Integrated vehicle scheduling and routing policies for cross-dock systems

Agustina, Dwi

2016

Agustina, D. (2016). Integrated vehicle scheduling and routing policies for cross-dock systems. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/68868>

<https://doi.org/10.32657/10356/68868>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**INTEGRATED VEHICLE SCHEDULING AND ROUTING
POLICIES FOR CROSS-DOCK SYSTEMS**

DWI AGUSTINA

**SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY**

2016

**INTEGRATED VEHICLE SCHEDULING AND ROUTING
POLICIES FOR CROSS-DOCK SYSTEMS**

DWI AGUSTINA

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

A thesis submitted to the
Nanyang Technological University
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

June 2016

ABSTRACT

Cross-docking has become an increasingly popular logistics strategy in which products from several suppliers are consolidated into a single shipment to enable FTL, rather than LTL, shipments. The cross-dock warehouse ships the received goods directly to the customers, with little or no storage in between. The advantages of implementing cross-docking include shortened delivery time, improved customer satisfaction, and increased cost savings. Cross-docking depends heavily on the use of information technology and sophisticated planning software that coordinate the physical product flows and information flows. However, unavailability of effective IT systems is one of the biggest barriers faced by companies implementing cross-docking. In addition, contemporary research only considers vehicle scheduling, product consolidation and routing issues separately, leading to sub-optimal solutions.

This research takes a step towards integrating operations planning issues at a cross-dock, and develops vehicle scheduling and routing policies at a cross dock warehouse using an integrated model. There are four key contributions of this research.

- (1) First is a comprehensive literature review of cross-dock operations planning issues. Extensive literature review in cross-docking is done and research gap identified. The main research gap identified and addressed in this research is the integration of scheduling and vehicle routing policies at cross docks. This research focuses on delivery vehicle scheduling and routing problem at a cross-dock facility by combining these two problems, and considering the customer delivery time window and the product consolidation decision.
- (2) Second contribution of this research is a model for vehicle routing and scheduling problems at a cross-dock, namely VRSP, and modelled as a MILP. The model consists of suppliers, inbound trucks, cross-dock facility, outbound trucks, and customers, and tries to determine the vehicle routing and scheduling policies jointly. The objective of VRSP

is to determine the schedule of inbound and outbound vehicles at the cross-dock and the routing of outbound vehicles to deliver products within customer specified time windows. VRSP aims to minimize the total cost, which consists of the earliness penalty cost, the tardiness penalty cost, the inventory holding cost, and the transportation cost of deliveries. The performance of VRSP is experimented using CPLEX Solver version 12.3 running on an Intel® Celeron® CPU N2840 2.16 GHz computer. The experiments show that CPLEX Solver can solve the VRSP in a reasonable and acceptable time only for small scale problems. The CPLEX Solver takes a long time to solve medium size problems, and is unable to solve large-scale, real-life size problems.

- (3) The third contribution of this research is a modified model incorporating customer zones and hard-time windows to eliminate late and early delivery costs and substantially reduce the solution space. Through experiments using the same software and computer platform as VRSP, it is shown that the modified model (VRSP-CZHTW) can be solved for medium to large scale, real-life problems within an acceptable time. Only when the number of customers reaches five hundred or higher does the Solver reach its time limit without a solution.
- (4) The fourth and final contribution of this research is a meta-heuristics (TS algorithm) based approach to solve the original VRSP. Further experiments are done to validate and test the TSA-VRSP on an Intel® Celeron® CPU N2840 2.16 GHz computer. The algorithm is coded using MATLAB version 7.9.0. The experiments show that TSA-VRSP can be solved even for large scale problems in an acceptable time. The experiments also show that TSA-VRSP has better performance than the two previously developed methods (VRSP and VRSP-CZHTW, using CPLEX Solver).

Thus, TSA-VRSP has the potential to be the foundation of a decision support tool for cross-dock managers, and help develop effective and integrated vehicle routing and scheduling policies for day-to-day operations.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my supervisor, Assoc. Prof. Rajesh Piplani, for his patience, invaluable advice, helpful guidance, and encouragement throughout the course of my study. Without his advice and patience, this work would not have been possible.

I would also like to express my appreciation to my former supervisor, Asst. Prof. Lee Ka Man, Carman for her patience, advice, guidance, and encouragement when I became her student. Many thanks are also due to my friends in the research group, Lv Yaqiong, Lin Danping, Hong Zhen, Hassan Mirzahosseini, Ang Wei Hern Alvin, and Wang Feng Yu for the friendship and support given during my study.

I would like to acknowledge the financial support received from Nanyang Technological University (NTU) and ASEAN University Network/South East Engineering Development-Japan International Cooperation Agency (AUN/Seed-Net JICA) which enabled me to carry out this study as full-time PhD graduate student.

Many thanks are also due to Azhar, Fitri, Bowo and Feni, Djati and Muti, Pak Kus and Teh Evi, Tono and Audi for our friendship and support during my PhD.

Finally, special thanks are reserved for my husband, my mother and my father, my children (Hakim, Karim, and Mutia), and all my big families for their faith in me and encouragement through the years; to them this work is humbly dedicated.

TABLE OF CONTENTS

ABSTRACT.....	I
ACKNOWLEDGMENTS.....	III
TABLE OF CONTENTS.....	IV
LIST OF FIGURES.....	VIII
LIST OF TABLES.....	IX
PUBLICATIONS.....	XI
CHAPTER 1 INTRODUCTION.....	1
1.1 RESEARCH BACKGROUND.....	1
1.2 MOTIVATION OF RESEARCH.....	3
1.3 RESEARCH PROBLEMS.....	6
1.4 OBJECTIVES OF RESEARCH.....	9
1.5 RESEARCH SCOPE.....	10
1.6 REPORT OUTLINE.....	11
CHAPTER 2 LITERATURE REVIEW.....	13
2.1 CROSS-DOCK PLANNING MODEL.....	13
2.1.1 Strategic Level	14
2.1.1.1 Cross-dock Network Design.....	15
2.1.2 Tactical Level.....	19
2.1.2.1 Cross-dock Layout Design.....	20
2.1.3 Operational Level.....	21

TABLE OF CONTENTS

2.1.3.1 Cross-dock Scheduling.....	22
2.1.3.2 Dock-door Assignment.....	30
2.1.3.3 Trans-shipment.....	33
2.1.3.4 Vehicle Routing Problem.....	34
2.1.3.5 Product Allocation Problem.....	41
2.2 VEHICLE ROUTING AND SCHEDULING WITH TIME WINDOWS.....	41
2.3 TABU SEARCH.....	46
2.4 SUMMARY OF MODELLING AND SOLUTION APPROACHES.....	50
2.5 RESEARCH GAP.....	50
CHAPTER 3 VEHICLE ROUTING AND SCHEDULING AT A CROSS-DOCK	53
3.1 MODEL DESCRIPTION.....	53
3.2 MODEL DEVELOPMENT.....	54
3.3 MODEL FORMULATION.....	57
3.4 TEST DATA SET GENERATION.....	70
3.4.1 The Data Sets	70
3.4.2 The Problem Parameter.....	72
3.5 NUMERICAL EXPERIMENT.....	75
3.5.1 CPLEX Parameters.....	76
3.5.2 Example Application of VRSP.....	76
3.5.3 Experiment Investigation.....	78
3.6 SUMMARY OF VRSP MODEL.....	81
CHAPTER 4 VEHICLE ROUTING AND SCHEDULING PROBLEM WITH CUSTOMER ZONES AND HARD TIME WINDOWS.....	83

4.1 THE SIMPLIFICATION STRATEGY	84
4.1.1 Determination of the Customer Zones.....	87
4.2 VRSP-CZHTW MODEL.....	88
4.3 VRSP-CZHTW MODEL FORMULATION.....	93
4.4 VRSP-CZHTW NUMERICAL EXPERIMENTS.....	98
4.4.1 Example Application of VRSP-CZHTW.....	98
4.4.2 Experiment Results of VRSP-CZHTW.....	100
4.5 SUMMARY OF VRSP-CZHTW MODEL.....	104
CHAPTER 5 TABU SEARCH BASED ALGORITHM FOR VEHICLE ROUTING AND SCHEDULING.....	106
5.1 TABU SEARCH.....	107
5.1.1 Basic TS Algorithm.....	107
5.1.2 The TS Algorithm Selection.....	110
5.2 DEVELOPMENT OF TABU-SEARCH BASED ALGORITHM FOR VRSP	112
5.2.1 Initial Phase.....	113
5.2.2 Neighborhood Generation Mechanism	115
5.2.3 Neighborhood Selection	117
5.2.4 Stopping Criteria	119
5.3 THE FRAMEWORK OF TSA-VRSP	119
5.4 TSA-VRSP PROCEDURE.....	125
5.5 EXPERIMENTS WITH TSA-VRSP	130
5.5.1 TS Parameter Setting.....	131

5.5.2 Experiment Result of TSA-VRSP.....	131
5.6 COMPARISON OF TSA-VRSP WITH VRSP AND VRSP-CZHTW.....	133
5.7 SUMMARY OF TSA-VRSP.....	137
CHAPTER 6 CONCLUSIONS AND FUTURE WORK.....	139
6.1 CONCLUSIONS.....	139
6.2 FUTURE WORK.....	145
REFERENCES.....	147
APPENDIX-A.....	163
APPENDIX-B.....	173

LIST OF FIGURES

Figure 1.1 The cross-dock system under study.....	11
Figure 2.1 Classification of cross-dock planning problem.....	15
Figure 3.1 Cross-dock process-flow.....	57
Figure 3.2 Decision variables related to cross-dock yard and inbound area..	60
Figure 3.3 Decision variables related to outbound vehicle and customer location.....	62
Figure 3.4 Decision variables of VRSP model.....	62
Figure 4.1 Schema for VRSP-CZHTW.....	84
Figure 4.2 Effect of time window on determination of customer clustering	88
Figure 5.1 Simple tabu search.....	109
Figure 5.2 Adjacent pairwise interchange mechanism	116
Figure 5.3 TSA-VRSP workflow.....	122
Figure 5.4 The framework of TSA-VRSP.....	124
Figure 5.5 TSA-VRSP main program	129
Figure 5.6 TTC calculation routine	130
Figure B.1 The pseudo-code of TSA-VRSP	182
Figure B.2 The pseudo-code of TTC calculation routine in TSA-VRSP....	182

LIST OF TABLES

Table 1.1 Greatest barriers of cross-docking.....	6
Table 2.1 Cross-dock scheduling models.....	29
Table 2.2 Vehicle routing models and their variants in logistics.....	45
Table 2.3 Tabu Search-based algorithm in VRP.....	47
Table 2.4 Tabu Search-based algorithm in scheduling problems.....	49
Table 3.1 Small size data sets	71
Table 3.2 Medium size data sets.....	72
Table 3.3 Large size data sets.....	72
Table 3.4 The problem parameter values	75
Table 3.5 Experiment result of VRSP for small data sets.....	79
Table 3.6 Experiment result of VRSP for medium data sets.....	79
Table 3.7 Experiment result of VRSP for large data sets	80
Table 3.8 Experiment to determine the optimum number of outbound trucks.....	82
Table 4.1 The elapsed run time for K-Means clustering with MATLAB.....	101
Table 4.2 Experiment result of VRSP-CZHTW with CPLEX Solver (small data sets).....	102
Table 4.3 Experiment result of VRSP-CZHTW with CPLEX Solver (medium data sets).....	103
Table 4.4 Experiment result of VRSP-CZHTW with CPLEX Solver (large data sets).....	103
Table 5.1 Example of travel time of 4 customers (in minutes)	114
Table 5.2 TSA-VRSP on small dataset.....	132
Table 5.3 TSA-VRSP on medium dataset	132
Table 5.4 TSA-VRSP on large dataset	133
Table 5.5 Comparison of VRSP, VRSP-CZHTW, and TSA-VRSP for small datasets	134

Table 5.6 Comparison of VRSP, VRSP-CZHTW, and TSA-VRSP for medium datasets	135
Table 5.7 Comparison of VRSP, VRSP-CZHTW, and TSA-VRSP for large datasets	136

PUBLICATIONS

1. Agustina, D., Lee, C.K.M., and Piplani, R., 2010. A review: mathematical models for cross docking planning. *International Journal of Engineering Business Management*. Vol. 2, No. 2, pp. 47-54.
2. Agustina, D., Lee, C.K.M., and Piplani, R., 2011. Cross docking: scheduling with delivery time window and temporary storage. *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 6-9 December, Singapore, pp. 131-135.
3. Agustina, D., Lee, C.K.M., and Piplani, R., 2012. Scheduling and vehicle routing model of cross docking. *2012 International Conference on Future Information Technology and Management Science & Engineering (FITMSE2012)*. Lecture Notes in Information Technology Vol.14 IERI, pp. 373-378.
4. Piplani, R., Agustina, D., and Lee, C.K.M., 2013. Routing and scheduling at a cross-dock warehouse. *INFORMS Annual Meeting 2013 Minneapolis*. October 6-9, USA.
5. Agustina, D., Lee, C.K.M., and Piplani, R., 2014. Vehicle scheduling and routing at a cross docking center for food supply chains. *International Journal of Production Economics*. Vol. 152, pp. 29-41.
6. Piplani, R. and Agustina, D., 2014. Vehicle scheduling and routing at a cross-docking warehouse: a metaheuristic-based approach. *The 6th International Conference on Operations and Supply Chain Management (OSCM)*. 10 – 13 December 2014, Bali, Indonesia.

7. Piplani, R. and Agustina, D., 2016. Scheduling and routing problem at a cross-dock warehouse: application of tabu search. *Forty-Fifth Annual Meeting Western Decision Sciences Institute (WDSI 2016)*. April 5 – 9. Las Vegas, Nevada.

CHAPTER 1

INTRODUCTION

1.1 RESEARCH BACKGROUND

One of the contemporary strategies to improve the efficiency of distribution is cross-dock warehousing. In a traditional warehouse the following four main activities are performed: receiving, storing, picking, and shipping; the cross-dock warehouse eliminates the storing and picking activities, which are the two most expensive activities (Schaffer, 1998). The cross-dock warehouse ships the received goods directly to the customers with little or no storage in between so that the inventory holding cost can be reduced (Schaffer, 1998; Apte and Viswanathan, 2004).

Cross-docking has been known as the logistics strategy in which products from several suppliers can be consolidated into a single shipment. Cross-docking can also minimize (eliminate) order picking and storage activity, since the receiving products are moved immediately, or within 24 hours, to the outbound dock door (Rohrer, 1995; Schwind, 1996; Schaffer, 1998; Witt, 1998).

In a traditional warehouse, the received products are put away for storage into the storage area. Then when a customer order arrives, the products are picked and shipped to the customer. Compared with the traditional warehouse, in a cross-dock warehouse, it is possible to consolidate several products from different

suppliers. Cross-docking makes it possible to ship products in Full Truck Load (FTL) rather than Less Than Truck Load (LTL). Therefore, applying cross-docking in a distribution network can reduce transportation and inventory holding cost, shorten the delivery lead time and increase customer satisfaction (Apte and Viswanathan, 2004).

Due to shorter product life cycles, many companies prefer to use cross-docking, rather than traditional warehousing, to distribute their products, as reported by Saddle Creek Corp., an independent company commissioned to survey professionals in warehousing, distribution, and transportation (Saddle Creek Corp., 2008 and 2011). From the 2008 survey, Saddle Creek found that 52% of respondents are currently using cross-dock, 13% do not cross-dock now but plan to cross-dock in the next 18 to 24 months, and 31% have no plans to cross-dock (Saddle Creek Corp., 2008). In the follow-up survey, in 2011, they found that the number of respondents using cross-docking had increased to 68.5%, 15.1% planned to cross-dock in the next 18 to 24 months and only 15.5% had no plans to cross-dock (Saddle Creek Corp., 2011). The higher percentage and the increasing trend of companies using cross-docking shows the important role of cross-docking in the logistics industry.

The main advantage of implementing cross-docking is that the products can be transferred more quickly to the customers, as compared to traditional warehousing. It can save distribution time, shorten the delivery cycle time, and improve customer satisfaction. Another advantage of cross-docking is cost savings compared to traditional warehousing. Cross-docking can help to reduce warehouse space cost, inventory cost, transportation cost, and labour cost. Cross-

docking can save in excess of \$1 Million per year for a typical warehouse or \$87,600 per month (Saddle Creek Corp., 2008). Thus, cross-docking plays an important role in product distribution and supply chain.

1.2 MOTIVATION OF RESEARCH

As mentioned in the previous section, there is an increasing trend of companies turning from traditional to cross-dock warehousing. Cross-docking can be applied by LTL trucking industry, retail and grocery industry, manufacturing, and parcel and delivery service industry.

As reported by Witt (1998), Toyota has used cross-docking to great effect. With the increased reliance on Just-in-Time (JIT), parts are being shipped at higher frequency and in lower quantity. By making use of cross-docking, Toyota has reduced distribution cost by consolidating smaller part supplies into consolidated loads. Cross-docking has allowed Toyota to increase JIT and remove waste. The increased speed in the supply chain also helps companies to reduce product cycle time and move product quickly and efficiently down the supply chain. In Toyota's case, this has allowed them to increase delivery frequency.

Witt (1998) also reported that Supervalu, a food distributor based in Minneapolis, is well suited for cross-docking. It built a new distribution centre in Alabama to serve its stores in the South eastern states. The cross-docking played an integral role in the operation of this centre. By using cross-docking, it was able to have the inbound trailers unloaded and the product on outbound trailers within 30 minutes.

