scheduling trucks in cross-docking systems: A robust meta-heuristics Approach”, Transport. Res. Part E, DOI:10.1016/j.tre.2009.12.011

Song K., Chen F. (2007), “Scheduling Cross Docking Logistics Optimization Problem with Multiple Inbound Vehicles and One Outbound Vehicle”, Proceedings of the IEEE International Conference on Automation and Logistics August 18 - 21, 2007, Jinan, China

Tang O. (2004), “Simulated annealing in lot sizing problems”, Int. J. Production Economics, Volume 88, Pages 173–181

Tavakkoli-Moghaddam R., Safaei N., Kah O.M.M., Rabbani M. (2007), “A New Capacitated Vehicle Routing Problem with Split Service for Minimizing Fleet Cost by Simulated Annealing”, Journal of the Franklin Institute, Volume 344, Pages 406-425

The Computational Infrastructure for Operations Research Web Site. Retrieved, Jun 10, 2011 from http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/

Toth P., Vigo D. (2002), “Models, relaxations and exact approaches for the capacitated vehicle routing problem”, Discrete Applied Mathematics, Volume 123, Pages 487 – 512

Vahdani B., Soltani R., Zandieh M. (2010), “Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics”, Int J Adv Manuf Technol, Volume 46, Pages 769–783

Vahdani B., Zandieh M. (2010), “Scheduling trucks in cross-docking systems: Robust meta-heuristics”, Computers & Industrial Engineering, Volume 58, Pages 12-24

Wade C.A., Salhi S. (2002), “An investigation into a new class of vehicle routing problem with backhauls”, Omega, Volume 30, Pages 479-487

Wang J-F., Regan A. (2008), “Real-Time Trailer Scheduling for Crossdock Operations”, Transportation Journal, Pages 5-20

Yu W., Egbelu J.P. (2008), “Scheduling of inbound and outbound trucks in cross docking systems with temporary storage”, European Journal of Operational Research, Volume 184, Pages 377-396

Zachariadis E.E., Tarantilis D.C., Kiranoudis T.C. (2009), “A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service”, Expert Systems with Applications, Volume 36, Pages 1070-1081

Zhao Q-H., Chen S., Zang C-X. (2008), “Model and algorithm for inventory/routing decision in a three-echelon logistics system”, European Journal of Operational Research, Volume 191, Pages 623-635

Zhong Y., Cole H.M. (2005), “A vehicle routing problem with backhauls and time windows: a guided local search solution”, Transportation Research, Part E, Volume 41, Pages 131-144

APPENDIX A

COMPUTATIONAL RESULTS FOR THE FIRST SCENARIO

In Appendix A, computational results for both proposed variable neighborhood search and simulated annealing approaches are provided in detail for the selected CVRP literature instances where the end time of the first shift is exactly equals to the longest tour length of the best solutions. At this point the end time of the second shift is two times greater than the first one and similarly, the end time of the third shift is three times greater than the first one.

Table A - 1. Results for Christophides and Eilon (n22 - v4) Problem for the First Scenario

Christophides and Eilon (n22 - v4)	**VNS**			**SA**		
Initial Solution: 1125	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1062	2.23	9.93	1040	3.48	13.91
2nd Trial	1048	2.69	9.32	1002	4.03	15.59
3rd Trial	1048	2.50	7.79	1027	2.94	14.83
4th Trial	1048	2.19	7.79	1006	3.79	13.45
5th Trial	1048	3.11	9.32	1040	3.41	14.37
Minimum	**1048**	**2.19**	**7.79**	**1002**	**2.94**	**13.45**
Average	**1050.8**	**2.54**	**8.83**	**1023**	**3.53**	**14.43**
Maximum	**1062**	**3.11**	**9.93**	**1040**	**4.03**	**15.59**

Table A - 2. Results for Christophides and Eilon (n23 - v3) Problem for the First Scenario