JC Penney Corporation, Inc., a leading retail chain in the US, has also implemented cross-dock in its distribution channel. Using cross-docking, it can move 165,000 cases a day from the receiving dock to the shipping dock with 99.9% accuracy, thereby increasing throughput. The cross-docking implementation also reduces the material handling cost of JC Penney Corporation, Inc. (<http://www.mmh.com>). Other companies, such as DHL, WalMart, and Mitsubishi have also taken advantage of cross-docking and improved the efficiency of their supply chain.

Beside the advantages of implementing cross-docking, there are many other factors that should be taken into consideration in order to implement cross-docking successfully. A normal size cross-dock warehouse can manage about 10,000 stock keeping unit (SKU) and 100 trucks every day. Since many products and trucks are managed and operated in the facility in a day, an optimal schedule is a prerequisite to efficiently and effectively operate the cross-dock facility. By having an optimal schedule, the operation of the cross-dock can be managed to minimize cost and time, and satisfy the customer.

The cross-dock warehouse handles many inbound and outbound trucks, many incoming and outgoing products, the unloading and loading schedule, the assignment of inbound and outbound dock doors, and truck routing. Another factor that should be considered is the objective of cross-docking facility. In order to achieve the advantages, the cross-dock operation must result in reduction of the distribution cost, delivery cycle time, and penalty cost of earliness and tardiness.

Based on the literature, the common practice is to optimize the truck schedule and the vehicle routes separately. The current practice of scheduling trucks in the cross-dock is by using first-come-first-served (FCFS) rule. This policy has some disadvantages such as large number of early and tardy jobs and long waiting time (Gue, 1999; Bartholdi and Gue, 2000; Wei et al., 2009).

There is software available to control every aspect of warehouse operations (Warehouse Management System (WMS)) and software for transportation management systems (TMS). A WMS can control every aspect of warehouse operations and TMS Software enables effective and efficient management of transportation operations, including determination of the vehicles to be used in the distribution network. The TMS software is used for effective vehicle planning, scheduling and planning of delivery routes. The WMS and TMS software can be used together to develop strategies that can minimize costs related to warehousing and transportation.

Of course, communication and good information flow between suppliers, cross dock facility and customers is very important to enhance the supply chain performance (Schaffer, 1998; Saddle Creek Corp., 2008). From their surveys in 2008 and 2011, Saddle Creek Corp. in US found that IT support system is the second biggest barrier faced by companies implementing cross-docking.

Table 1.1 Greatest barriers of cross-docking (Saddle Creek Corp., 2011)

Barrier	Percentage
Unpredictable customer demand	35%
IT system support	28%
Business dynamic always changing	25%
Reliability of suppliers	14%
Reliability of carriers	13%
Facility design	6%
Achieving ROI	5%
Management support	3%
Lack of resources	3%
Other	2%
Don't know	2%

1.3 RESEARCH PROBLEMS

From the above research background and motivation, several research problems can be identified. These are:

(1) Complexity of scheduling large number of deliveries within time windows

The cross-docking center has to handle a large number of products and trucks every day. The number of inbound and outbound dock doors, the number of workers, the capacity of the staging area in cross-dock, and the number of handling equipment is limited. On the other hand, the manager at the cross-dock facility is responsible for an efficient schedule so the products can be delivered on time and within customer specified time windows. A good schedule can

minimize early and late deliveries. With the large number of products for trans-shipment but limited resources, it is difficult to schedule the deliveries while minimizing their earliness and tardiness.

(2) Trading-off FTL deliveries with LTL but on-time delivery

The manager of cross-dock faces the challenge of deciding whether the products are delivered in LTL within customer's time window or delivered in FTL but maybe outside of customer's delivery window. The transportation cost is minimized if the product is delivered in FTL. On time delivery results in no penalty cost but deliveries outside of time window generally result in earliness or tardiness penalty. Thus the manager has to find ways to minimize not only the transportation cost but also the penalty cost of earliness and tardiness.

(3) Complicated delivery routing problem

Apart from considering delivery in FTL or LTL, the manager of cross-dock facility also has to establish the delivery route. The product delivery quantities and the delivery route determine the transportation cost. To benefit from using cross-docking, it is not enough just to minimize the transportation cost. The manager of cross-dock also needs to consider the penalty cost of earliness and tardiness and the inventory holding cost. The selection of the route of delivery has to take the customer's time window into consideration as well. Thus, the delivery route becomes another key variable in optimizing cross-dock operations.

(4) Difficult to optimize end-to-end operations at cross-dock

As mentioned above, there are three main cost terms which need to be minimized: the penalty cost of early and tardy deliveries, the inventory holding cost, and the transportation cost. On the other hand, there are many constraints to this problem, such as limits on inbound and outbound dock doors, limitation of the cross-dock space, the large number of the products and customers, and the customer's delivery time windows. These factors make the cross-dock operations a challenge to formulate and solve as an optimization problem.

The problem is complex since it is a combination of vehicle scheduling and routing problems, in addition to scheduling of unloading and loading operations. A trade-off is required to minimize the total cost of earliness and tardiness penalty costs, inventory holding cost, and transportation cost within the constraints mentioned.

(5) Challenge in solving real life size problems

While formulating an optimization problem of cross-dock operations is difficult, what makes the problem impossible to solve in a reasonable time is the number of variables involved in a real-life size implementation. The amount of time taken to find an optimum solution may make the solution technique impractical. What is needed is an approach to reduce the solution space and solve the resulting problem in a reasonable amount of time. Alternative methods such as metaheuristics (Tabu Search, Ant Colony Optimization, Genetic Algorithm, etc) may also need to be explored.

1.4 OBJECTIVES OF RESEARCH

To address the problems stated above, this research looks to design and develop a comprehensive optimization model for distribution of products through the cross-dock facility. The model would consider the customer delivery time windows, the product consolidation decisions, and the routing of deliveries so that the products are received by customers on time, at minimum total cost. The product consolidation decision selects the products for consolidation and considers the trade-offs of FTL vs. LTL deliveries. The routing problem refers to the determination of the truck route (delivery route) from the cross-dock warehouse to the customers.

By considering the factors mentioned above, the model aims to minimize the total operating costs consisting of: (1) penalty cost of earliness, (2) penalty cost of tardiness, (3) inventory holding cost, and (4) transportation cost. Therefore the objectives of this research can be stated as follows:

1. Develop a model for vehicle routing and scheduling problem of cross-dock facility (VRSP).
2. Develop a model for vehicle routing and scheduling problem of cross-dock with customer zones and hard time windows (VRSP-CZHTW).
3. Propose a meta-heuristics based methodology to solve the large scale vehicle routing and scheduling problem of cross-dock, within acceptable time.

1.5 RESEARCH SCOPE

The scope of this research is as follows:

1. The system developed is for product distribution using cross-dock warehouse. The product in this model can be either perishable or non-perishable.
2. The system modelled consists of the following elements:
 - i. Suppliers
 - ii. Inbound trucks
 - iii. Cross-dock facility, consisting of
 - a. Cross-dock yard,
 - b. Inbound docks,
 - c. Unloading staging area,
 - d. Temporary storage area,
 - e. Loading staging area, and
 - f. Outbound docks
 - iv. Outbound trucks
 - v. Customers.

The system described above is shown in Figure 1.1 below

Figure 1.1 The cross-dock system under study

1.6 REPORT OUTLINE

Chapter 1 discusses the background, research problem, objective, scope, motivation, and outline of the research. Chapter 2 reviews the literature related to cross-docking, scheduling with time windows in the warehouse, and Tabu Search based algorithm. Chapter 3 describes the mathematical model for vehicle routing and scheduling problem of cross-docking (VRSP). Chapter 4 describes the mathematical model for vehicle routing and scheduling problem of cross-docking with customer zones and hard time windows (VRSP-CZHTW). Chapter 5 describes the proposed metaheuristics method, Tabu Search based algorithm,

to solve the original VRSP model, without the simplification of customer zones and hard time-windows. Chapter 6 provides conclusions of research and gives some directions for future research work.

CHAPTER 2

LITERATURE REVIEW

As mentioned in Chapter 1, this research aims to develop vehicle scheduling and routing policies for cross-dock operational planning. The policies should consider vehicle scheduling, the customers' time windows, the product consolidation decision, and the routing of delivery so the product can be shipped on time and at minimum cost.

The literature review is performed using 'Engineering Village' search engine and only includes papers published in English. From the literature review, it is found that there are numerous models related to cross-dock planning. This chapter reviews these models and highlights the state-of-the-art in the cross-dock facility planning domain. The models of vehicle scheduling and routing problems with time windows are also reviewed as they are two keys problems in planning of cross-dock operations. Finally the analysis of the literature and the research gaps are presented at the end of this chapter.

2.1 CROSS-DOCK PLANNING MODEL

There are various planning models proposed by researchers to optimize the operations of a cross-dock facility. This section reviews the models and classifies

them, based on the level of decision making, into: operational, tactical and strategic levels. This classification is depicted in Figure 2.1, which shows that there are seven groups of problems in cross-dock planning, with operational level having the most (5 types). Each level and group of problems is reviewed and discussed in the following sections.

2.1.1 Strategic Level

Mathematical models at the strategic level help to make decisions over the long term horizon. The strategic level addresses decisions such as determination of the number and location of cross-dock facilities and the number of vehicles in the network (suppliers-cross dock-customers). Thus, the problems arising at this level are related to design of the cross-dock network.

Figure 2.1 Classification of cross-dock planning problem

2.1.1.1 Cross-dock Network Design

Ratliff et al. (1998) develop a model for load driven network design, in the context of a railroad network, to determine the number and location of the mixing centres in the automotive delivery system. The objective of the model is to

minimize the total delay, consisting of transportation delay (i.e. travel time) and loading delay. The model is formulated as a Mixed-Integer Linear Program (MILP); Branch and Bound and Linear Programming (LP) relaxation methods are used to solve the model.

Donaldson et al. (1998) develop a model in a schedule driven cross-dock network context. The model determines the number of vehicles and their route and flow for the first class mail service of United States Postal Service. The objective of the model is to minimize the transportation cost. The model is formulated as an Integer Program (IP) and solved using Branch and Bound method and relaxation. Syarif et al. (2002) present a MILP model to determine which plants and distribution centres should be opened in order to minimize the network operation cost. A spanning tree based Genetic Algorithm is proposed to solve the model. Jayaraman and Ross (2003) propose a two phase model to determine the Production Logistic Outbound and Transportation design system. This model determines which warehouse and cross-docks are opened and their product allocation, so as to minimize the total cost. They present the model as a Mixed Integer Program (MIP) and propose a Simulated Annealing based meta-heuristic procedure to solve the model.

The work of Jayaraman and Ross (2003) is extended in Ross and Jayaraman (2008). The model is similar but two different meta-heuristics are proposed to solve the model, namely TABU-SA and RESCALE-SA. The contribution of this extended study is a better meta-heuristic method to solve the previous model.

Sung and Song (2003) propose an integer programming model to determine the location of cross-dock and vehicles allocated for the associated direct services in

a service network. The objective of the model is to minimize the cost of locating cross-dock and the cost of allocating vehicles. Tabu Search (TS) algorithm is proposed for the model. This work is later extended in Sung and Yang (2008) who propose a Branch-and-Price algorithm as an exact algorithm to solve their earlier model. The new approach provides more efficient solution than TS algorithm.

Gümüs and Bookbinder (2004) propose a LP model to determine a minimum cost cross-dock network design. Chen et al. (2006) propose a model to design the cross-dock network that considers delivery and pick up time windows, warehouse capacities, and inventory handling cost. The model is formulated as an LP that minimizes the transportation and inventory costs. TS and Simulated Annealing (SA) are proposed to solve the model.

Bachlaus et al. (2008) propose a novel algorithm titled Hybrid Taguchi-Particle Swarm Optimization (HTPSO) to design a network consisting of suppliers, plants, distribution centres, cross-docks, and customer zones. The problem is formulated as a multi-objective optimization model that aims to minimize the cost (fixed and variable) and maximize the plant and volume flexibility. Kreng and Chen (2008) develop a model to determine a production-distribution strategy so that the total percentage saving is maximized. This model helps in selecting the best distribution approach for the manufacturer's production process, whether using cross-docking or traditional distribution. Liu and Sun (2010) propose a LP model for cross-dock location problem; the objective of the model is to locate cross-dock terminals and minimize the sum of transport cost and running cost of cross-dock terminals.

Objective of Wu et al. (2011) is to determine the order point for each product type in a cross-dock facility that minimizes the average total cost including inventory holding cost, transportation cost and backorder penalty cost. The problem is solved by integrating simulation, Genetic Algorithm (GA) and smart computing budget allocation for a comprehensive total cost minimization problem. Mousavi et al. (2013) propose a zero-one program (ZOP) to determine the minimum number of cross-docks among a set of location centres so that each retailer's demand is met. A hybrid simulated annealing (HSA) algorithm embedded within TS is proposed to solve the model. Computational results show that the proposed HSA algorithm outperforms the SA and TS algorithms and converges quickly to high-quality solutions.

Shi et al. (2013) design a robust configuration for a cross-dock distribution centre that is insensitive to supply uncertainty, and provides steady parts supply to downstream assembly plants. A simulation model using factorial design and central composite design (CCD) is used to identify the models that best describe the relationship between simulation responses and system factors. Response surface methodology (RSM) identifies the factor levels that maximize the system potential. Javanmard et al. (2014) focus on modelling a system of cross-docks with multi-product type deliveries and pickups within specified time windows. The objective is to minimize the sum of transport and holding costs. In order to solve the model an efficient heuristic procedure is proposed which is later improved by an imperialist competitive algorithm.

Mehrizi et al. (2014) consider a two-echelon supply chain where the demand of a set of retailers is satisfied by a set of suppliers and shipped through a set of

capacitated cross-docks. The objective is to determine the number and location of cross-docks, and the assignment of retailers to suppliers via cross-docking so that the total cost of pipeline and retailer inventory, transportation, and facility location cost is minimized. The problem is formulated as a nonlinear MIP; a Column Generation algorithm is proposed to solve the problem. Tootkaleh et al. (2014) develop a model for cross-dock transportation network that considers direct shipment. To solve the model, Branch and Bound algorithm and dual price concept for sensitivity analysis is developed. When managers encounter problems, such as budget limit, they may decide to change the capacity of the trucks to reduce transportation cost. The algorithm provides a lower bound to the solutions of the problems and helps the managers to make appropriate selection of vehicle capacity.

2.1.2 Tactical Level

At the tactical decision level, the planning horizon of the decision is medium term, from few weeks to a few months. Research at the tactical level mainly addresses issues related to layout of the cross-dock network. A good layout can improve the movement (of inventory) and reduce congestion at the cross-dock, shortening the cycle time and improving efficiency of operations.

2.1.2.1 Cross-dock Layout Design

Some literature on cross-dock layout design is related to dock door assignment as it affects the material flow inside the cross-dock. Thus, dock door assignment needs to be considered in order to design an efficient cross-dock facility layout. Gue (1999) develops a material flow model to minimize the flow inside the cross-dock and finds that using ‘look ahead scheduling’ combined with dock door assignment results in a layout much better than First Come First Served policy. Bartholdi and Gue (2000) also propose a layout that considers the dock door assignment. The idea is to create a good layout through assignment of the dock doors. The model is formulated to minimize the cost of travel time, material handling and congestion; SA is used to solve the problem. With this model, the authors are able to reduce the labour load and labour cost through efficient layout of the cross-dock facility. Later Bartholdi and Gue (2004) investigate the various shapes of the cross-dock facility and find that the best shape is determined by the number of facilities and the freight flow pattern. As the size of cross-dock (number of dock doors) increases, the best shapes, in decreasing order, are I, T, and X.

Heragu et al. (2005) propose a LP model to jointly determine the functional area of the warehouse (storage, forward, and cross-docking area) and product allocation so that the total material handling cost is minimized. The model is solved using a heuristic algorithm. Hauser and Chung (2006) study cross-docking in manufacturing industry context (Toyota Motor Manufacturing plant, USA). They propose an algorithm based on GA to rearrange the layout. The objective

is to minimize the labour workload and lead time. The study finds that V layout performs better than the current I and T shapes.

Vis and Roodbergen (2008) develop a layout for the cross-dock to determine the temporary storage location for incoming unit loads. The objective is to minimize the travel distance of forklift trucks with unit loads. They model it as a minimum cost flow problem and propose “row-based storage assignment algorithm” to solve the model. Yanchang and Min (2009) compare the performance of three cross-dock designs: rectangular with conveyor, rectangular without conveyor and cross shape with conveyor. An Integer programming model is proposed to assign trucks to dock doors so as to minimize the total indoor distance travelled. Using GA, the study finds that rectangular shape with conveyor is the most efficient.

Vis and Roodbergen (2011) later propose a dynamic methodology to design the storage area of cross-dock facilities. The methodology can be applied for the selection of routing and storage assignment policies. Additionally, a dynamic layout rule can be derived based on the selected control policies. The experiments show that by applying the proposed methodology, 16% savings can be obtained in total distance travelled by employees.

2.1.3 Operational Level

In order to operate and manage the cross-dock facility efficiently, right decisions need to be made in operations at the right time. Operational decision making involves decisions over a short term planning horizon (daily, weekly). As mentioned earlier, research at this level is grouped along five dimensions:

scheduling, trans-shipment, dock door assignment, product allocation, and vehicle routing. Each of this set of problems and the related literature is reviewed next.

2.1.3.1 Cross-dock Scheduling

Inbound truck (with or without a trailer) scheduling is an important decision in cross-dock operations as it affects the receiving process. Poor vehicle scheduling can lead to congestion, slower product flow, longer processing times (make-span), all of which increase operation cost and lower customer satisfaction. In light of this, the objective of majority of the models reviewed is to minimize the make-span which leads to lower cost.

Li et al. (2004) propose a model based on “Just In Time” (JIT) scheduling to minimize the storage and order picking activity times. They develop an IP model to minimize the total penalty of earliness and tardiness of incoming and outgoing containers and propose two meta-heuristics methods, Squeaky Wheel Optimization embedded in a GA (SWOGA) and Linear Program within a GA (LPGA), to solve the model. Their work is later extended by Alvarez-Pérez et al. (2009) who use the same model and develop a different meta-heuristic procedure, namely Reactive GRASP and Tabu Search (RGTS) that works more efficiently and effectively.

McWilliams et al. (2005) study the parcel hub scheduling problem and model it as an LP with the objective to minimize the time span of transfer operation. The model determines the schedule of trailers at the unloading docks. A simulation-

based scheduling algorithm using GA is then proposed to solve the model. McWilliams (2009) proposes a dynamic load-balancing algorithm to solve the parcel hub scheduling problem, modeling it as a linear binary model.

Larbi et al. (2007) study the scheduling of transshipment operations inside a cross-dock facility to minimize the total cost of inventory holding and truck replacement. The model consists of one stack and one strip door (one inbound and one outbound door). The problem is modelled as a graph and a solution using shortest path methodology is proposed. This work is extended by Larbi et al. (2009) who consider multiple inbound and outbound doors, model it as a Dynamic Program (DP) and propose heuristics to solve it.

Song and Chen (2007) model a two stage cross-docking facility serviced by multiple inbound vehicles and one outbound vehicle. The objective of the model is to minimize the make-span; the heuristic methods based on Johnson's rule are proposed to solve it. Chen and Lee (2009) develop a model related to Song and Chen (2007), consisting of one inbound and one outbound trailer, instead of multiple inbound vehicles and a single outbound vehicle; the objective of their model is to minimize the make-span. The system is modelled as a two-machine flow shop problem and Branch and Bound (B&B) algorithm is used to solve it. Later, Chen and Lee (2009)'s work is extended by Chen and Song (2009), who modify the model so that at least one stage has parallel machines, and call it a hybrid cross-dock scheduling problem. In this model, there can be more than one inbound and outbound trailer. The problem is modelled as a MIP and solved using a heuristic method based on Johnson's rule.

Yu and Egbelu (2008) develop a scheduling model for a single truck, inbound and outbound docks and combine it with the assignment of the products. The model determines truck schedule and product allocation simultaneously, with the objective of minimizing the total operation time when a temporary storage buffer holds items at the shipping dock. Their work is extended by Vahdani and Zandieh (2010) who propose five meta-heuristics to solve and improve the solution obtained by the heuristic approach of Yu and Egbelu (2008). These five methods are GA, TS, SA, Electromagnetism-like Algorithm (EMA) and Variable Neighbourhood Search (VNS). These methods are shown to solve the problem faster and better than Yu and Egbelu (2008).

Arabani et al. (2011a) focus on applying five meta-heuristics (GA, TS, Differential Evolution (DE), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO)) to solve the model of Yu and Egbelu (2008). The experiments show that GA, PSO, ACO and DE algorithms have a relatively similar behaviour. Liao et al. (2012) propose two hybrid DE algorithms (hybrid DE #1 and #2) to determine the optimal inbound and outbound truck sequences at a cross-dock facility which has the objective to minimize the total operation time (or make-span). The two new hybrid DE algorithms have better performance than the pure DE algorithm that was reported to have the best performance in a recent comparative study of several metaheuristic algorithms (Arabani et al., 2011a). In addition, a more realistic and efficient operating policy is proposed that leads to shorter make-span than Yu and Egbelu (2008).

Shakeri et al. (2008) combine the truck scheduling problem with the dock door assignment problem. They develop a generic model for truck scheduling and

truck-dock-door assignment as a MIP. To minimize the total operation time this model is later improved by Shakeri et al. (2010) with the objective to determine the sequence of trailers and assign those to free doors. Two constructive heuristics are developed to solve the model: Dependency Ranking (DR) heuristic for truck sequencing and a Machine Fitness (MF) heuristic for door assignment. Maknoon and Baptiste (2009) propose a model for scheduling of inbound and outbound semi-trailers. This model differs from similar models in its objective function, which is to minimize the moving path of the products, rather than the travel distance or congestion. The model is limited to a single inbound and a single outbound door. Heuristic approaches and DP are proposed to solve the model.

Boysen et al. (2010) study the truck scheduling problem in which a single inbound door serves an outbound door. This work is similar to McWilliams et al. (2005) and Yu and Egbelu (2008). The problem is modelled as an MIP with the objective to minimize the make-span, and is then solved using the decomposition approach. In the same year, Boysen (2010) proposes a truck scheduling model for the food industry which prohibits any storage. The products arriving at the receiving dock are loaded immediately onto an outbound truck, and shipped to the customer immediately due to strict temperature requirements. The objective of the model is to minimize the flow time, processing time, and tardiness of outbound trucks. DP and SA are used to solve the model.

In the spirit of JIT delivery, Arabani et al. (2010) develop a multi-criteria objective function, consisting of earliness and tardiness, to schedule incoming and outgoing trailers at a cross-dock facility. The objective of the model is to

minimize earliness and tardiness, simultaneously, via a unified objective function. Three meta-heuristics are applied to solve the problem, namely GA, PSO, and DE.

Forouharfard and Zandieh (2010) propose a scheduling model for receiving and shipping trucks at cross-dock facilities which have a temporary storage in front of the shipping dock. The items can be held temporarily in storage until the delivery truck arrives. The objective is to find the best sequence of receiving and shipping trucks so as to minimize the total number of products that pass through temporary storage, which in turn reduces cost. Imperialistic Competitive Algorithm (ICA) and GA are proposed to tackle the problem.

Vahdani et al. (2010) study a cross-docking system in which there is no temporary storage and trucks are allowed not to load or unload all of their products; the trucks can instead be held (in the yard) and can come back to the dock to continue their loading later, if required. The system is modelled as a non-deterministic polynomial time hard problem. GA and EMA are proposed to solve the scheduling problem which has the objective of minimizing total flow time of the system, achieved through the best sequence of truck pairs.

Larbi et al. (2011) study the transshipment scheduling problem at a cross-dock with a single receiving and a single shipping door under three scheduling policies: complete, partial or no information on the order of arrivals and the contents of all inbound trucks. An optimal graph based model is used for the full information case, and a polynomial time algorithm is proposed to solve it; heuristics are developed for the other two cases.

Alpan et al. (2011) address transshipment scheduling problem for a multiple inbound and outbound cross-dock configuration. The problem is modelled as a DP, which is used to determine: (1) transshipment of products directly from inbound to an outbound truck if an outbound truck is available; (2) temporary storage of certain products which can be loaded later on; and (3) replace an outbound truck to facilitate direct loading. The objective is to find the best schedule of transshipment operations that minimizes the sum of inventory holding and truck replacement costs. Heuristics methods are proposed to solve the model.

Arabani et al. (2011b) propose a multi-objective scheduling model for a cross-dock system, with the objectives of minimizing the make-span and the lateness of all outbound trailers. Three multi-objective algorithms, including non-dominated sorting genetic algorithm-II (NSGA-II), strength Pareto evolutionary algorithm-II (SPEA-II), and sub-population genetic algorithm-II (SPGA-II) are expanded to solve the cross-dock scheduling problem, in which products are unloaded from inbound trailers at the receiving dock which are then categorized and loaded onto outbound trailers in the shipping dock. The experiments show the relative superiority of SPEA-II over the other two algorithms. Arabani et al. (2012) propose three multi-objective algorithms to solve the problem of Arabani et al. (2011b): SPGA-II, sub-population particle swarm optimization-II (SPPSO-II), and sub-population differential evolution algorithm-II (SPDE-II). The experiments show that the SPPSO-II algorithm is significantly better in terms of the quality of solution; however SPGA-II is the faster procedure.

Maknoon et al. (2014) study a scheduling problem in a satellite cross-dock by modelling the internal handling process and identify double handling as a key

factor in transshipment deficiency. The problem is modelled as an LP and a sequential priority-based heuristic is proposed to solve the problem.

Liao et al. (2014) develop five hybrid metaheuristic algorithms, including three hybrid ACO variants (hACO variant 1, 2, and 3), hybrid of SA and TS, and hybrid of SA and DE. They compare the performance of these algorithms in two related applications: unrelated parallel machine scheduling and inbound truck sequencing in a multi-door cross-dock system considered as a sequence dependent setup. The results of computational experiments indicate that the hybrid of SA and TS is the best if maximum CPU time is used as the stopping criterion, and the 2nd variant of hACO is the best if maximum number of evaluations is the stopping criterion.

The rest of the relevant research in cross-dock scheduling is shown in Table 2.1 Boysen and Fliedner (2010) also present a comprehensive literature review of cross-dock scheduling models; they classify and identify future research in the area of cross-dock scheduling.

Based on the above reviewed research, it can be concluded that the cross-dock scheduling problem is mainly modelled as LP, MILP, and IP. Most scholars model the scheduling problem to determine the schedule for both inbound and outbound sides (inbound truck and outbound truck), while some others model only to schedule one side (inbound or outbound truck only). In most of the models, the objective is to minimize the make-span, earliness, and tardiness. Most papers also propose heuristics and meta-heuristics methods to solve the proposed model, such as GA, TS, SA, ACO, PSO, DE, and VNS, as the LP solver cannot solve the large scale problems in acceptable time.

Table 2.1 Cross-dock scheduling models

AUTHOR	MODELLING APPROACH	CONTRIBUTION	SOLUTION APPROACH
Ley and Elfayoumy (2007)	LP	Develop a scheduling model for inbound and outbound truck to minimize the distance travelled.	GA
Chen et al. (2009)	LP	Develop a two stage scheduling model with the objective to minimize the total flow time of inbound and outbound jobs.	Heuristic algorithms
Wei et al. (2009)	LP	Develop a model to find the best truck dispatching rule. The system is incorporated with RFID. The best strategy is Shortest Remaining production Time (SRT) rule.	Simulation
Lee et al. (2012)	MIP	Propose a scheduling model of inbound and outbound trucks to maximize the number of products that are shipped over the planning horizon at the centres.	GA
Belle et al. (2013)	MIP	Introduce the truck scheduling problem for both inbound and outbound trucks at multiple dock doors. The objective is to minimize the weighted combination of total travel time and total tardiness.	TS
Konur and Golias (2013a)	Bi-objective bi-level optimization	Study truck scheduling problem at inbound doors in case of unknown truck arrival times. They determine a cost-stable scheduling strategy while minimizing the average of total service costs for the cross-dock operator.	GA
Konur and Golias (2013b)	Bi-level optimization -	Using the same model in Konur and Golias (2013a), in this paper, they propose three approaches to determine a scheduling strategy: deterministic, pessimistic and optimistic approach.	GA

Isfahani et al. (2014)	MIP	Consider multiple cross-docks in presence of temporary storage in front of the shipping docks. Receiving and shipping trucks can intermittently move in and out of the docks in between their task execution, and trucks can enter any of the cross-docks. Its objective is to minimize the total operation time or maximize the throughput of the cross-docking system.	SA and Firefly algorithms (FA)
Ladier and Alpan (2014)	IP	The objective is to minimize both the total number of pallets put in storage and the dissatisfaction of the transportation providers, by creating a truck schedule as close as possible to the desired schedule communicated in advance.	Heuristics
Zarandi et al. (2014)	IP and constraint satisfaction problem (CSP)	Minimizing the total of earliness, tardiness and the number of pre-emption for the outbound trucks on a cross-dock system. JIT scheduling optimization problem is modelled for solving the problem.	GA

2.1.3.2 Dock-door Assignment

Many researchers have studied the dock door assignment problem as it is a critical short to mid-term planning issue in cross-dock operations. Tsui and Chang (1990) develop a model to determine the assignment of receiving doors to the origins and shipping doors to the destinations. The objective of the model is to minimize the travel distance of the forklifts. A computer based tool with bi-linear programming is proposed to solve the model. The work is extended by Tsui and Chang (1992) who propose a solution to the bilinear programming model using branch and bound algorithm.

Oh et al. (2006) develop a model for assigning the destinations to the shipping doors in a cross-dock system handling mail. The problem is modelled as a non-linear mathematical model with the objective of minimizing the travel distance of the pallets within the centre. Two solution methods, three-phase heuristic procedure and GA-based, are developed to address the problem.

Lim et al. (2006a) model the truck dock assignment problem as an IP. The model considers the capacity of cross-dock and time window constraints with the objective to minimize the total shipping distance of cargo from inbound to outbound docks. TS and GA based algorithms are proposed to solve the model. In another paper, Lim et al. (2006b) modify the objective of the model to minimize the total cost, consisting of operation cost and penalty cost of unfulfilled demand. The problem is modelled as an IP and GA is used to solve the model. This work is then continued by Miao et al. (2009) who propose TS combined with GA to solve the model.

Bozer and Carlo (2008) study the assignment of trailers to dock doors, modelling it as an MIP, which is solved using SA. The objective is to minimize the material handling workload in a rectangular cross-dock. Ko, et al. (2008) present an approach for assigning destinations to shipping dock doors. The objective of the model is to minimize both the number of workers engaged in the loading operation and the imbalance ratio among the workers. GA approach is proposed with a line balancing heuristic.

Choy et al. (2012) study a job assignment problem in an industrial logistics cross-dock hub with limited storage space. They propose a method to minimise the waiting time of inbound and outbound trucks by coordinating schedules for order

picking and deliveries in the storage area. A GA-based meta-heuristic algorithm is developed to decrease the computational time in solving large-scale problems. Acar et al. (2012) introduce a mixed integer quadratic model with the objective of generating trailer-to-door assignments that equally distribute idle times at doors to accommodate operational level uncertainty encountered in truck arrival times. The objective of the model is to minimize the make-span by minimizing idle times at each door. The trailer assignments to doors are dynamic, based on truck arrival times and service times rather than static, only based on their origin and destination. A door assignment heuristic (DAH) is presented to solve the problem.

Liao et al. (2013) propose a model for simultaneous dock assignment and sequencing of inbound trucks for a multi-door cross-dock operation. The objective is to minimize total weighted tardiness under a fixed outbound truck departure schedule. Six different metaheuristic algorithms are proposed to solve the problem, including SA, TS, ACO, DE, and two hybrid DE algorithms. The experiments show that ACO is the best among all the metaheuristic algorithms employed.

Nassief et al. (2015) study the cross-dock door assignment problem in which the assignment of incoming trucks to strip doors and outgoing trucks to stack doors is determined. The objective of the model is to minimise the total material handling cost. The problem is formulated as MIP embedded in a Lagrangean relaxation.

2.1.3.3 Trans-shipment

The trans-shipment problem tries to answer the following four questions: how much to ship, between which locations, on which routes, and at what times (Lim et al., 2005). Some work has been done in the trans-shipment problem since operation inside the cross-dock can be modelled as trans-shipping from inbound to outbound dock. It determines the product flow allocation to the outbound dock for fulfilling customer demand.

Lim et al. (2005) extend the traditional trans-shipment model by considering inventory, capacity of cross-dock facility and time window constraints in the model. The study develops some variant models combining fixed/flexible scheduling, multiple/single shipping, and multiple/single delivery.

The work of Miao et al. (2008) is related to Lim et al. (2005). They study the trans-shipment problem where the transportation activities have fixed schedule and shipping and delivery are only executed within their time windows. The objective is to minimize the shipping and inventory holding cost and is solved by GA.

Extending the two previous studies, Miao et al. (2009) formulate a trans-shipment model in the context of a fixed schedule, single shipping, and single delivery. Their model is adopted from Lim et al. (2005) and is formulated as an IP with the objective to minimize the transportation cost, penalty cost, and inventory holding cost; the problem is then solved using GA based methodology. Tang and Yan (2010) develop a model for two typical cross-dock operations: pre-distribution cross-docking (Pre-C) and post- distribution cross-docking (Post-C)

when trans-shipment among retail stores is considered. Different measures of operational performance are investigated and compared. Considering the inventory cost, trans-shipment cost and operations cost at the cross-dock, the analytical results show that the suitability of Pre-C and Post-C is highly sensitive to operating environment, such as uncertainty of demand, unit operations cost at the cross-dock, unit inventory holding and shortage costs.

2.1.3.4 Vehicle Routing Problem

Cross-dock vehicle routing problem is an operational level problem, affecting decisions made over the short term. The vehicle routing problem considers the cross-dock as the network between the suppliers and the customers.

Lee et al. (2006) first relate the vehicle routing problem with cross-docking and consider the time window, calling it Vehicle Routing Problem with Time Windows (VRPTW). In VRPTW, each node must be served by exactly one vehicle within its time window. The model determines the number of vehicles and the best route, schedule, and the arrival time of each vehicle at the cross-dock so that the suppliers and customers are visited within their time windows. The objective of the model is to minimize the transportation cost. The model assumes that all the vehicles arrive simultaneously at the cross-dock from their pick up routes. Thus synchronized schedule of vehicle arrivals at the cross-dock becomes the constraint. The problem is modelled as a MIP and solved by TS algorithm.

Wen et al. (2009) study the problem of vehicle routing with cross-docking, VRPCD, which is similar to Lee et al. (2006). However, Wen et al. (2009) do not

put a constraint on simultaneous arrival of the vehicles. There is a dependency among the vehicles that is determined by the consolidation decisions. Moreover, this model has a predetermined time window for each delivery and pick up process. The problem is modelled as MIP; a heuristic method embedding TS within an Adaptive Memory Procedure (AMP) is proposed to solve it.