Christophides and Eilon (n23 - v3)	**VNS**			**SA**		
Initial Solution: 1707	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1236	5.01	12.77	950	2.69	49.42
2nd Trial	1155	7.25	26.28	779	2.69	50.98
3rd Trial	1165	4.58	32.37	911	3.71	51.73
4th Trial	1283	5.62	14.59	831	3.46	52.08
5th Trial	1265	4.30	20.09	876	3.83	49.56
Minimum	**1155**	**4.30**	**12.77**	**779**	**2.69**	**49.42**
Average	**1220.8**	**5.35**	**21.22**	**869.4**	**3.27**	**50.75**
Maximum	**1283**	**7.25**	**32.37**	**950**	**3.83**	**52.08**

Table A - 3. Results for Christophides and Eilon (n30 - v3) Problem for the First Scenario

Christophides and Eilon (n30 - v3)	**VNS**			**SA**		
Initial Solution: 1611	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1294	7.30	13.15	1300	3.85	22.64
2nd Trial	1492	9.90	9.11	1295	4.40	22.34
3rd Trial	1491	2.90	10.46	1304	3.65	20.25
4th Trial	1475	5.63	13.9	1251	3.94	26.45
5th Trial	1498	4.26	13	1303	4.64	23.84
Minimum	**1294**	**2.90**	**9.11**	**1251**	**3.65**	**20.25**
Average	**1450**	**6.00**	**11.62**	**1290.6**	**4.10**	**23.10**
Maximum	**1498**	**9.90**	**13.9**	**1304**	**4.64**	**26.45**

Table A - 4 . Results for Augerat, et al. (n37 - v5) Problem for the First Scenario

Augerat, et al. (n37 - v5)	**VNS**			**SA**		
Initial Solution: 2007	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1833	10.47	9.25	1615	9.19	29.07
2nd Trial	1880	11.04	6.96	1605	10.57	31.61
3rd Trial	1841	9.64	11.3	1532	8.40	26.53
4th Trial	1815	5.97	13.59	1557	8.68	30.63
5th Trial	1850	11.82	9.5	1573	8.94	28.66
Minimum	**1815**	**5.97**	**6.96**	**1532**	**8.40**	**26.53**
Average	**1843.8**	**9.79**	**10.12**	**1576.4**	**9.16**	**29.3**
Maximum	**1880**	**11.82**	**13.59**	**1615**	**10.57**	**31.61**

Table A - 5. Results for Augerat, et al. (n40 - v5) Problem for the First Scenario

Augerat, et al. (n40 - v5)	**VNS**			**SA**		
Initial Solution: 1374	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1345	6.36	4.36	1260	14.58	9.06
2nd Trial	1353	8.22	5.82	1284	11.90	11
3rd Trial	1298	11.83	4.63	1282	14.90	9.87
4th Trial	1350	8.31	5.82	1236	12.85	8.95
5th Trial	1345	7.96	6.31	1268	14.70	11.21
Minimum	**1298**	**6.36**	**4.36**	**1236**	**11.90**	**8.95**
Average	**1338.2**	**8.54**	**5.39**	**1266**	**13.78**	**10.02**
Maximum	**1353**	**11.83**	**6.31**	**1284**	**14.90**	**11.21**

Table A - 6. Results for Augerat, et al. (n45 - v7) Problem for the First Scenario

Augerat, et al. (n45 - v7)	**VNS**			**SA**		
Initial Solution: 3438	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	3253	10.51	8.2	3177	14.91	14.72
2nd Trial	3241	12.24	10.25	3307	13.53	10.72
3rd Trial	3254	9.01	8.93	3302	14.35	13.98
4th Trial	3252	13.72	9.25	3213	17.51	17.77
5th Trial	3254	9.96	9.35	3212	15.65	15.72
Minimum	**3241**	**9.01**	**8.2**	**3177**	**13.53**	**10.72**
Average	**3250.8**	**11.09**	**9.20**	**3242.2**	**15.19**	**14.58**
Maximum	**3254**	**13.72**	**10.25**	**3307**	**17.51**	**17.77**

Table A - 7. Results for Augerat, et al. (n65 - v10) Problem for the First Scenario