Musa et al. (2010) study the transportation problem where the loads are transferred from the suppliers to the retailers through a cross-dock, without any storage in a distribution centre. The objective is to minimize the transportation cost in a network by loading trucks at the supplier locations and then routing them either directly to the customers, or indirectly through cross-dock facilities where they can be consolidated. The problem is formulated as an IP model and solved using a novel ACO algorithm.

Liao et al. (2010) extend the work of Lee et al. (2006) and propose a new TS algorithm. The computational results show that the proposed TS algorithm provides a superior solution while using much less computation time.

Santos et al. (2011a) propose an exact algorithm to solve the VRPCD problem as in Wen et al. (2009). In this system, each vehicle leaves the cross-dock to pick up or deliver orders and returns to the cross-dock. The VRPCD consists of finding routes for all K vehicles to visit suppliers and customers once in order to collect and deliver items, respectively, without exceeding the vehicle capacity. Each route must start and end at the cross-dock. The objective of VRPCD is to find routes so that the total cost (given by the sum of the arcs traversed by the vehicles and the cost of changing loads at the cross-dock) is minimized. The authors propose a column generation formulation to evaluate tighter lower

bounds and implement a branch-and-price algorithm to solve VRPCD instances derived from Wen et al. (2009). Aiming to improve their earlier results, Santos et al. (2011b) propose a novel column generation algorithm to fix shortcomings of their previous algorithm. In the column generation sub-problem of the new algorithm, they implement a Branch-and-Cut algorithm to obtain optimal solutions and a heuristic algorithm based on GRASP meta-heuristic to approximate such solutions faster. The experiments show that the results outperform those obtained from the previous algorithm for both lower and upper bounds.

Dondo et al. (2011) study N-echelon Vehicle Routing Problem with Cross-Docking in Supply Chain Management (VRPCD-SCM). The objective is to satisfy the customer demands at minimum transportation cost. Factories deliver multiple items to customers that are managed by routing and consolidating shipments in warehouses. The model applies hybrid strategies combining direct shipping, warehousing and cross-docking. The problem is modelled as a MILP and a monolithic optimization framework is applied to solve the problem.

Petersen and Ropke (2011) develop a model of pickup and delivery with cross-docking opportunity. The model considers pickup requests, delivery requests, and pickup-and-delivery requests. Each pickup-and-delivery request can be served either as direct delivery by one vehicle, or by being picked up and transported to the cross-dock by one vehicle, and subsequently delivered to its final destination by a different vehicle. The objective of the model is to minimize transportation cost. The problem is solved using Large Neighbourhood Search (LNS) approach.

Tarantilis (2013) proposes a TS algorithm embedded within an adaptive multi-restart framework to solve the problem of Wen et al. (2009). The objective is to design minimum cost routes for the transportation of products via a cross-dock. Vehicles cannot travel directly from suppliers to customers, and thus, products arriving on inbound vehicles are sorted and consolidated onto outbound vehicles. The computational experiments show that the proposed approach is highly efficient and more effective than Wen et al. (2009).

Vahdani et al. (2012) extend the work of Lee et al. (2006) by proposing a new hybrid metaheuristic for vehicle routing scheduling in cross-dock systems. They propose a new hybrid algorithm to solve the problem of Lee et al. (2006), which incorporates the elements of PSO, SA, and VNS to enhance its search capabilities. The new method outperforms the TS algorithm of Lee et al. (2006). Dondo and Cerdá (2013) address the VRPCD. Two different solution approaches are developed: MILP and a sweep-heuristic based MILP formulation. Both approaches assume that the cross-dock facility has a sufficiently large number of doors so that every truck can immediately start unloading/loading after it arrives at the terminal, or is ready for delivery duties. Pickup and delivery tasks are carried out by the same fleet of homogeneous vehicles. Moreover, goods collected and delivered by the same vehicle are not unloaded at the cross-dock, and pickup/delivery splits are not allowed. Numerical experiments show that the sweep-based model provides near-optimal solutions in a much shorter (CPU) time than the exact formulation.

Harks et al. (2013) propose an approximation algorithm for different variants of capacitated location routing, a generalization of vehicle routing which takes into

account the cost of opening the depots from which the vehicles operate. A computational study of the algorithm is presented for capacitated location routing on benchmark instances and large-scale randomly generated instances. The quality of the computed solutions is much closer to optimality than the provable approximation factor.

Hu et al. (2013) consider a model with multiple suppliers and customers in a single cross-dock centre. They formulate the optimal route selection problems from the suppliers to the cross-dock centre and from the cross-dock centre to the customers as an MILP. Based on the relationship between the suppliers and the customers, the proposed model integrates the two vehicle routing models to optimize the overall travel time, distance, and waiting time at the cross-dock centre.

Mousavi and Moghaddam (2013) address the location and routing scheduling problems with cross-docking. Their aim is to concurrently design a cross-dock centre location and a vehicle routing schedule. They model the problem as a two-stage MIP and propose an algorithm based on two-stage hybrid SA (HSA) with a tabu list taken from TS to solve it. The computational results show that the proposed HSA performs well and converges fast to reasonable solutions.

Mousavi et al. (2013) consider the design of cross-dock systems under uncertainty in two phases: (1) a strategic decision-making process for selecting the location of cross-docks, and (2) an operational decision-making process for vehicle scheduling with multiple cross-docks. This is the first attempt to introduce an integrated model for cross-dock system design under a fuzzy environment. To solve the model, a new fuzzy mathematical programming based

possibilistic approach is used. The computational results confirm the applicability and suitability of the proposed method.

Fakhrzad et al. (2014) address a particular type of vehicle routing problem, called Vehicle Routing Problem of Cross-Dock with Time Windows (VRPCDTW), considering a time limitation for each customer/retailer. Two meta-heuristic algorithms based on TS and VNS are proposed for its solution. Computational experiments confirm that the proposed TS algorithm performs better than VNS algorithm, in total cost and computation time. Santos et al. (2013) introduce the PDPCD model that allows the vehicles to avoid the dock, thus reducing transportation costs. They propose IP formulation and a Branch-and-Price algorithm for the problem.

Hosseini et al. (2014) develop a novel IP for a consolidation network where a set of vehicles are used to transport goods from suppliers to their customers using one of three transportation systems: direct shipment, shipment through cross-dock (indirect shipment) and milk run. They propose a hybrid of Harmony Search (HS) and SA (HS-SA algorithm) to solve the problem. The objective is to minimize the total shipping cost in the network by reducing the number of required vehicles, using an efficient vehicle routing strategy. The experiments show that the proposed approach performs much better in reducing both the shipping cost and the computational time requirement.

Moghadam et al. (2014) model a vehicle routing and scheduling problem for a network consisting of suppliers, customers and a cross-dock. A set of homogenous vehicles transfer products from suppliers to customers through a cross-dock. Each vehicle has a limited capacity and both suppliers and customers

must be visited within their time windows, and a customer can be visited more than once by different vehicles. The problem is modelled as mixed integer nonlinear mathematical program (MINLP). SA and hybrid metaheuristic algorithm combining ACO and SA are proposed to solve it. The experimental results verify the superior performance of the hybrid algorithm compared to the proposed SA algorithm.

Morais et al. (2014) address the VRPCD by defining a minimum cost set of routes for a fleet of vehicles that satisfies the demands of products for a set of suppliers and customers. The vehicles leave a single cross-dock for the suppliers, pick up products and return to the cross-dock, where products can be exchanged before being delivered to their customers. The vehicle routes must respect the vehicle capacity constraints, as well as the time window constraints. They adapt a constructive heuristic and propose three Iterated Local Search heuristics.

Mousavi et al. (2014a) propose a new mathematical model for the location of cross-dock facilities and vehicle scheduling problems in distribution networks. A two-phase MIP is formulated and a new SA-based heuristic is developed for solving the proposed model. The computational results for different-sized problems show that the proposed algorithm performs effectively.

Mousavi et al. (2014b) study the location of multiple cross-dock centres (CDCs) and vehicle scheduling problem. They introduce new two-phase deterministic MILP that is integrated for the location of CDCs and the vehicle scheduling problem with multiple CDCs. They propose a hybrid fuzzy possibilistic-stochastic programming solution approach to solve the model.

2.1.3.5 Product Allocation Problem

The product allocation problem is related to the determination of product distribution strategy through the cross-dock warehouse. Companies usually use a combination of cross-docking and traditional warehousing at their distribution centres. Since each product has different characteristics, there is a need for a model that determines the best strategy for product distribution. Li et al. (2008, 2009) propose a systematic approach and an optimization model for product allocation decision for Fast Moving Consumer Goods (FMCG). This model supports development of alternative plans for allocating products to a cross-dock and a traditional warehouse.

2.2 VEHICLE ROUTING AND SCHEDULING WITH TIME WINDOWS

This section reviews general vehicle routing and scheduling models with time windows that do not necessarily include a cross-dock facility. This review is important since it gives insight into various vehicle routing and scheduling approaches with time windows in a broader sense. This research area has attracted many scholars since good vehicle routes are a pre-requisite for an efficient logistics network. For example, a well-developed vehicle route can minimize the cost of distribution and lead to increased profits.

Solomon (1987) seems to be the first to study the vehicle routing problem considering time windows, calling it Vehicle Routing Scheduling Problem with Time Windows (VRSPTW). Solomon and Derosiers (1988) then survey the

significant advances made for the class of routing problems with time windows: these are the single and multiple travelling salesman problem, the shortest path problem, the minimum spanning tree problem, the generic vehicle routing problems, the pickup and delivery problem including the dial-a-ride problem, the multi-period vehicle routing problem and the shoreline problem.

Fagerholt and Christiansen (2000a, b) propose a model that combines ship scheduling problem with multi-ship pickup and delivery problem, time windows, and multi-allocation problem. Christiansen and Fagerholt (2002) focus on the ship scheduling problem that is concerned with the pickup and delivery of bulk cargo within given time windows. Al-Khayyal and Hwang (2007) study ship routing and scheduling for chemical maritime transport companies. They propose a model to determine minimum cost routing in a network for a heterogeneous fleet of ships engaged in pickup and delivery of several liquid bulk products.

Another body of literature focuses on solving the vehicle routing problem for hazardous material transportation. Zografos and Androutsopoulos (2004) propose a bi-objective vehicle routing and scheduling problem of hazardous materials. The model takes into account time windows and minimizes the risk of accident and the cost to determine the routes of the trucks distributing hazardous materials. Meng et al. (2005) address a novel time-varying multi-objective vehicle routing and scheduling problem for hazardous materials transportation. The problem is formulated as a time-varying multi objective shortest path network reconstructed by the space-time network technique and efficient DP method.

Yet another body of literature looks at the pick-up and delivery problem as a variant of the vehicle routing problem. Pankratz (2005) studies the Dynamic Pickup and Delivery Problem with Time Windows (DPDPTW) and proposes a new GA to solve the model. Jia et al. (2009) also study a dynamic pickup and delivery problem, calling it Real-Time PDPTW (RTPDPTW). A quick heuristic based on λ -opt algorithm is proposed to solve the model.

Still others variants of vehicle routing model can also be found in the literature. Hooker and Natraj (1995) modify the classical chain decomposition model for vehicle routing and present a general model for vehicle routing and scheduling problem with time windows. Another variant of the vehicle routing problem is proposed by Ibaraki et al. (2005) who consider soft time windows in vehicle routing problem. Ho and Haugland (2004) introduce split deliveries in the general vehicle routing problem. Tan et al (2006) develop a vehicle routing model for truck and trailer. The proposed model can determine a complete minimum distance routing schedule for serving jobs that utilizes least number of trucks.

Chunyu (2007) focuses on e-commerce and proposes a VRPTW model. Chen and Chen (2008) consider load balancing in VRPTW. The model uses minimum number of vehicles, the shortest distribution distance, load balancing and customer's time demand. Cheng and Wang (2009) develop the VRPTW model that assists the frozen-food distribution centers in solving their vehicle dispatching and routing problem. Chiang et al. (2009) focus on the supply chain of US newspaper and introduce zoning constraints in the VRPTW. Androutsopoulos and Zografos (2009) develop a new model of multi-criteria

time-dependent routing and scheduling problem on a multimodal fixed schedule network.

Other variants of vehicle routing problem can be constructed by combining vehicle routing problem with another problem. Lau and Song (2002) combine inventory and routing schedule so that the system-wide total cost over the planning horizon is minimized. The problem is modeled as a dynamic lot sizing and vehicle routing problem. Han et al. (2007) combine supply location selection with routing problem for emergency material delivery. The problem includes warehouse selection and fleet scheduling and routing to meet the demand within the required time window.

Yan et al. (2008) propose an optimal scheduling model of distribution, a vehicle scheduling model, with multiple distribution centres and multiple types of vehicles that considers constrained condition for types of vehicles, vehicle capacity, customers' time windows, and traffic conditions. Fan (2009) develops a new dynamic vehicle scheduling algorithm with changeable time windows that considers bespoke demand (i.e. make to order) and real-time demand. Karimi et al. (2005) develop an event-based "pull" approach for minimum-cost or maximum-profit scheduling of transport and cleaning of multi product tank containers (loaded and empty) from a set of projected shipment orders.

Some research work focuses on developing new algorithms to solve the vehicle routing problem more efficiently as modelled by other researchers. First one is an improved ACO algorithm proposed by Zhang and Zhang (2008). Second algorithm, proposed by Chen et al. (2009), is a new algorithm that integrates TS and Threshold Accepting (TA) entitled Tabu Threshold Algorithm (TTA).

Thirdly, Jiang et al. (2009) propose PSO algorithm with crossover to solve the vehicle routing problem. Table 2.2 lists some key publications in the area of vehicle routing in logistics.

Table 2.2 Vehicle routing models and their variants in logistics

CONTRIBUTION	RESEARCHERS
Propose Vehicle Routing Scheduling Problem with Time Windows (VRSPTW) model in logistics.	Solomon (1987)
Consider soft time window in VRP	Ibaraki et al. (2005)
Develop the Vehicle Routing Problem for water transportation (ships).	Fagerholt and Christiansen (2000a,b); Christiansen and Fagerholt (2002); Al-Khayyal and Hwang (2007)
Study the vehicle routing and scheduling problem for hazardous materials.	Zografos and Androutsopoulos (2004); Meng et al. (2005)
Focus on the Pickup and Delivery Problem with Time Windows.	Pankratz (2005); Jiang et al. (2009)
Introduce split deliveries in the general Vehicle Routing Problem (VRP).	Ho and Haugland (2004)
Develop VRP model for truck and trailer.	Tan et al (2006)
Develop Vehicle Routing Problem with Time Window (VRPTW) for electronic commerce.	Chunyu (2007)
Consider load balancing in Vehicle Routing Problem with Time Windows (VRPTW).	Chen and Chen (2008)
Propose the vehicle routing problem with time windows (VRPTW) for frozen-food distribution centers.	Cheng and Wang (2009)
Introduce zoning constraints in the VRPTW.	Chiang et al. (2009)
Propose the multi-criteria time-dependent routing and scheduling problem on a multimodal fixed scheduled network.	Androutsopoulos and Zografos (2009)

Combine the inventory and the vehicle routing problem.	Lau and Song (2002)
Combine the supply location selection with routing problem for emergency material delivery.	Han et al. (2007)

2.3 TABU SEARCH

This section describes various applications of TS in the literature. TS is one of the most popular metaheuristics to solve the NP hard combinatorial problems (Glover et al., 1993). TS can provide near optimal solutions quickly, compared with other heuristics methods. TS has been used to find solutions superior to the previously obtained best by alternative methods (Glover, 1990). Since, VRSP is basically an integration of vehicle routing and scheduling problems, this section highlights successful applications of TS in both these areas.

In recent years, TS has shown to be one of the most popular generic heuristics in solving VRP, especially VRPTW (Chen et al., 2009). In cross docking research, TS has been frequently used as the solution approach for VRP. Lee et al. (2006) propose TS algorithm for a two-level distribution network with a single cross-dock, assuming that all vehicles performing pickups arrive at the cross-dock simultaneously, within a predefined time window. Liao et al. (2010) propose an improved TS approach and report shorter computational times for the set of problem instances introduced by Lee et al. (2006).

Wen et al. (2009) propose a TS algorithm embedded within an Adaptive Memory Programming framework to solve the VRPCD. The algorithm is tested on problem instances with up to 200 supplier-customer pairs. Tarantilis (2013)

focuses on VRPCD as defined by Wen et al. (2009). He proposes a TS algorithm embedded within an adaptive multi-restart framework to solve these problems. Madani (2013) investigates TS for finding optimal sequence of incoming and outgoing trailers at cross-docks. The results indicate that TS performs significantly better than the descent method for large problem instances. Some other applications of TS method to VRP are listed in Table 2.3.

Table 2.3 Tabu Search-based algorithm in VRP

No	Reference	Contribution
1	Semet and Taillard (1993)	Applies TS to a real world VRP with many side constraints, including time windows.
2	Barnes and Carlton (1995), Carlton (1995), Rochat and Taillard (1995), and Potvin et al. (1996)	Produce near optimal solutions to Vehicle Routing Problem with Hard Time Window (VRPHTW) using TS.
3	Cordeau et al. (1997)	Propose a unified TS that is capable of solving periodic VRP, periodic traveling salesman problem, and multi-VRP.
4	Taillard et al. (1997)	Propose TS for vehicle routing problem with soft time windows.
5	Schulze and Fahle (1999)	Propose a new parallel TS for VRPTW.
6	Cordeau and Laporte (2001)	Study the Site Dependent Vehicle Routing Problem (SDVRP). A TS heuristic for SDVRP and SDVRP with Time Windows is described and tested on randomly generated instances.
7	Cordeau et al. (2001)	Present a unified TS for periodic VRPTW and the multi-depot VRPTW.

8	Cordeau and Laporte (2003)	Propose TS for dial-a-ride problem. The aim is to design a set of least cost vehicle routes capable of accommodating all requests.
9	Ho and Haugland (2004)	Propose TS for VRP with time windows and split deliveries (VRPTWSD).
10	Fu et al. (2008)	Propose a unified penalty function and a unified TS for vehicle routing problems with soft time windows (VRPSTW).
11	Yao et al. (2008)	Propose a TS for Vehicle Routing Problem with Time Windows.
12	Brandao (2009)	Proposes a deterministic TS for fleet size and mix VRP.
13	Derigs and Reuter (2009)	Propose TS for solving open VRP.
14	Bolduc et al. (2010)	Propose TS for split delivery Vehicle Routing Problem with Production and Demand Calendars (VRPPDC).
15	Cordeau and Maischberger (2012)	Propose parallel iterated TS for solving four different routing problems: classical VRP, periodic VRP, multi-depot VRP, and site-dependent VRP.
16	De Freitas (2012)	Propose TS for Multiple Depot VRP with Time Windows.
17	Jinhui (2012)	Propose TS for open VRP with soft time windows.

TS has become one of the most popular methods in solving scheduling problems.

The application of TS in scheduling problems can be found in Table 2.4.

Table 2.4 Tabu Search-based algorithm in scheduling problems

No	Reference	Contribution
1	Mooney and Rardin (1993)	Propose TS for a class of scheduling problems.
2	Liaw (1999)	Propose TS for open shop scheduling problem.
3	Lim et al. (2002)	Determine crane Scheduling using TS.
4	Ponnambalam et al. (2002)	Determine the scheduling of Flexible Manufacturing System using TS.
5	Liaw (2003)	Proposes TS for two-machine pre-emptive open shop scheduling problem.
6	Chen et al. (2007)	Propose TS for integrated scheduling problem of container handling systems in a maritime terminal.
7	Eswaramurthy and Tamilarasi (2007)	Propose TS for solving job shop scheduling problems.
8	Meng and Tang (2010)	Propose TS to solve the scheduling problem for a batch-processing machine with non-identical job sizes.
9	Adamuthe and Bichkar (2012)	Propose TS for Solving Personnel Scheduling problem.
10	Solimanpur and Elmi (2013)	Propose TS for cell scheduling problem with makespan criterion.

Since the literature-review above has shown, TS is an effective approach to solve VRP and scheduling problems, this research proposes that TS-based algorithm be applied to solve the VRSP problem. The TS-based algorithm for VRSP is presented in Chapter 5.

2.4 SUMMARY OF MODELLING AND SOLUTION APPROACHES

From the literature review, it can be seen that the existing models in cross-dock research area are solved mainly by heuristics and meta-heuristics approaches as these problems cannot be solved to optimality in an effective and efficient way. The literature mentions that the commonly available software (such as LINGO or CPLEX) can only solve (taking a long time) small to medium size problems. Since a cross-dock warehouse manages many products and trucks in a day and requires a system that is able to give decisions quickly, the existing tools are not applicable to solve the real world problems in a timely manner. For these reasons, the researchers have proposed heuristics and meta-heuristics-based approaches to solve the model in more efficient and effective way.

2.5 RESEARCH GAP

A detailed analysis of the comprehensive literature review is performed to identify the research gap in cross-dock operations planning and scheduling. The analysis and the research gap are presented next.

For the assignment problem, most researchers only consider the dock door assignment itself (Tsui and Chang, 1990); some models also consider time windows in their assignment model (Lim et al., 2006a and b). Some models of layout design are related to dock door assignment (Bartholdi and Gue, 2000), while others, by Heragu et al. (2005) and Li et al. (2008, 2009), are related to

product allocation. Gue (1999) considers scheduling for the proposed layout design while Bartholdi and Gue (2004) only consider the layout itself.

Some models for cross-dock network design relate it to scheduling (Donaldson et al., 1998), others consider time windows also (Sung and Song, 2003; Chen et al., 2006) while network-based models only consider the network itself (Jayaraman and Ross, 2003). Trans-shipment models are related to scheduling and the network (Lim et al., 2005). For the vehicle routing problem, Lee et al. (2006) propose a model that considers vehicle routing with consolidation while Wen et al. (2009) consider vehicle routing with time window and consolidation. Majority of scheduling models only consider scheduling itself (Maknoon and Baptiste, 2009; Boysen et al., 2010), while Li et al. (2004) model it by considering time windows also. Shakeri et al. (2008) combine scheduling with assignment problem and Larbi et al. (2007) propose a scheduling model for trans-shipment operations inside the cross-dock facility.

From the above analysis, it is found that what the models are lacking is integration between vehicle scheduling and vehicle routing that also considers time-windows and product consolidation. Therefore, it can be concluded that while much work exists to determine the best sequence for inbound and outbound jobs or trucks in cross-dock, what is still missing is a scheduling and vehicle routing model that also considers product consolidation and time-windows, all integrated within one model. By combining the two problems together and considering time windows and product consolidation, the resulting model would optimize the two problems jointly leading to improved performance of the cross-dock warehouse.

The potential benefit of solving the two problems in an integrated manner is that the truck schedules and the truck delivery decisions are bound to be synchronized, and should lead to minimum total cost of delivery, including earliness, tardiness, inventory holding, and transportation costs.

In this research, the customer satisfaction is measured by the fact that delivery takes place within the specified time window, leading to zero earliness or tardiness costs. A delivery out of the specified delivery window leads to a dissatisfied customer. Thus, customer satisfaction can be measured by the earliness and tardiness costs that occur. Higher the earliness or tardiness costs higher the dissatisfaction of the customers, and vice versa.

As for the practical relevance of the proposed research, the resulting solution has the potential to be incorporated into a decision-support system for the cross-dock warehouse. As mentioned by Apte and Viswanathan (2000) and Saddle Creek Corp. (2011) (section 1.2), the implementation of an IT system is highly important to facilitate efficient and effective operation of the cross-dock facility.

CHAPTER 3
VEHICLE ROUTING AND SCHEDULING
AT A CROSS-DOCK

This chapter describes the model for vehicle routing and scheduling problem at a cross-dock, abbreviated as VRSP. The model consists of suppliers, inbound trucks, cross-docking, outbound trucks, and customers. The model tries to determine the vehicle routing and vehicle scheduling simultaneously. The objective of VRSP is to minimize the earliness, tardiness, inventory holding, and transportation costs in distributing the products from suppliers to customers through a cross-dock warehouse. The problem is modelled as a Mixed Integer Linear Program (MILP). After developing and formulating VRSP model, the experiments are done to validate the model using CPLEX Solver version 12.3. The results of experiment are discussed at the end of this chapter, along with a summary.

3.1 MODEL DESCRIPTION

The cross-dock system modelled in this chapter consists of several sub systems as shown in Figure 1.1. One of the key features of the VRSP model is that it specifies the arrival time, release time, departure time, routing of deliveries and ensures that the products are delivered to the customers at minimum cost, which includes earliness, tardiness, inventory holding, and transportation costs. The

earliness and tardiness costs capture the product quality and customer satisfaction, while the inventory holding cost and transportation cost represent the operational cost. The product is delivered to the customer in good quality and on time when it is delivered within the customer-specified time-window.

In term of the vehicle's departure time, VRSP model is different from Yu and Egbelu (2008) and Li et al. (2004). In Yu and Egbelu (2008), the departure time only minimizes the total operation time inside the cross-dock facility, while in Li et al. (2004) the departure time only minimizes the earliness and tardiness penalty cost of the processes inside the cross-dock facility. In VRSP model, the departure time specifies the time the outbound vehicles must depart from cross-dock to the customer location so that the total cost, consisting of penalty cost (of earliness and tardiness), the inventory holding cost inside the cross-dock, and the transportation cost from the cross-dock to the customer location, is minimized.

VRSP model also differs from other vehicle routing models in that the proposed model integrates the vehicle routing problem with vehicle scheduling problem. VRSP also specifies the schedule for inbound vehicle arrival and release time and outbound vehicle departure time and completion time of delivery at the customer location.

3.2 MODEL DEVELOPMENT

When the customer places the order to the supplier, the supplier's deliveries are processed through the cross-dock and consolidated with other incoming deliveries. The customers specify the orders and delivery requirements, such as

the items, the quantity, and its requested delivery time window. Based on this information, the manager of cross-dock facility needs to develop a plan for incoming and outgoing deliveries.

A good plan should specify a schedule for inbound vehicle arrival time to the cross-dock, schedule for unloading activity in the cross-dock (inbound vehicle release time), and schedule for delivery to the customers, that consists of decisions for order/product consolidation, outbound vehicle departure time from cross-dock, outbound vehicle routing of delivery, and completion time for order delivery to the customer location. The plan should minimize the earliness and tardiness costs, the inventory holding cost, and the transportation costs.

There is a temporary storage area in the cross-dock center where unloaded orders, yet to be consolidated, are stored. The consolidated products are loaded onto the outbound vehicle before the vehicle can depart from the cross-dock to deliver the products to different customer locations, finally returning to the cross-dock center at the end of the day.

The process flow at the cross-dock center is described next. The inbound vehicle arrives at the cross-dock facility and is processed in the inbound area and unloaded. After unloading, the order can be sent directly to the outbound area for loading or stored in the intermediate storage area while the orders are waiting to be loaded. These choices are determined by the decision of the release time and departure time. Release time is the time at which inbound vehicle unloads its products at the cross-dock. Then, based on the departure time and consolidation decisions, the orders are loaded onto the outbound vehicle for delivery. Departure time is the time at which the outbound vehicle departs from cross-dock to the

customer location for delivery. So at the time of departure, the loading processes of products onto the outbound truck should have been completed and the truck should be ready to depart.

In this model, there are two key decisions to be made. The first decision relates to handling inbound shipments: the operations and schedule at the cross-dock yard and inbound area. The second decision is related to the outbound-side of the facility: operations and schedule of the outbound area up to the customer location.

At the receiving yard, the manager determines the schedule for inbound vehicle arrival time to cross-dock center (AT_l). In the inbound area, once the inbound vehicle has arrived and is parked at the cross-dock yard, the plan has to specify

- (1) the release time of inbound vehicles (RT_l),
- (2) sequence of inbound vehicles at inbound dock door (x_{lj}), and
- (3) assignment of inbound vehicles to an inbound dock door (y_{lf}).

After the products are unloaded, the decisions related to their loading and delivery need to be made; these are:

- (1) product consolidation (v_{ik}),
- (2) scheduling of outbound vehicles, consisting of
 - a. departure time of outbound vehicles (s_{ok}),
 - b. sequence of outbound vehicles at outbound dock door (X_{kj}), and
 - c. assignment of outbound vehicles to outbound dock door (Y_{kh}),
- (3) routing of outbound vehicle for customer order deliveries (z_{ijk}), and
- (4) completion time of order delivery (s_{ik}).

Once these decisions have been made, product earliness, tardiness, inventory holding, and outbound transportation costs can be computed. The process-flow at the cross-dock facility is shown in Figure 3.1.

Figure 3.1 Cross-dock process-flow

3.3 MODEL FORMULATION

The VRSP model formulation is inspired by the cross-dock scheduling model proposed by Li et al. (2004) and the cross-dock vehicle routing model proposed by Wen et al. (2009). The VRSP model also incorporates product consolidation and delivery time window considerations. The objective of the model is to

determine the schedule for inbound vehicles (their arrival and release times), outbound vehicle departure times, and the routes of delivery, so as to fulfill the customer delivery time window requirements and minimize the total cost of deliveries.

The following assumptions are used in the development of the model:

- 1) Cross-dock yard space is unlimited.
- 2) Every inbound and outbound dock door is used for unloading and loading (no empty inbound and outbound dock doors).
- 3) Temporary storage area is unlimited.
- 4) Outbound vehicle can deliver multiple products to multiple destinations.
- 5) All outbound vehicles are available at the beginning of the planning horizon.
- 6) All inbound and outbound vehicles have same, fixed capacity.
- 7) The vehicle's speed is assumed to be constant.

The following notation is used in the formulation of the model:

- A set of orders i ; $i = 1 \dots n$.
- A set of inbound vehicles l ; $l = 1 \dots L$.
- A set of inbound dock doors f ; $f = 1 \dots F$. At inbound area, the unloading and break-down processes are performed.
- A set of outbound vehicles k ; $k = 1 \dots K$. At outbound area, the consolidation and loading processes are performed.
- A set of outbound dock doors h ; $h = 1 \dots H$.

In addition to the above notation, for ease of formulation, two dummy vehicles for both inbound and outbound sides and the nodes for cross-dock are introduced.

Two dummy inbound vehicles are inbound vehicles 0 and $L+1$ ($l = 0$ and $l = L+1$) and two dummy outbound vehicles are outbound vehicles 0 and $K+1$ ($k = 0$ and $k = K+1$). The nodes for cross-dock are denoted as node 0 and node $n+1$ ($i = 0$ and $i = n+1$).

Notations for the parameters of the model are as follows:

- tt : distance from cross-dock parking yard to inbound dock door.
- u_{il} : load of inbound vehicle l , $u_{il} = 1$ if order i is brought by inbound vehicle l ; 0, otherwise.
- PU : processing time for unloading at the cross-dock (in minutes).
- PL : processing time for loading at the cross-dock (in minutes).
- $[a_i, b_i]$: customer delivery time window of order i .
- c_{ij} : travel time between nodes i and j (in minutes).
- d_i : size of order i (in pallets).
- Q : vehicle capacity (in pallets).
- T : planning horizon (in minutes).
- α : earliness penalty cost per unit (\$ per pallet per minute).
- β : tardiness penalty cost per unit (\$ per pallet per minute).
- γ : inventory holding cost per unit (\$ per pallet per minute).
- θ : transportation cost per unit (\$ per minute).
- M : big number

There are two sets of decision variables used in this model. The first set relates to cross-dock yard and inbound area, while the second set of decision variables

relates to outbound area and customer location. The decision variables related to cross-dock yard and inbound area are (Figure 3.2):

AT_l : arrival time of inbound vehicle l ; for $l = 1 \dots L$.

RT_l : release time of inbound vehicle l ; for $l = 1 \dots L$.

rt_i : release time of order i ; for $i = 1 \dots n$.

y_{lf} : binary variables. $y_{lf} = 1$ if inbound vehicle l is processed at inbound dock door f , 0 otherwise, for $l = 0, 1, \dots, L+1$; $f = 1, \dots, F$.

x_{lj} : binary variables. $x_{lj} = 1$ if inbound vehicles l and j are processed at the same inbound dock door and l immediately precedes j , 0 otherwise, for $l = 0, 1, \dots, L$; $j = 1, \dots, L+1$; $l \neq j$.

Figure 3.2 Decision variables related to cross-dock yard and inbound area

The decision variables related to outbound area and customer location are (Figure 3.3):

- z_{ijk} : binary variable, $z_{ijk} = 1$ if outbound vehicle k travels from node i to j , 0 otherwise, for $i = 0, 1, \dots, n; j = 1, \dots, n+1; i \neq j; k = 1, \dots, K$.
- v_{ik} : binary variable, $v_{ik} = 1$ if outbound vehicle k loads order i , and 0 otherwise, for $i = 1, \dots, n; k = 1, \dots, K$.
- s_{ik} : time at which outbound vehicle k completes delivery of order i , for $i = 1, \dots, n; k = 1, \dots, K$.
- s_{0k} : departure time of outbound vehicle k (the node for cross-dock is 0), for $k = 1, \dots, K$.
- dt_i : departure time of product (order) i , for $i = 1, \dots, n$.
- Y_{kh} : binary variable, $Y_{kh} = 1$ if outbound vehicle k is processed at outbound dock door h , 0 otherwise, for $k = 0, 1, \dots, K+1; h = 1, \dots, H$.
- X_{kj} : binary variable, $X_{kj} = 1$ if outbound vehicle k and j are processed at the same outbound dock door and k immediately precedes j , 0 otherwise, for $k = 0, 1, \dots, K; j = 1, \dots, K+1; k \neq j$.
- e_i : earliness of order i , for $i = 1, \dots, n$ (in minutes).
- t_i : tardiness of order i , for $i = 1, \dots, n$ (in minutes).

Figure 3.3 Decision variables related to outbound vehicle and customer location

Figure 3.4 Decision variables of VRSP model

All decision variables of the VRSP model (fourteen decision variables) are shown in Figure 3.4. In this model the terms *order* and *product*, and *node* and *customer* are used interchangeably. All the outbound vehicles are available at time zero. It is also assumed that each node i ($i \neq 0, i \neq n+1$) is visited only once by an outbound vehicle. The delivery route for each outbound vehicle starts at cross-dock (node 0) and ends at cross-dock (node = $n+1$). M is a large constant.

The objective of the model is to minimize the total operational cost, consisting of earliness and tardiness penalty costs, inventory holding cost, and transportation cost. The earliness penalty cost depends on the size of the order and how far ahead of time it is delivered. The earliness cost per order is represented as the earliness of order i (e_i , in minutes) multiplied by the number of pallets in order i (d_i), multiplied by the unit earliness cost (α , in \$ per pallet per minute). Thus the earliness cost for all orders can be written as $\sum_{i=1}^n \alpha d_i e_i$.