Augerat, et al. (n65 - v10)	**VNS**			**SA**		
Initial Solution: 2376	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	2319	11.52	4.78	2355	31.94	9.65
2nd Trial	2320	15.79	5.05	2240	66.79	9.07
3rd Trial	2322	12.17	4.94	2256	74.91	8.53
4th Trial	2289	22.42	6.61	2306	52.54	7.87
5th Trial	2311	28.33	6.17	2309	54.96	7
Minimum	**2289**	**11.52**	**4.78**	**2240**	**31.94**	**7**
Average	**2312.2**	**18.04**	**5.51**	**2293.2**	**56.23**	**8.42**
Maximum	**2322**	**28.33**	**6.61**	**2355**	**74.91**	**9.65**

Table A - 8. Results for Augerat, et al. (n78 - v10) Problem for the First Scenario

Augerat, et al. (n78 - v10)	**VNS**			**SA**		
Initial Solution: 3663	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	3578	38.14	4.51	3495	40.74	21.48
2nd Trial	3578	34.05	2.17	3478	37.50	24.19
3rd Trial	3582	42.66	3.34	3464	40.86	23.97
4th Trial	3574	71.69	3.59	3491	39.51	22.58
5th Trial	3556	88.61	5.76	3565	37.22	21.8
Minimum	**3556**	**34.05**	**2.17**	**3464**	**37.22**	**21.48**
Average	**3573.6**	**55.03**	**3.87**	**3498.6**	**39.16**	**22.80**
Maximum	**3582**	**88.61**	**5.76**	**3565**	**40.86**	**24.19**

Table A - 9. Results for Christophides and Eilon (n101 - v8) Problem for the First Scenario

Christophides and Eilon (n101 - v8)	**VNS**			**SA**		
Initial Solution: 2445	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	2392	84.84	3.42	2315	80.93	15.34
2nd Trial	2368	46.32	4.22	2399	82.18	14.24
3rd Trial	2361	38.15	5.87	2351	86.62	15.16
4th Trial	2384	29.00	4.54	2387	82.96	14.56
5th Trial	2338	86.47	5.89	2399	82.80	14.34
Minimum	**2338**	**29.00**	**3.42**	**2315**	**80.93**	**14.24**
Average	**2368.6**	**56.95**	**4.79**	**2370.2**	**83.10**	**14.73**
Maximum	**2392**	**86.47**	**5.89**	**2399**	**86.62**	**15.34**

APPENDIX B

COMPUTATIONAL RESULTS FOR THE SECOND SCENARIO

In Appendix B, computational results for both proposed variable neighborhood search and simulated annealing approaches are provided in detail for the selected CVRP literature instances where the end time of the first shift is 1.2 times that is 20% greater than the longest tour length of the best solutions. At this point the end time of the second shift is 2.4 times greater than the first one and similarly, the end time of the third shift is 3.6 times greater than the first one.

Table B - 1. Results for Christophides and Eilon (n22 - v4) Problem for the Second Scenario

Christophides and Eilon (n22 - v4)	**VNS**			**SA**		
Initial Solution: 1125	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1029	3.72	10.39	931	2.79	20.64
2nd Trial	1038	4.47	11.92	955	2.40	23.24
3rd Trial	1038	3.39	11.92	958	2.54	20.79
4th Trial	1048	3.70	9.32	930	2.79	19.87
5th Trial	1038	3.40	11.31	974	2.62	20.03
Minimum	**1029**	**3.39**	**9.32**	**930**	**2.40**	**19.87**
Average	**1038.2**	**3.74**	**10.97**	**949.6**	**2.63**	**20.91**
Maximum	**1048**	**4.47**	**11.92**	**974**	**2.79**	**23.24**

Table B - 2. Results for Christophides and Eilon (n23 - v3) Problem for the Second Scenario