Similarly, the tardiness penalty cost depends on the order size and delay in delivery; the tardiness cost for all orders can be written as $\sum_{i=1}^n \beta d_i t_i$.

The inventory holding cost depends on the inventory level at the cross-dock. In this model, the inventory level of every order is based on how long the order spends in the facility (departure time of order i minus release time of order i , $dt_i - rt_i$) multiplied by the size of the order (number of pallets in order i , d_i). The inventory holding cost of each order is then determined by multiplying the inventory level of the order with the unit inventory holding cost (γ , \$ per pallet per minute). The inventory holding cost for all orders can be written as

$$\sum_{i=1}^n \gamma d_i (dt_i - rt_i).$$

The transportation cost of each vehicle is calculated as the travel time of the vehicle multiplied by the unit transportation cost (θ , \$ per minute). The travel time of each vehicle, which is proportional to the distance travelled by the

vehicle, is calculated as $\sum_{j=1, j \neq i}^{n+1} \sum_{i=0}^n (z_{ijk} c_{ij})$. The transportation cost for all outbound

vehicles can then be written as $\sum_{k=1}^k \sum_{j=1, j \neq i}^{n+1} \sum_{i=0}^n \theta (z_{ijk} c_{ij})$.

The model is formulated as a MILP. The formulation of the VRSP model is presented next.

$$\textbf{Minimize} \quad \sum_{i=1}^n \alpha d_i e_i + \sum_{i=1}^n \beta d_i t_i + \sum_{i=1}^n \gamma d_i (dt_i - rt_i) + \sum_{k=1}^k \sum_{j=1, j \neq i}^{n+1} \sum_{i=0}^n \theta (z_{ijk} c_{ij})$$

Subject to:

$$AT_l = RT_l - tt ; l = 1, 2, \dots, L \quad (1)$$

$$RT_l \leq dt_i - (PU + PL) + M(1 - u_{il}) ; i = 1, 2, \dots, n; l = 1, 2, \dots, L \quad (2)$$

$$rt_i = \sum_{l=1}^L (RT_l \cdot u_{il}) ; i = 1, 2, \dots, n \quad (3)$$

$$\sum_{f=1}^F y_{lf} = 1 ; l = 1, 2, \dots, L \quad (4)$$

$$x_{lj} - 1 \leq y_{lf} - y_{jf} \leq 1 - x_{lj} ; l, j = 1, 2, \dots, L; l \neq j; f = 1, 2, \dots, F \quad (5)$$

$$RT_j - RT_l + M(3 - x_{lj} - y_{lf} - y_{jf}) \geq PU ; l, j = 1, 2, \dots, L; l \neq j; f = 1, 2, \dots, F \quad (6)$$

$$\sum_{l=0, l \neq j}^L x_{lj} = 1 ; j = 1, 2, \dots, L \quad (7)$$

$$\sum_{j=1, j \neq l}^{L+1} x_{lj} = 1 ; l = 1, 2, \dots, L \quad (8)$$

$$\sum_{j=1}^L x_{0j} = F \quad (9)$$

$$\sum_{l=1}^L x_{l, L+1} = F \quad (10)$$

$$x_{0l} + x_{0j} + y_{lf} + y_{jf} \leq 3 ; l, j = 1, 2, \dots, L+1; l \neq j; f = 1, 2, \dots, F \quad (11)$$

$$\sum_{k=1}^K \sum_{i=0, i \neq j}^n z_{ijk} = 1 ; j = 1, 2, \dots, n \quad (12)$$

$$\sum_{j=1}^n z_{0jk} = 1 ; k = 1, 2, \dots, K \quad (13)$$

$$\sum_{j=1}^n z_{i, n+1, k} = 1 ; k = 1, 2, \dots, K \quad (14)$$

$$\sum_{i=0, i \neq h}^n z_{ihk} - \sum_{j=1, j \neq h}^{n+1} z_{hjk} = 0 ; h = 1, 2, \dots, n; k = 1, 2, \dots, K \quad (15)$$

$$s_{ik} \leq T ; i = 1, \dots, n ; k = 1, 2, \dots, K \quad (16)$$

$$s_{jk} \geq s_{ik} + c_{ij} - M(1 - z_{ijk}) ; i = 0, 1, \dots, n; j = 1, 2, \dots, n+1; i \neq j; k = 1, 2, \dots, K \quad (17)$$

$$s_{ik} - b_i \leq t_i - e_i ; i = 1, 2, \dots, n ; k = 1, 2, \dots, K \quad (18)$$

$$a_i - s_{ik} \leq e_i - t_i ; i = 1, 2, \dots, n ; k = 1, 2, \dots, K \quad (19)$$

$$\frac{1}{M} \sum_{i=0, i \neq j}^n z_{ijk} \leq v_{jk} \leq \sum_{i=0, i \neq j}^n z_{ijk} ; j = 1, 2, \dots, n ; k = 1, 2, \dots, K \quad (20)$$

$$\sum_{i=1}^n d_i \cdot v_{ik} \leq Q ; k = 1, 2, \dots, K \quad (21)$$

$$s_{ok} \geq rt_i + (PU + PL) - M(1 - v_{ik}) ; i = 1, 2, \dots, n ; k = 1, 2, \dots, K \quad (22)$$

$$dt_i \geq s_{ok} - M(1 - v_{ik}) ; i = 1, 2, \dots, n ; k = 1, 2, \dots, K \quad (23)$$

$$rt_i \leq dt_i - (PU + PL) ; i = 1, 2, \dots, n \quad (24)$$

$$\sum_{h=1}^H Y_{kh} = 1 ; k = 1, 2, \dots, K \quad (25)$$

$$X_{kj} - 1 \leq Y_{kh} - Y_{jh} \leq 1 - X_{kj} ; k, j = 1, 2, \dots, K ; k \neq j ; h = 1, 2, \dots, H \quad (26)$$

$$s_{oj} - s_{ok} + M(3 - X_{kj} - Y_{kh} - Y_{jh}) \geq PL ; k, j = 1, 2, \dots, K ; k \neq j ; h = 1, 2, \dots, H \quad (27)$$

$$\sum_{k=0, k \neq j}^K X_{kj} = 1 ; j = 1, 2, \dots, K \quad (28)$$

$$\sum_{j=1, j \neq k}^{K+1} X_{kj} = 1 ; k = 1, 2, \dots, K \quad (29)$$

$$\sum_{j=1}^K X_{0j} = H \quad (30)$$

$$\sum_{k=1}^K X_{k, K+1} = H \quad (31)$$

$$X_{0k} + X_{0j} + Y_{kh} + Y_{jh} \leq 3 ; k, j = 1, 2, \dots, K+1 ; k \neq j ; h = 1, 2, \dots, H \quad (32)$$

$$y_{lf} \in \{0, 1\} ; l = 0, 1, \dots, L+1 ; f = 1, 2, \dots, F$$

$$x_{lj} \in \{0, 1\} ; l = 0, 1, \dots, L ; j = 1, 2, \dots, L+1 ; l \neq j$$

$$Y_{kh} \in \{0,1\} ; k = 0,1,\dots,K+1 ; h = 1,2,\dots,H$$

$$X_{kj} \in \{0,1\} ; k = 0,1,\dots,K ; j = 1,2,\dots,K+1 ; k \neq j$$

$$z_{ijk} \in \{0,1\} ; i = 0,1,\dots,n ; j = 1,2,\dots,n+1 ; i \neq j ; k = 1,2,\dots,K$$

$$v_{ik} \in \{0,1\} ; i = 1,2,\dots,n ; k = 1,2,\dots,K$$

$$AT_l, rt_i, RT_l, s_{0k}, dt_i, s_{ik}, e_i, t_i \in \mathbb{Z}_+ ; i = 1,2,\dots,n ; l = 1,2,\dots,L ; k = 1,2,\dots,K$$

The objective of the model is to minimize the total cost, consisting of four components: the earliness penalty cost and tardiness penalty cost of the orders, the inventory holding cost of the products, and the transportation cost of deliveries. The constraints of the model are divided into four sets. First set of constraints is related with inbound vehicle scheduling in inbound area of the cross-dock (constraints 1-11). The second set of constraints is related to outbound vehicle routing in delivering orders to customer locations (constraints 12-19); the third set of constraints is related to product consolidation (constraints 20-23), and the fourth set of constraints is related to outbound vehicle scheduling in outbound area of the cross-dock (constraints 24-32).

Constraint 1 determines the inbound vehicle's arrival time to the cross-dock parking yard. Constraint 2 enforces the inbound vehicle's release time; the release time of inbound vehicle l must be such that each product (order) i that is brought by vehicle l has enough time for unloading and loading processes before its departure. Constraint 3 enforces the product's release time, which is the same as the release time of the inbound vehicle that brought it. Constraint 4 ensures

that each inbound vehicle is processed at only one inbound dock door. Constraint 5 specifies that if inbound vehicle l precedes inbound vehicle j , they must be on the same inbound dock door f . Constraint 6 states that if inbound vehicle l precedes inbound vehicle j , then the release time of vehicle j must ensure that there is enough time for vehicle l to complete its release (unloading).

Constraints 7-11 are related to the dummy inbound vehicle. Constraint 7 ensures that each non-dummy inbound vehicle has exactly one preceding inbound vehicle (possibly a dummy inbound vehicle). Constraint 8 ensures that each non-dummy inbound vehicle has exactly one succeeding inbound vehicle (possibly a dummy inbound vehicle). Constraint 9 restricts dummy inbound vehicle 0 in the inbound area to be the first inbound vehicle at each inbound dock door. Constraint 10 restricts the dummy inbound vehicle ($l+1$) to be the last job at each inbound dock door. Constraint 11 specifies that if both inbound vehicle l and inbound vehicle j immediately follow inbound vehicle 0, they must be at different inbound dock doors.

Constraints 12-19 (the second set of constraints) are related to outbound vehicle routing. Constraint 12 specifies that each node (order) i is visited exactly once by an outbound vehicle. Constraint 13 ensures that each delivery route starts at the cross-dock, while constraint 14 ensures that each delivery route ends at the cross-dock. Constraint 15 restricts the sequence of the outbound vehicle's route of delivery. Constraint 16 ensures that each delivery order must be completed within the planning horizon. Constraint 17 computes the completion time of delivery order to a customer. Constraint 18 and 19 specify the earliness and tardiness of each node (order) i .

Constraints 20-23 (the third set of constraints) are related to the product consolidation decision. Constraint 20 forces v_{jk} to be 1 if the outbound vehicle k loads (delivers) order j ; this constraint captures the relation between z_{ijk} and v_{jk} . Constraint 21 restricts the outbound vehicle's load, as an outbound vehicle's load (total number of orders that are consolidated and loaded onto the outbound vehicle) cannot exceed its capacity. Constraint 22 enforces the departure time of outbound vehicle (s_{0k}). Constraint 23 captures the consequence of consolidation; as a result of consolidation, the departure time of product (order) i is the same as departure time of outbound vehicle that would carry it.

Constraints 24-32 are the fourth set of constraints. Constraint 24 restricts the departure time and the release time of product (order) i . Constraint 25 ensures that each outbound vehicle is processed at only one outbound dock door. Constraint 26 specifies that if outbound vehicle k precedes outbound vehicle j , they must be on the same outbound dock door h . Constraint 27 restricts the departure time of the outbound vehicle. If outbound vehicle k precedes outbound vehicle j , departure time of vehicle k must be such that there is enough time for vehicle j to complete its loading. Constraints 28-32 are related to dummy outbound vehicle. Constraint 28 ensures that each non-dummy outbound vehicle has exactly one preceding outbound vehicle (possibly a dummy outbound vehicle). Constraint 29 ensures that each non-dummy outbound vehicle has exactly one succeeding outbound vehicle (possibly a dummy outbound vehicle). Constraint 30 restricts the dummy outbound vehicle 0 in the outbound area to be the first outbound vehicle at each outbound dock door. Constraint 31 restricts dummy outbound vehicle $(K+1)$ to be the last job at each outbound dock door.

Constraint 32 specifies that if outbound vehicles k and j both immediately follow outbound vehicle 0 , they must be at different outbound dock doors.

3.4 TEST DATA SET GENERATION

This section describes the data set generation used for experiment of VRSP. In addition, the problem parameters and the CPLEX parameter are also determined.

3.4.1 The Data Sets

This section describes the generation of the problem instances used for the experiments in this thesis. Since the cross docking problem is relatively new, especially the focus on integrated vehicle routing and scheduling, there is no benchmark test data set available. Thus, practical datasets (based on real-life cross-dock warehouses) are generated for use in experiments.

Test data generation requires five parameters: number of inbound dock doors, number of inbound trucks, number of outbound dock doors, number of outbound trucks, and number of customers. The size of cross docks determines the number of customers that can be served which in turn determines the size of the dataset. Gue (1999), Bartholdi and Gue (2000) and Bartholdi and Gue (2004) mention that the LTL cross dock size can vary from fewer than 10 doors to more than 200; the largest cross dock in the world, in Dallas, has more than 500 doors. Trussell (2001) mentions that the percentage of receiving doors ranges between 21% to 67%. Also Gue (1999) mentions that the average destinations per trailer is around 3.7 to 6.4.

The datasets used in experiments in this thesis are based on the above information. The datasets are divided into three categories: small size, medium size, and large size, depending on the size of the warehouse. Each data set specifies the number of customers, no of inbound trucks (IT), number of inbound dock doors (IDD), number of outbound trucks (OT), and number of outbound dock doors (ODD). The three data sets are presented next.

1) Small size data-set

The small size data set is shown in Table 3.1. The details of these data sets are presented in Appendix-A

Table 3.1 Small size data sets

Problem set	No of customers	No of IT	No of IDD	No of OT	No of ODD
SS_1	2	2	2	1	1
SS_2	2	2	2	2	2
SS_3	3	2	2	1	1
SS_4	3	2	2	2	2
SS_5	4	2	2	1	1
SS_6	4	2	2	2	2
SS_7	5	2	2	1	1
SS_8	5	2	2	2	2

2) Medium size data-set

The medium size data sets are shown in Table 3.2. The details of these data sets are presented in Appendix-A.

Table 3.2 Medium size data sets

Problem set	No of customer	No of IT	No of IDD	No of OT	No of ODD
MS_1	10	2	2	2	2
MS_2	10	4	4	4	4
MS_3	20	5	5	5	5
MS_4	20	10	10	10	10
MS_5	40	10	10	10	10
MS_6	40	20	20	20	20
MS_7	60	10	10	10	10
MS_8	60	20	20	20	20

3) Large size data-sets

The large size data sets are shown in Table 3.3. The details of these data sets are presented in Appendix-A.

Table 3.3 Large size data sets

Problem set	No of customer	No of IT	No of IDD	No of OT	No of ODD
LS_1	100	20	20	20	20
LS_2	100	40	40	40	40
LS_3	200	40	40	40	40
LS_4	200	80	80	80	80
LS_5	500	100	100	100	100
LS_6	500	200	200	200	200
LS_7	900	200	200	200	200
LS_8	900	300	300	300	300

3.4.2 The Problem Parameters

The notation of VRSP problem parameters is explained in the previous section.

Here, the values of the problem parameters of VRSP are determined:

- tt : The time to travel from cross-dock parking yard to inbound dock

door is assumed as 5 minutes.

- u_{il} : u_{il} follows a uniformly random distribution in the interval of 1 to the number of inbound trucks for all the products. So $u_{il} = \text{unif}([1, \text{number of inbound truck}], 1, \text{number of orders}) = \text{unif}([1, L], 1, n)$, where L is number of inbound trucks and n in number of customers.
- PU : Wen et al. (2009) mention that the unloading time is 1 minute for each pallet. Therefore, in this experiment, it is assumed that the unloading time is 30 minutes for each truck.
- PL : Similar to the unloading time, the loading time per truck is also assumed as 30 minutes.
- $[a_i, b_i]$: customer delivery time window of order 'i' is uniformly distributed in the interval $[a_i, b_i]$. The earliest arrival time (a_i) is uniformly distributed in the interval (480, 780), while the latest arrival time (b_i) is uniformly distributed in the interval (960, 1260). So $[a_i, b_i] = [\text{unif}(480, 780), \text{unif}(960, 1260)]$.
- c_{ij} : The travel time is generated based on x and y coordinates of each node, which are randomly generated as follows $[x, y] = [\text{unif}(10, 600), \text{unif}(10, 600)]$. The travel time is then calculated based on the euclidean distance between the x and y coordinates.
- d_i : Size of order i (d_i) is uniformly distributed as $\text{unif}([4, 40], 1, i)$, while i is the number of order (customer). The number of pallets for each order is also uniformly distributed in the range of 4 to 40 pallets.
- Q : Based on Wen et al. (2009) who used vehicle capacity of 33 pallet, in these experiments, the capacity is rounded up to 40 pallets.

- T : in these experiments, the time horizon is set as 1440 minutes (equal to 24 hours) since in cross docking the products spend no more than 24 hours in the cross dock warehouse (Apte and Viswanathan, 2004 and Schaffer, 1998).

- α and β : earliness penalty cost per unit (\$ per pallet per minute) and tardiness penalty cost per unit (\$ per pallet per minute).

The unit tardiness cost is set higher than unit earliness cost since tardy delivery has more severe impact than early delivery. The impact of tardiness can be loss of future sales and customer dissatisfaction. The earliness delivery will create another cost for warehousing at the customer location. Thus in these experiments, the unit earliness cost and unit tardiness cost for each pallet per minute are specified as \$ 0.1 and \$ 0.33. These values are based on Vanhoucke (2002).

- γ and θ : inventory holding cost per unit (\$ per pallet per minute) and transportation cost per unit (\$ per minute). Chen et al. (2006) and Miao et al. (2008) state that in the cross-docking context, the inventory holding cost is usually smaller than the transportation cost, since in cross-docking the products are sent directly to the customer with little or no storage in between, usually less than 24 hours. Thus in these experiments, the unit inventory holding cost is set as \$ 0.04 per pallet per minute and unit transportation cost for each truck is set as \$ 0.6 per minute.

- M : a large number, set at 10,000,000

The summary of the VRSP problem parameters is presented in the table below.

Table 3.4 The problem parameter values

Parameter	Value
tt	5 minutes
u_{il}	$unif([1, L], 1, n)$,
PU	30 minutes
PL	30 minutes
$[a_i, b_i]$	$[unif(480, 780); unif(960, 1260)]$
c_{ij}	Euclidean distance between two nodes (node i and j), each node's coordinates are $[x,y] = [unif(10,600), unif(10,600)]$
d_i	$unif([4,40], 1, i)$,
Q	40 pallet
T	1440 minutes
α	\$ 0.1 per pallet per minute
β	\$ 0.33 per pallet per minute
γ	\$ 0.04 per pallet per minute
θ	\$ 0.6 per truck per minute
M	10,000,000

3.5 NUMERICAL EXPERIMENTS

The proposed VRSP model is solved using CPLEX Solver version 12.3 for a range of variables. The experiments are run for different number of vehicles, dock doors, and orders, on a computer platform Intel® Celeron® CPU N2840 2.16 GHz. The values of the parameters and the problem instances of data sets used in these experiments are shown in Section 3.4 and in Appendix-A.

3.5.1 CPLEX Parameters

Besides determining the problem datasets, the CPLEX Solver parameter setting also must be determined as they determine the quality of solution obtained and the computation time. The parameter settings of the CPLEX Solver used for experiment with VRSP model are discussed next.

1. Global time limit is varied based on the size of the data set. It is set as 1800 seconds, 3600 seconds, and 7200 seconds, for small scale, medium scale, and large scale data sets respectively.
2. Deterministic time limit is 1.0E75 sec.
3. Memory available for working storage is 2048.0.
4. MIP branching direction is “down branch first”.
5. MIP dive strategy: probing strategy
6. Use local branching heuristics.
7. MIP node selection strategy is “best bound search”.
8. Absolute MIP gap tolerance is 1.0E-6.
9. Relative MIP gap tolerance is 1.0E-4.
10. Number of search points in multi points is 30.

The rest of parameters are set at their default values by the CPLEX Solver.

3.5.2 Example application of VRSP

Before presenting the experiment results of VRSP, this section illustrates one example application of VRSP in determining the product delivery strategy based

on minimizing the total cost. It illustrates how VRSP model is able to determine the delivery strategy from supplier to customer via cross-dock warehouse, based on minimizing the total cost of earliness, tardiness, inventory holding, and transportation costs.

The example is taken from a small problem data set SS_7, consisting of 5 customers, 2 inbound trucks and 2 inbound dock doors, and 1 outbound truck and 1 outbound dock door. The detail problem set SS_7 is presented in Appendix-A. The demand quantities and the delivery time window of each customer is also listed in Appendix-A. The solution for dataset SS_7, using the CPLEX Solver is as follows:

Inbound truck 1 that brings order 3 and 5 arrives at cross dock yard at time 372 (inbound truck 1 arrival time is 372) while inbound truck 2 that brings orders 1, 2, and 4 arrives at cross-dock yard at time 372 also (inbound truck 2 arrival time is 372). Then inbound truck 1 goes to inbound dock door 1, unloads its products (order 3 and 5) at time 377 (the inbound truck 1 release time is 377), while inbound truck 2 goes to inbound dock door 2 and unloads its product (order 1, 2, and 4) also at time 377 (the inbound truck 2 release time is 377). After that, the product consolidation decision specified that orders 1, 2, 3, 4 and 5 are consolidated and loaded onto outbound truck 1. Then outbound truck 1 is assigned to outbound dock door 1. After finishing the loading of orders 1, 2, 3, 4 and 5 to the outbound truck 1, outbound truck 1 departs from cross-dock at time

437 (outbound truck departure time is 437), and goes on to deliver the customer orders.

The outbound truck routing 1 is as follows: start from cross-dock, go to customer 5, then to customer 1, to customer 3, customer 2, customer 4 (in that order), and finally return to cross-dock (CD-5-1-3-2-4-CD). Following this route, truck 1 departs from cross-dock at time 437, goes to customer 5 and completes the delivery at time 633; then goes to customer 1 completing the delivery at time 832; next it goes to customer 3 and completes the delivery at time 940; followed by delivery to customer 2 which is completed at time 1127. This is followed by delivery to customer 4 at time 1248. Finally truck 1 returns to cross dock. From the delivery schedule of each of its order, it can be seen that one order (order 5) is delivered earlier than its earliest arrival time. The rest of the orders are delivered on time, within their time windows. The earliness of order 5 is 44 minutes. Therefore, based on the above delivery strategy proposed by VRSP model results in minimum total cost (\$728.8), consisting of earliness cost of \$ 35.2, tardiness cost of \$ 0, inventory holding cost of \$ 72, and transportation cost of \$ 621.6.

3.5.3 Experiment Investigation

This section presents the summary of the experimental investigation of the VRSP model, conducted using the CPLEX Solver. Table 3.1, 3.2, and 3.3 show the

experimental result of VRSP for small, medium, and large size problems. For small size problems, the run time taken is very short. For medium size problems, for some data sets, the run time exceeds 3600 seconds; and for the larger data sets, the approach fails to obtain the optimal solution because of “out of memory” condition.

Table 3.5 Experiment result of VRSP for small data sets

Problem set	VRSP with CPLEX Solver	
	Total cost (\$)	Run Time (sec)
SS_1	376.8	0.77 sec
SS_2	579.6	0.28 sec
SS_3	661.8	2.11 sec
SS_4	958.2	1.39 sec
SS_5	636	0.51 sec
SS_6	808.8	3.28 sec
SS_7	728.8	1.95 sec
SS_8	1402.8	2.67 sec

Table 3.6 Experiment result of VRSP for medium data sets

Problem set	VRSP with CPLEX Solver	
	Total cost (\$)	Run Time (sec)
MS_1	1054.8	1800
MS_2	1693.2	2057
MS_3	795.26	>3600
MS_4	571.2	>3600
MS_5	4275.45	>3600
MS_6	6890.432	>3600
MS_7	No solution	
MS_8	No solution	

Table 3.7 Experiment result of VRSP for large data sets

Problem set	VRSP with CPLEX Solver	
	Total cost (\$)	Run Time (sec)
LS_1	No solution	
LS_2	No solution	
LS_3	No solution	
LS_4	No solution	
LS_5	No solution	
LS_6	No solution	
LS_7	No solution	
LS_8	No solution	

Based on the results of the computational experiments above, it can be seen that the VRSP model can be solved in a fair amount of time only for small and some medium sized problems. For larger data sets, CPLEX Solver is unable to solve the VRSP model.

The runtime increases exponentially as it is directly correlated to the number of orders, vehicles, and dock doors. In VRSP model, the number of combinations depends upon the following factors: number of orders, total number of inbound vehicles and inbound dock doors, and total number of outbound vehicles and outbound dock doors, represented as a 3-tuple $n_1/n_2/n_3$ ' n_1 ' represents the number of orders, ' n_2 ' represents the total number of inbound vehicles and inbound dock doors minus one, ' n_3 ' represents the total number of outbound vehicles and outbound dock doors minus one. Then, the total number of combinations of this model can be represented in asymptotic notation as follows: $O(n_1!n_2!n_3!)$.

As an example, for 9 orders, 2 inbound and 2 outbound trucks, and 2 inbound and 2 outbound dock doors, the total number of combinations to be explored is $O(9! 3! 3!) = 13,063,680$. As the results of the experiments above indicate, the computation time increases exponentially as the problem increases in size. To summarize, the CPLEX Solver can only solve small-sized VRSP model in a reasonable and acceptable time.

3.6 SUMMARY OF THE VRSP MODEL

The scheduling and vehicle routing problem of cross-dock facilities is modelled as an MILP. The objective of the model is to determine the schedule of inbound and outbound vehicles at the cross-dock (the arrival time and release time of inbound vehicles and the departure time of outbound vehicles) and the routing of outbound vehicles to deliver products within customer specified time windows. The objective of the model is to minimize the total cost, which consists of four components: the earliness penalty cost of the orders, the tardiness penalty cost of the orders, the inventory holding cost of the products, and the transportation cost of deliveries.

The constraints of the model are specified in four sets. First set is related to inbound truck scheduling in inbound area of the cross-dock; the second set is related to outbound truck routing in delivering orders to customer locations; the third set is related to product consolidation, and the fourth set is related to outbound truck scheduling in outbound area of the cross-dock. The model

determines the product delivery strategy that minimizes the total cost. In the next chapter, a strategy to modify and simplify the VRSP model in order to solve large-scale problems in acceptable run time is proposed. The proposed strategy applies the concept of customer zones and introduces hard time windows for delivery.

One limitation of the VRSP is that all available outbound trucks must be utilized to deliver the customer orders. For example if there are 40 customers and 10 outbound trucks available, the solution of VRSP will use all the 10 outbound trucks to deliver the 40 customer orders. The optimum number of trucks to be utilized can only be determined through experimentation.

As an example, Table 3.4 shows several experiments performed with VRSP using CPLEX Solver for the case of 12 customers with 1 inbound truck and 1 inbound dock door, while the number of outbound trucks and outbound dock doors is varied. The objective of these experiments is to determine the optimum number of outbound truck to be used for delivering to the 12 customers.

Table 3.8 Experiment to determine the optimum number of outbound trucks

No	# Customer	# IT	# IDD	# OT	# ODD	Total cost (\$)	Run time (sec)
1	12	1	1	1	1	743.46	10.47
2	12	1	1	2	2	745.01	14.80
3	12	1	1	3	3	749.41	14.72
4	12	1	1	4	4	754.79	11.57
5	12	1	1	2	1	776.89	41.78

The above experiments indicate that the setting of 1 outbound truck with 1 outbound dock door is the best setting for delivering 12 customers.

CHAPTER 4

VEHICLE ROUTING AND SCHEDULING PROBLEM WITH CUSTOMER ZONES AND HARD TIME WINDOWS

From the experimental analysis of the VRSP model (Chapter 3) it is shown that the model is not able to solve real-life size problems; most cross-dock warehouses have dozens of inbound and outbound dock-doors and handle hundreds of orders in a day. An alternative solution is needed to handle the real-life complexity inherent in cross-dock warehouse operations. This chapter discusses a simplification strategy to modify the VRSP model developed in Chapter 3 and facilitate development of operations policies for real-life sized cross-dock warehouses. The modified model incorporates customer zones and hard-time windows to eliminate late and early delivery costs and reduce the solution space. The model is denoted as Vehicle Routing and Scheduling Problem of Cross-Dock with Customer Zones and Hard Time Windows (VRSP-CZHTW). Through numerical experiments with the VRSP-CZHTW model, it is shown that the modified model can facilitate the solution of large scale, real-life problems within an acceptable time.

4.1 THE SIMPLIFICATION STRATEGY

The proposed strategy to simplify the VRSP model consists of two main components. These are:

1. Adoption of customer zones (to reduce the number of routing candidates), and
2. Consider hard time windows of delivery (to eliminate the earliness and tardiness costs)

Adoption of those simplified strategies can reduce the solution space of the problem tremendously. The schema for the modified model (VRSP-CZHTW) is shown in Figure 4.1.

Figure 4.1 Schema for VRSP-CZHTW

The concept of customer zone for vehicle routing problem has been mentioned before but in a simplified context. For example, Crainic et al. (2012) proposed multi zone and multi trip concept for their vehicle routing problem with time window (VRPTW) model. However the routes within a zone were fixed as they were school bus routes.

In VRSP-CZHTW model, the customers are clustered based on their closeness to each other. In the model, determination of the customer zone (clustering of the customers) is performed using K-Means clustering method in MATLAB. The output of K-Means clustering is the customer zones along with the customers that belong to each zone. The result of the K-Means clustering becomes the input to the VRSP-CZHTW model.

Application of the customer zone concept reduces the size of solution space since each route only needs to consider the customers within a zone. This effectively breaks down the routing problem into multiple but smaller-size problems. Application of the customer zone is reflected in the objective function and constraints of the VRSP-CZHTW model.

In addition, the outbound trucks in VRSP-CZHTW model are restricted to serve only one customer zone and are not allowed to crossover into another zone. Thus, the delivery route of an outbound truck is restricted within one customer zone. However, depending upon the number of customers and size of the orders, one customer zone may need to be served by more than one outbound truck.

The hard time windows are also applied to reduce the size of the solution space and reflect a tactical decision by the company. When a company applies hard time-windows for deliveries and does not allow any delivery outside of the stipulated time-window, all deliveries must take place within the window without any earliness or tardiness. In the case of soft time windows, in contrast, the deliveries can take place outside of the time window, but incur earliness or tardiness penalty. In VRSP-CZHTW model, the earliness and tardiness costs are, by definition, zero and can be removed from the model objective. Wen et al. (2009) also use hard time window in their model (vehicle routing problem of cross-dock). They apply hard time window for delivery and make it a constraint of the model.

Therefore, in VRSP-CZHTW, besides considering the closeness of the location, determination of the customer zone must also consider the information of delivery time window. Customers i and j can only be in one customer zone if they are located closer to each other and their delivery time windows are in the same time period. For example, if customers i and j are located next to each other they cannot be in the same zone for purposes of delivery if customer i requests delivery in the morning (7am-11am) and customer j asks for delivery order in the evening (4pm-6pm). Thus to be considered in the same delivery zone, the customers must not only be in close proximity, but their delivery time windows must also overlap (or be close to each other).

4.1.1 Determination of the Customer Zones

As mentioned in the previous section in VRSP-CZHTW, determination of the customer zone does not only consider the closeness of distance, but also the closeness of the time windows. The closeness of time windows is considered in determination of customer zones in order to avoid infeasible solutions. As long as the customer orders are clustered around similar delivery windows, the orders are likely to be delivered on time.

Let's say the time window of customer 1 is in the morning from 6 am to 9 am while customer 2's time window is in the evening from 3 pm to 6 pm. So in this scenario, even if customers 1 and 2 are close by they cannot be clustered into one zone. So the customer orders have to be placed in different zones in order to be able to deliver the orders during their delivery windows.

Thus, the determination of customer zones must consider the closeness of distance and also the closeness of the time window. In VRSP-CZHTW, the determination of the customer zones considers the distance first (in this experiment, K-Means clustering method is used), which is followed by a check of time window overlap.

In this research, the degree of overlap between the time windows is not yet studied. As long as the customers have overlapping delivery windows, they can be clustered together. Figure 4.2 illustrates how the determination of customer zone is affected by the customer location and time window.

Figure 4.2 Effect of time window on determination of customer clustering

4.2 VRSP-CZHTW MODEL

For the application of customer zones and hard time windows, the VRSP-CZHTW model now incorporates 6 indices in the notation. The six indices are i for customer, l for inbound truck, k for outbound truck, f for inbound dock door, h for outbound dock door, and m for customer zone. In this model, similar to the VRSP model, at the cross-dock inbound yard, the manager determines the schedule for inbound vehicle arrival time to the cross-docking warehouse (AT_i). At the inbound area, when the vehicle arrives and is parked at the cross-docking yard, the cross-dock manager specifies:

- (1) release time of inbound vehicles and orders (RT_l and rt_{i_m}),
- (2) sequence of inbound vehicles at inbound dock door (x_{lj}), and
- (3) assignment of inbound vehicle to an inbound dock door (y_{lj}).

After the products have been unloaded, the following decisions related to outbound area and customer location need to be made:

- (1) product consolidation ($v_{i_m k_m}$);
- (2) scheduling of outbound vehicles, consisting of:
 - a. departure time of outbound vehicles ($s_{0_m k_m}$),
 - b. sequence of outbound vehicles at outbound dock door ($X_{k_m j_m}$),
 - c. assignment of outbound vehicles to outbound dock door ($Y_{k_m h_m}$)
- (3) routing of outbound vehicles ($z_{i_m j_m k_m}$), and
- (4) completion time of order delivery ($s_{i_m k_m}$).

Once these decisions have been made, the inventory holding cost and outbound transportation costs can be computed.

VRSP-CZHTW model simplification strategy makes three key differences in the model with respect to underlying VRSP model. These are:

1. The objective of the VRSP-CZHTW model is to minimize only the transportation and inventory holding costs, as the earliness and tardiness costs are now zero.
2. Customer zone is an input into the VRSP-CZHTW model.
3. In VRSP-CZHTW model, the assignment of outbound truck and outbound dock door to each customer zone is a tactical decision of the user, ensuring

that there are sufficient numbers of outbound trucks and outbound dock doors available to serve the orders in each customer zone and satisfying the customer's hard time window constraint.

The following assumptions are used in the development of the VRSP-CZHTW model:

- 1) Cross-dock yard space is unlimited.
- 2) Every inbound and outbound dock door is used for unloading and loading (no empty inbound and outbound dock doors).
- 3) Temporary storage area is unlimited.
- 4) Outbound vehicle can deliver multiple products to multiple destinations.
- 5) All outbound vehicles are available at the beginning of the planning horizon.
- 6) All inbound and outbound vehicles have same, fixed capacity.
- 7) Each vehicle's speed is assumed to be constant.
- 8) Customer delivery windows are strictly enforced.

With these assumptions, the mathematical model of VRSP-CZHTW is formulated as a MILP which is presented next.