Christophides and Eilon (n23 - v3)	**VNS**			**SA**		
Initial Solution: 1707	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	892	13.18	45.36	768	3.04	51.84
2nd Trial	1104	7.90	30.23	858	3.00	43.51
3rd Trial	905	11.71	35.31	790	2.89	51.84
4th Trial	986	17.74	37.29	922	2.86	51.21
5th Trial	984	7.24	47.36	829	2.99	40.79
Minimum	**892**	**7.24**	**30.23**	**768**	**2.86**	**40.79**
Average	**974.2**	**11.55**	**39.11**	**833.4**	**2.96**	**47.84**
Maximum	**1104**	**17.74**	**47.36**	**922**	**3.04**	**51.84**

Table B - 3. Results for Christophides and Eilon (n30 - v3) Problem for the Second Scenario

Christophides and Eilon (n30 - v3)	**VNS**			**SA**		
Initial Solution: 1611	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1076	10.86	22.57	953	4.32	30.11
2nd Trial	1199	9.19	17.71	1028	5.00	28.62
3rd Trial	1029	24.25	29.89	992	4.78	31.53
4th Trial	1248	10.25	17.71	987	4.44	33.63
5th Trial	1147	6.82	22.72	1000	4.61	28.99
Minimum	**1029**	**6.82**	**17.71**	**953**	**4.32**	**28.62**
Average	**1139.8**	**12.27**	**22.12**	**992**	**4.63**	**30.58**
Maximum	**1248**	**24.25**	**29.89**	**1028**	**5.00**	**33.63**

Table B - 4. Results for Augerat, et al. (n37 - v5) Problem for the Second Scenario

Augerat, et al. (n37 - v5)	**VNS**			**SA**		
Initial Solution: 2007	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1859	8.65	11.3	1513	7.83	32.84
2nd Trial	1825	14.34	8.68	1538	7.60	35.7
3rd Trial	1792	37.58	11.05	1428	7.67	32.18
4th Trial	1843	11.40	12.53	1436	7.99	31.44
5th Trial	1814	19.66	10.89	1589	7.81	32.51
Minimum	**1792**	**8.65**	**8.68**	**1428**	**7.60**	**31.44**
Average	**1826.6**	**18.32**	**10.89**	**1500.8**	**7.78**	**32.93**
Maximum	**1859**	**37.58**	**12.53**	**1589**	**7.99**	**35.7**

Table B - 5. Results for Augerat, et al. (n40 - v5) Problem for the Second Scenario

Augerat, et al. (n40 - v5)	**VNS**			**SA**		
Initial Solution: 1374	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1317	7.04	5.39	1272	8.54	14.56
2nd Trial	1338	6.95	4.26	1255	7.63	15.96
3rd Trial	1315	6.33	4.58	1261	7.91	14.34
4th Trial	1324	7.03	4.31	1282	7.95	12.94
5th Trial	1323	6.65	4.58	1209	8.85	17.69
Minimum	**1315**	**6.33**	**4.26**	**1209**	**7.63**	**12.94**
Average	**1323.4**	**6.80**	**4.62**	**1255.8**	**8.17**	**15.10**
Maximum	**1338**	**7.04**	**5.39**	**1282**	**8.85**	**17.69**

Table B - 6. Results for Augerat, et al. (n45 - v7) Problem for the Second Scenario

Augerat, et al. (n45 - v7)	**VNS**			**SA**		
Initial Solution: 3438	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	3253	26.01	8.09	3198	12.87	22.55
2nd Trial	3260	15.71	7.78	3131	12.22	21.18
3rd Trial	3256	19.58	9.09	3165	16.89	26.65
4th Trial	3255	19.75	9.88	3121	15.27	23.71
5th Trial	3247	16.37	8.25	3118	14.04	25.44
Minimum	**3247**	**15.71**	**7.78**	**3118**	**12.22**	**21.18**
Average	**3254.2**	**19.48**	**8.62**	**3146.6**	**14.26**	**23.91**
Maximum	**3260**	**26.01**	**9.88**	**3198**	**16.89**	**26.65**

Table B - 7. Results for Augerat, et al. (n65 - v10) Problem for the Second Scenario

Augerat, et al. (n65 - v10) Initial Solution: 2376	**VNS**			**SA**		
	fheuristic	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	2276	31.12	7.3	2280	32.97	11.94
2nd Trial	2320	13.01	4.92	2273	29.49	13.61
3rd Trial	2311	27.30	6.75	2241	36.10	17.33
4th Trial	2263	26.64	9.87	2292	42.79	7.35
5th Trial	2264	24.19	6.94	2259	32.76	12.19
Minimum	**2263**	**13.01**	**4.92**	**2241**	**29.49**	**7.35**
Average	**2286.8**	**24.45**	**7.16**	**2269**	**34.82**	**12.48**
Maximum	**2320**	**31.12**	**9.87**	**2292**	**42.79**	**17.33**

Table B - 8. Results for Augerat, et al. (n78 - v10) Problem for the Second Scenario

Augerat, et al. (n78 - v10) Initial Solution: 3663	**VNS**			**SA**		
	fheuristic	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	3531	165.94	8.78	3512	32.41	25.68
2nd Trial	3550	69.22	7.54	3502	37.46	26.68
3rd Trial	3534	116.87	8.57	3537	33.77	26.78
4th Trial	3526	161.31	8.92	3541	38.77	27.1
5th Trial	3516	131.99	10.13	3394	33.58	26.21
Minimum	**3516**	**69.22**	**7.54**	**3394**	**32.41**	**25.68**
Average	**3531.4**	**129.06**	**8.79**	**3497.2**	**35.20**	**26.49**
Maximum	**3550**	**165.94**	**10.13**	**3541**	**38.77**	**27.1**

Table B - 9. Results for Christophides and Eilon (n101 - v8) Problem for the Second Scenario

Christophides and Eilon (n101 - v8) **Initial Solution: 2445**	**VNS**			**SA**		
	fheuristic	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	2428	91.28	5.66	2283	78.80	22.35
2nd Trial	2297	84.26	5.73	2373	88.76	22.61
3rd Trial	2304	54.17	6.81	2426	82.44	20.78
4th Trial	2355	56.92	6.14	2306	89.44	24.87
5th Trial	2277	116.80	8.25	2241	79.41	24.65
Minimum	**2277**	**54.17**	**5.66**	**2241**	**78.80**	**20.78**
Average	**2332.2**	**80.68**	**6.52**	**2325.8**	**83.77**	**23.05**
Maximum	**2428**	**116.80**	**8.25**	**2426**	**89.44**	**24.87**

APPENDIX C

COMPUTATIONAL RESULTS FOR THE THIRD SCENARIO

In Appendix C, computational results for both proposed variable neighborhood search and simulated annealing approaches are provided in detail for the selected CVRP literature instances where the end time of the first shift is 1.4 times that is 40% greater than the longest tour length of the best solutions. At this point the end time of the second shift is 2.8 times greater than the first one and similarly, the end time of the third shift is 4.2 times greater than the first one.

Table C - 1. Results for Christophides and Eilon (n22 - v4) Problem for the Third Scenario

Christophides and Eilon (n22 - v4) Initial Solution: 1125	**VNS**			**SA**		
	fheuristic	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1025	5.44	8.56	853	2.27	23.7
2nd Trial	885	7.66	22.78	820	2.63	26.6
3rd Trial	925	8.37	16.97	854	2.88	26.6
4th Trial	981	4.71	14.37	847	2.62	27.98
5th Trial	979	7.35	14.83	866	2.70	26.75
Minimum	**885**	**4.71**	**8.56**	**820**	**2.27**	**23.7**
Average	**959**	**6.70**	**15.50**	**848**	**2.62**	**26.33**
Maximum	**1025**	**8.37**	**22.78**	**866**	**2.88**	**27.98**

Table C - 2. Results for Christophides and Eilon (n23 - v3) Problem for the Third Scenario

Christophides and Eilon (n23 - v3) Initial Solution: 1707	**VNS**			**SA**		
	fheuristic	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	809	53.70	43.71	806	2.88	52.61
2nd Trial	1054	11.73	35.28	823	2.67	51.09
3rd Trial	962	14.04	44.18	808	3.20	51.42
4th Trial	905	10.39	41.65	810	3.00	52.33
5th Trial	889	32.09	40.89	812	2.79	51.56
Minimum	**809**	**10.39**	**35.28**	**806**	**2.67**	**51.09**
Average	**923.8**	**24.39**	**41.14**	**811.8**	**2.91**	**51.80**
Maximum	**1054**	**53.70**	**44.18**	**823**	**3.20**	**52.61**

Table C - 3. Results for Christophides and Eilon (n30 - v3) Problem for the Third Scenario

Christophides and Eilon (n30 - v3) Initial Solution: 1611	**VNS**			**SA**		
	fheuristic	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1040	49.60	30.11	961	4.79	34.6
2nd Trial	1076	25.51	31.01	980	5.19	30.64
3rd Trial	1074	7.65	27.13	982	4.96	34.23
4th Trial	1063	7.48	26.23	985	4.75	27.05
5th Trial	1009	12.81	27.27	980	5.43	27.57
Minimum	**1009**	**7.48**	**26.23**	**961**	**4.75**	**27.05**
Average	**1052.4**	**20.61**	**28.35**	**977.6**	**5.02**	**30.82**
Maximum	**1076**	**49.60**	**31.01**	**985**	**5.43**	**34.6**

Table C - 4. Results for Augerat, et al. (n37 - v5) Problem for the Third Scenario

Augerat, et al. (n37 - v5) Initial Solution: 2007	**VNS**			**SA**		
	fheuristic	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	1762	21.18	12.2	1339	7.20	38.65
2nd Trial	1582	30.86	20.47	1320	7.96	37.51
3rd Trial	1585	54.55	20.8	1311	7.92	36.69
4th Trial	1594	38.73	26.04	1333	7.82	39.23
5th Trial	1784	25.58	15.31	1322	7.59	40.86
Minimum	**1582**	**21.18**	**12.2**	**1311**	**7.20**	**36.69**
Average	**1661.4**	**34.18**	**18.96**	**1325**	**7.70**	**38.59**
Maximum	**1784**	**54.55**	**26.04**	**1339**	**7.96**	**40.86**

Table C - 5. Results for Augerat, et al. (n40 - v5) Problem for the Third Scenario

Augerat, et al. (n40 - v5)	VNS			SA		
Initial Solution: 1374	fheuristic	CPU (min)	% Early Collected	fheuristic	CPU (min)	% Early Collected
1st Trial	1304	8.35	8.46	1130	7.84	22.86
2nd Trial	1301	8.42	8.19	1121	7.83	25.08
3rd Trial	1272	10.53	9.33	1110	8.04	24.75
4th Trial	1171	14.42	13.53	1100	8.09	25.56
5th Trial	1302	6.43	6.79	1100	8.16	25.67
Minimum	**1171**	**6.43**	**6.79**	**1100**	**7.83**	**22.86**
Average	**1270**	**9.63**	**9.26**	**1112.2**	**7.99**	**24.78**
Maximum	**1304**	**14.42**	**13.53**	**1130**	**8.16**	**25.67**

Table C - 6. Results for Augerat, et al. (n45 - v7) Problem for the Third Scenario

Augerat, et al. (n45 - v7)	VNS			SA		
Initial Solution: 3438	fheuristic	CPU (min)	% Early Collected	fheuristic	CPU (min)	% Early Collected
1st Trial	3238	32.21	10.88	3088	13.76	28.49
2nd Trial	3145	49.86	15.61	3023	10.07	27.28
3rd Trial	3228	24.05	10.62	3138	11.45	24.55
4th Trial	3128	41.48	12.88	3108	12.65	27.97
5th Trial	3239	19.32	10.72	3131	11.27	27.54
Minimum	**3128**	**19.32**	**10.62**	**3023**	**10.07**	**24.55**
Average	**3195.6**	**33.38**	**12.14**	**3097.6**	**11.84**	**27.17**
Maximum	**3239**	**49.86**	**15.61**	**3138**	**13.76**	**28.49**

Table C - 7. Results for Augerat, et al. (n65 - v10) Problem for the Third Scenario

Augerat, et al. (n65 - v10)	**VNS**			**SA**		
Initial Solution: 2376	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	2212	57.20	9.43	2165	30.88	23.48
2nd Trial	2263	20.37	7.95	2242	28.17	24.28
3rd Trial	2292	20.95	7.79	2102	34.28	24.52
4th Trial	2269	49.79	8.09	2166	30.55	23.89
5th Trial	2255	22.86	9.7	2078	29.59	24.03
Minimum	**2212**	**20.37**	**7.79**	**2078**	**28.17**	**23.48**
Average	**2258.2**	**34.23**	**8.59**	**2150.6**	**30.69**	**24.04**
Maximum	**2292**	**57.20**	**9.7**	**2242**	**34.28**	**24.52**

Table C - 8. Results for Augerat, et al. (n78 - v10) Problem for the Third Scenario

Augerat, et al. (n78 - v10)	**VNS**			**SA**		
Initial Solution: 3663	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	3527	151.87	9.78	3537	32.80	29.42
2nd Trial	3538	164.64	8.18	3412	36.52	31.98
3rd Trial	3523	156.90	10.6	3589	31.09	31.55
4th Trial	3530	267.94	12.05	3409	31.75	28.92
5th Trial	3537	131.09	7.54	3528	32.69	29.09
Minimum	**3523**	**131.09**	**7.54**	**3409**	**31.09**	**28.92**
Average	**3531**	**174.49**	**9.63**	**3495**	**32.97**	**30.19**
Maximum	**3538**	**267.94**	**12.05**	**3589**	**36.52**	**31.98**

Table C - 9. Results for Christophides and Eilon (n101 - v8) Problem for the Third Scenario

Christophides and Eilon (n101 - v8)	**VNS**			**SA**		
Initial Solution: 2445	**fheuristic**	**CPU (min)**	**% Early Collected**	**fheuristic**	**CPU (min)**	**% Early Collected**
1st Trial	2279	74.80	8.87	2392	84.23	25.6
2nd Trial	2286	48.58	5.62	2373	84.31	24.89
3rd Trial	2277	115.33	9.05	2195	83.48	28.32
4th Trial	2395	180.29	7.54	2259	82.05	26.42
5th Trial	2358	63.66	7.99	2305	81.01	26.02
Minimum	**2277**	**48.58**	**5.62**	**2195**	**81.01**	**24.89**
Average	**2319**	**96.53**	**7.81**	**2304.8**	**83.01**	**26.25**
Maximum	**2395**	**180.29**	**9.05**	**2392**	**84.31**	**28.32**

APPENDIX D

COMPUTATIONAL RESULTS FOR THE 2-opt* APPROACH USING SECOND SCENARIO

In Appendix D, computational results for proposed variable neighborhood search algorithm with 2-opt* approach in the local search step, are provided in detail for the selected CVRP literature instances where the end time of the first shift is 1.2 times that is 20% greater than the longest tour length of the best solutions. Here, the end time of the second shift is 2.4 times greater than the first one and similarly, the end time of the third shift is 3.6 times greater than the first one.

Table D - 1. Results for Christophides and Eilon (n22 - v4) Problem for the Second Scenario

Christophides and Eilon (n22 - v4) Initial Solution: 1125	VNS		% Early Collected
	fheuristic	CPU (min)	
1st Trial	857	70.66	21.86
2nd Trial	920	26.17	19.72
3rd Trial	846	63.92	22.47
Minimum	**846**	**26.17**	**19.72**
Average	**874.33**	**53.58**	**21.35**
Maximum	**920**	**70.66**	**22.47**

Table D - 2. Results for Christophides and Eilon (n23 - v3) Problem for the Second Scenario

Christophides and Eilon (n23 - v3) Initial Solution: 1707	VNS		% Early Collected
	fheuristic	CPU (min)	
1st Trial	745	789.82	29.57
2nd Trial	745	666.31	49.53
3rd Trial	769	517.04	29.57
Minimum	**745**	**517.04**	**29.57**
Average	**753**	**657.72**	**36.22**
Maximum	**769**	**789.82**	**49.53**

Table D - 3. Results for Christophides and Eilon (n30 - v3) Problem for the Second Scenario

Christophides and Eilon (n30 - v3) Initial Solution: 1611	**VNS**		**% Early Collected**
	fheuristic	**CPU (min)**	
1st Trial	972	210.61	31.01
2nd Trial	990	494.68	24.81
3rd Trial	955	82.96	32.36
Minimum	**955**	**82.96**	**24.81**
Average	**972.33**	**262.75**	**29.39**
Maximum	**990**	**494.68**	**32.36**

Table D - 4. Results for Augerat, et al. (n37 - v5) Problem for the Second Scenario

Augerat, et al. (n37 - v5) Initial Solution: 2007	**VNS**		**% Early Collected**
	fheuristic	**CPU (min)**	
1st Trial	1344	848.12	31.77
2nd Trial	1396	999.17	29.97
3rd Trial	1395	935.89	26.53
Minimum	**1344**	**848.12**	**26.53**
Average	**1378.33**	**927.72**	**29.42**
Maximum	**1396**	**999.17**	**31.77**

Table D - 5. Results for Augerat, et al. (n40 - v5) Problem for the Second Scenario

Augerat, et al. (n40 - v5) Initial Solution: 1374	VNS		% Early Collected
	fheuristic	CPU (min)	
1st Trial	1105	94.52	19.25
2nd Trial	1121	88.26	18.98
3rd Trial	1123	127.22	17.79
Minimum	**1105**	**88.26**	**17.79**
Average	**1116.33**	**103.33**	**18.67**
Maximum	**1123**	**127.22**	**19.25**

Table D - 6. Results for Augerat, et al. (n45 - v7) Problem for the Second Scenario

Augerat, et al. (n45 - v7) Initial Solution: 3438	VNS		% Early Collected
	fheuristic	CPU (min)	
1st Trial	3100	792.21	21.34
2nd Trial	3073	1185.55	17.56
3rd Trial	3069	1449.99	18.76
Minimum	**3069**	**792.21**	**17.56**
Average	**3080.67**	**1142.58**	**19.22**
Maximum	**3100**	**1449.99**	**21.34**

Table D - 7. Results for Augerat, et al. (n65 - v10) Problem for the Second Scenario

Augerat, et al. (n65 - v10) Initial Solution: 2376	**VNS**		**% Early Collected**
	fheuristic	**CPU (min)**	
1st Trial	2236	337.46	10.09
2nd Trial	2210	305.07	9.1
3rd Trial	2216	412.71	9.76
Minimum	**2210**	**305.07**	**9.1**
Average	**2220.67**	**351.75**	**9.65**
Maximum	**2236**	**412.71**	**10.09**

Table D - 8. Results for Augerat, et al. (n78 - v10) Problem for the Second Scenario

Augerat, et al. (n78 - v10)* Initial Solution: 3663	**VNS**		**% Early Collected**
	fheuristic	**CPU (min)**	
1st Trial	3401	606.8	18.32
2nd Trial	3365	601.23	21.45
3rd Trial	3373	608.45	22.09
Minimum	**3365**	**601.23**	**18.32**
Average	**3379.67**	**605.49**	**20.62**
Maximum	**3401**	**608.45**	**22.09**

Table D - 9. Results for Christophides and Eilon (n101 - v8) Problem for the Second Scenario

Christophides and Eilon (n101 - v8)* Initial Solution: 2445	VNS		% Early Collected
	fheuristic	CPU (min)	
1st Trial	2236	604.5	14.1
2nd Trial	2291	606.28	8.64
3rd Trial	2236	607.3	14.1
Minimum	**2236**	**604.5**	**8.64**
Average	**2254.33**	**606.03**	**12.28**
Maximum	**2291**	**607.3**	**14.1**

* - Since the algorithm could not terminate in 7200 minutes that is 120 hours, we report the results obtained when the algorithm terminated after 10 hours.