The following notation (index) is used in the formulation of the VRSP-CZHTW model:

- A set of customer zones (m); $m = 1, \dots, M$.
- A set of orders i of customer zone m (i_m); $i_m = 1, \dots, n_m$ (a set of orders i_m starts from 1 to last customer of zone m).
- A set of inbound vehicles (l); $l = 1, \dots, L$.

- A set of inbound dock doors (f); $f=1,\dots,F$. At inbound area, the unloading and break-down processes are performed.
- A set of outbound vehicles k dedicated to customer zone m (k_m); $k_m = 1,\dots,K_m$; $m = 1,\dots,M$. At outbound area, the consolidation and loading processes are performed.
- A set of outbound dock doors (h) dedicated to customer zone m (h_m); $h_m = 1,\dots,H_m$; $m = 1,\dots,M$.

Similarly to the VRSP model, for ease of formulation the dummy vehicles for both inbound and outbound sides and two nodes for cross-dock are introduced. These are inbound vehicles 0 and $L+1$ ($l = 0$ and $l = L+1$), outbound vehicles 0 and K_m+1 ($k_m = 0$ and $k_m = K_m+1$) and node 0 and node $n+1$ ($i = 0$ and $i = n+1$).

Parameters of the VRSP-CZHTW model are:

- tt : distance from cross-dock parking yard to inbound dock door
- $u_{i_m l}$: load of inbound vehicle l which is $u_{i_m l} = 1$ if order i_m of customer zone m is brought by inbound vehicle l ; 0 otherwise
- PU : processing time for unloading at cross-dock (in minutes)
- PL : processing time for loading at cross-dock (in minutes)
- $[a_{i_m}, b_{i_m}]$: customer delivery time window of customer i_m of zone m
- $c_{i_m j_m}$: travel time between node i_m and j_m of zone m (in minutes)
- d_{i_m} : size of order i_m of zone m (in pallets)
- Q : vehicle capacity (in pallets)
- T : time planning horizon (in minutes)
- γ : inventory holding cost per unit (\$ per pallet per minute)

- θ : transportation cost per unit (\$ per minute)
- G : a large number

There are twelve decision variables in the VRSP-CZHTW model. Those decision variables can be divided into two sets: the first set of decision variables relates to cross-dock yard and inbound area, and the second set to outbound area and customer location.

The decision variables related to cross-dock yard and inbound area are:

- AT_l : arrival time of inbound vehicle l , for $l = 1, \dots, L$.
- RT_l : release time of inbound vehicle l , for $l = 1, \dots, L$.
- rt_{i_m} : release time of order i_m of customer zone m , for $i_m = 1, \dots, n_m$; $m = 1, \dots, M$.
- y_{lf} : binary variables. $y_{lf} = 1$ if inbound vehicle l is processed at inbound dock door f ; 0 otherwise, for $l = 0, 1, \dots, L+1$; $f = 1, \dots, F$.
- x_{lj} : binary variables. $x_{lj} = 1$ if inbound vehicles l and j are processed on the same inbound dock door and l immediately precedes j ; 0 otherwise, for $l = 0, 1, \dots, L$; $j = 1, \dots, L+1$; $l \neq j$.

The decision variables related to outbound area and customer location are:

- $z_{i_m j_m k_m}$: binary variable, $z_{i_m j_m k_m} = 1$ if outbound vehicle k_m travels from node i_m to j_m at customer zone m , and 0 otherwise, for $i_m = 0, 1, \dots, n_m$; $j_m = 1, \dots, n_m+1$; $i_m \neq j_m$; $k_m = 1, \dots, K_m$; $m = 1, \dots, M$.
- $v_{i_m k_m}$: binary variable, $v_{i_m k_m} = 1$ if outbound vehicle k_m loads order i_m of customer zone m ; 0 otherwise, for $i_m = 1, \dots, n_m$; $k_m = 1, \dots, K_m$; $m = 1, \dots, M$.
- $s_{i_m k_m}$: time at which outbound vehicle k_m completes delivery of order i_m of

customer zone m , for $i_m = 1, \dots, n_m$; $k_m = 1, \dots, K_m$. When $i_m = n_m + 1$ it becomes s_{n_m+1, k_m} , this is the time of outbound vehicle k_m goes back to cross-dock after completing its deliveries (finishes delivery order), for $k_m = 1, \dots, K_m$; $m = 1, \dots, M$.

s_{0, k_m} : departure time of outbound vehicle k_m (the node for cross-dock is 0), for $k_m = 1, \dots, K_m$; $m = 1, \dots, M$.

dt_{i_m} : departure time of product (order) i_m of customer zone m , for $i_m = 1, \dots, n_m$; $m = 1, \dots, M$.

Y_{k_m, h_m} : binary variable, $Y_{k_m, h_m} = 1$ if outbound vehicle k_m is processed on outbound dock door h_m ; 0 otherwise, for $k_m = 0, 1, \dots, K_m + 1$; $h_m = 1, \dots, H_m$.

X_{k_m, j_m} : binary variable, $X_{k_m, j_m} = 1$ if outbound vehicle k_m and j_m are processed on the same outbound dock door and k_m immediately precedes j_m ; 0 otherwise, for $k_m = 0, 1, \dots, K_m$; $j_m = 1, \dots, K_m + 1$; $k_m \neq j_m$.

4.3 VRSP-CZHTW MODEL FORMULATION

As mentioned in section 4.2, the VRSP-CZHTW model uses six indices: there is one additional index for customer zone (m). In addition, because of application of hard time windows, the earliness and tardiness costs are eliminated from the objective function. Further, as a consequence of those simplification strategies, some constraints are also modified. The VRSP-CZHTW model is also formulated as a MILP. The complete formulation of the VRSP-CZHTW model is presented next.

$$\text{Minimize } \sum_{m=1}^M \sum_{i_m=1}^{n_m} \gamma d_{i_m} (dt_{i_m} - rt_{i_m}) + \sum_{m=1}^M \sum_{k_m=1}^{K_m} \sum_{j_m=1, j_m \neq i_m}^{n_m+1} \sum_{i_m=0}^{n_m} \theta(z_{i_m j_m k_m} c_{i_m j_m})$$

Subject to:

$$AT_l = RT_l - tt ; l = 1, 2, \dots, L \quad (1)$$

$$\sum_{f=1}^F y_{lf} = 1 ; l = 1, 2, \dots, L \quad (2)$$

$$x_{lj} - 1 \leq y_{lf} - y_{jf} \leq 1 - x_{lj} ; l, j = 1, 2, \dots, L ; l \neq j ; f = 1, 2, \dots, F \quad (3)$$

$$RT_j - RT_l + G(3 - x_{lj} - y_{lf} - y_{jf}) \geq PU ; l, j = 1, 2, \dots, L ; l \neq j ; f = 1, 2, \dots, F \quad (4)$$

$$\sum_{l=0, l \neq j}^L x_{lj} = 1 ; j = 1, 2, \dots, L \quad (5)$$

$$\sum_{j=1, j \neq l}^{L+1} x_{lj} = 1 ; l = 1, 2, \dots, L \quad (6)$$

$$\sum_{j=1}^L x_{0j} = F \quad (7)$$

$$\sum_{l=1}^L x_{l, L+1} = F \quad (8)$$

$$x_{0l} + x_{0j} + y_{lf} + y_{jf} \leq 3 ; l, j = 1, 2, \dots, L+1 ; l \neq j ; f = 1, 2, \dots, F \quad (9)$$

$$RT_l \leq dt_{i_m} - (PU + PL) + G(1 - u_{i_m l}) ; i_m = 1, \dots, n_m ; l = 1, \dots, L ; m = 1, \dots, M \quad (10)$$

$$\sum_{h_m=1}^{H_m} Y_{k_m h_m} = 1 ; k_m = 1, 2, \dots, K_m \quad (11)$$

$$X_{k_m j_m} - 1 \leq Y_{k_m h_m} - Y_{j_m h_m} \leq 1 - X_{k_m j_m} ; k_m, j_m = 1, \dots, K_m ; k_m \neq j_m ; h_m = 1, \dots, H_m \quad (12)$$

$$s_{0_m j_m} - s_{0_m k_m} + G(3 - X_{k_m j_m} - Y_{k_m h_m} - Y_{j_m h_m}) \geq PL ; k_m, j_m = 1, \dots, K_m ; k_m \neq j_m ;$$

$$h_m = 1, \dots, H_m \quad (13)$$

$$\sum_{k_m=0, j_m \neq k_m}^{K_m} X_{k_m j_m} = 1 ; j_m = 1, 2, \dots, K_m \quad (14)$$

$$\sum_{j_m=1, j_m \neq k_m}^{K_m+1} X_{k_m j_m} = 1 ; k_m = 1, 2, \dots, K_m \quad (15)$$

$$\sum_{j_m=1}^{K_m} X_{0 j_m} = H_m \quad (16)$$

$$\sum_{k_m=1}^{K_m} X_{k_m, K_m+1} = H_m \quad (17)$$

$$X_{0 k_m} + X_{0 j_m} + Y_{k_m h_m} + Y_{j_m h_m} \leq 3 ; k_m, j_m = 1, \dots, K_m + 1 ; k_m \neq j_m ; h_m = 1, \dots, H_m \quad (18)$$

$$rt_{i_m} = \sum_{l=1}^L (RT_l \cdot u_{i_m l}) ; i_m = 1, 2, \dots, n_m ; m = 1, \dots, M \quad (19)$$

$$\frac{1}{G} \sum_{i_m=0, i_m \neq j_m}^{n_m} z_{i_m j_m k_m} \leq v_{j_m k_m} \leq \sum_{i_m=0, i_m \neq j_m}^{n_m} z_{i_m j_m k_m} ; j_m = 1, \dots, n_m ; k_m = 1, \dots, K_m ;$$

$$m = 1, \dots, M \quad (20)$$

$$\sum_{i_m=1}^{n_m} d_{i_m} \cdot v_{i_m k_m} \leq Q ; m = 1, \dots, M ; k_m = 1, \dots, K_m \quad (21)$$

$$rt_{i_m} + (PU + PL) - (G(1 - v_{i_m k_m})) \leq s_{0_m k_m} ; i_m = 1, \dots, n_m ; m = 1, \dots, M ;$$

$$k_m = 1, \dots, K_m \quad (22)$$

$$s_{0_m k_m} - (G(1 - v_{i_m k_m})) \leq dt_{i_m} ; i_m = 1, \dots, n_m ; m = 1, \dots, M ; k_m = 1, \dots, K_m \quad (23)$$

$$s_{0_m k_m} = s_{n+1_m k_m} - \sum_{j_m=1, j_m \neq i_m}^{n_m+1} \sum_{i_m=0}^{n_m} z_{i_m j_m k_m} c_{i_m j_m} ; i_m = 0, 1, \dots, n_m ; j_m = 1, \dots, n_m + 1 ; i_m \neq j_m ;$$

$$m = 1, \dots, M ; k_m = 1, \dots, K_m \quad (24)$$

$$s_{0_m k_m} \geq s_{j_m k_m} - c_{0_m j_m} - (G(1 - z_{0_m j_m k_m})); j_m = 1, \dots, n_m + 1; m = 1, \dots, M; \\ k_m = 1, \dots, K_m \quad (25)$$

$$s_{i_m k_m} \leq T; i_m = 1, \dots, n_m; m = 1, \dots, M; k_m = 1, \dots, K_m \quad (26)$$

$$\sum_{j_m=1}^{n_m} z_{0_m j_m k_m} = 1; m = 1, \dots, M; k_m = 1, \dots, K_m \quad (27)$$

$$\sum_{i_m=1}^{n_m} z_{i_m, n+1_m, k_m} = 1; m = 1, \dots, M; k_m = 1, \dots, K_m \quad (28)$$

$$\sum_{k_m=1}^{K_m} \sum_{i_m=0, i_m \neq j_m}^{n_m} z_{i_m j_m k_m} = 1; j_m = 1, \dots, n_m; m = 1, \dots, M \quad (29)$$

$$\sum_{i_m=0; i_m \neq g_m}^{n_m} z_{i_m g_m k_m} - \sum_{j_m=1; j_m \neq g_m}^{n_m+1} z_{g_m j_m k_m} = 0; g_m = 1, \dots, n_m; m = 1, \dots, M; k_m = 1, \dots, K_m \quad (30)$$

$$s_{i_m k_m} \geq a_{i_m} \cdot v_{i_m k_m}; i_m = 1, \dots, n_m; m = 1, \dots, M; k_m = 1, \dots, K_m \quad (31)$$

$$s_{i_m k_m} \leq b_{i_m} \cdot v_{i_m k_m}; i_m = 1, \dots, n_m; m = 1, \dots, M; k_m = 1, \dots, K_m \quad (32)$$

$$y_{lf} \in \{0, 1\}; l = 0, 1, \dots, L+1; f = 1, \dots, F$$

$$x_{lj} \in \{0, 1\}; l = 0, 1, \dots, L; j = 1, \dots, L+1; l \neq j$$

$$Y_{k_m h_m} \in \{0, 1\}; k_m = 0, 1, \dots, K_m + 1; h_m = 1, \dots, H_m; m = 1, \dots, M$$

$$X_{k_m j_m} \in \{0, 1\}; k_m = 0, 1, \dots, K_m; j_m = 1, \dots, K_m + 1; k_m \neq j_m$$

$$z_{i_m j_m k_m} \in \{0, 1\}; i_m = 0, 1, \dots, n_m; j_m = 1, \dots, n_m + 1; i_m \neq j_m; k_m = 1, \dots, K_m; m = 1, \dots, M$$

$$v_{i_m k_m} \in \{0, 1\}; i_m = 1, \dots, n_m; k_m = 1, \dots, K_m; m = 1, \dots, M$$

$$AT_l, rt_{i_m}, RT_l, s_{0_m k_m}, dt_{i_m}, s_{i_m k_m} \in \mathbb{Z}_+; i_m = 1, \dots, n_m; l = 1, \dots, L; k_m = 1, \dots, K_m;$$

$$m = 1, \dots, M$$

In VRSP-CZHTW model, the objective is to minimize total cost, which now consists only of two components: the inventory holding cost of the products and the transportation cost of delivery. Constraint 1 determines the inbound truck's arrival time to the cross-dock. It depends on the inbound truck's release time and the distance between the inbound dock doors and the cross-dock yard. Constraints 2-9 determine inbound truck scheduling and assignment at inbound dock door.

Constraint 10 states that determination of the release time and departure time must accommodate the required times for unloading and loading the order. Constraints 11-18 determine the outbound truck scheduling and assignment at outbound dock door. Constraint 19 relates the release time of product order i_m for zone m and release time of inbound truck. Constraint 20 presents the relationship between product consolidation decision $v_{i_m k_m}$ and outbound truck routing $z_{i_m j_m k_m}$. Constraint 21 limits the outbound truck load so that the consolidated order for a truck does not exceed its capacity.

Constraints 22-23 present the relationship between the release time, departure time, and product consolidation decision. Constraints 24-26 relate outbound truck departure time and the completion time of the outbound truck in delivering the orders until goes back to cross-dock ($s_{0_m k_m}$ and $s_{i_m k_m}$). Constraints 27-30 determine the outbound truck route $z_{i_m j_m k_m}$. Constraints 31 and 32 represent the consequence of applying hard time windows. These constraints ensure that all orders are delivered no later than the end of time window (b_{i_m}) and no earlier than the beginning of time window (a_{i_m}). So the delivery must happen within the

delivery time window. Else, an infeasible solution might result. Thus, the customer delivery windows are imposed by way of constraints 31 and 32, while the rest of the constraints of VRSP-CZHTW are similar to the VRSP model (except constraints 18 and 19). Also in VRSP-CZHTW model, an extra index (m) is required to specify the customer zone.

4.4 VRSP-CZHTW NUMERICAL EXPERIMENTS

In this section, the experiment results of the VRSP-CZHTW model are discussed; the model is solved using CPLEX Solver version 12.3, on a computer platform Intel® Celeron® CPU N2840 2.16 GHz.

To validate the VRSP-CZHTW, the experiments are performed using the same data sets (problem instances), same problem parameters, and same CPLEX parameter setting as the experiment in VRSP, already presented in section 3.4 and in Appendix-A. The VRSP-CZHTW experimental results performed using the same problem sets, problem parameters, and the CPLEX parameter setting with VRSP experiments, are presented in this section.

4.4.1 Example application of VRSP-CZHTW

This section describes the example application of VRSP-CZHTW in determining the product delivery strategy which is based on minimizing the total delivery cost. This section illustrates how VRSP-CZHTW model is able to determine the delivery strategy from suppliers to customers via cross-dock warehouse by

minimizing the total cost of inventory holding and transportation costs; the earliness and tardiness costs are now zero since VRSP-CZHTW uses hard time window constraints.

The example is based on the SS_7 data set presented in section 3.4.1 (example application for VRSP). There are 5 customers, 2 inbound dock doors, 2 inbound trucks, 1 outbound dock doors, and 1 outbound trucks. The detail data set SS_7 and the problem parameters are presented in Appendix-A, and in section 3.4, while the CPLEX parameters settings used in VRSP-CZHTW are presented in section 3.4.1.

In VRSP-CZHTW, firstly, the customer zone must be determined using K-Means clustering method of MATLAB. The result of K-Means clustering of SS_7 is presented in table below.

Customer Zone	Customer
1	1,2,3,4,5

Based on the table above, it is shown that there is only one customer zone, zone 1 with all the 5 customers clustered into this single zone.

This customer zone now becomes the input of VRSP-CZHTW. Based on the input data of SS_7, the solution proposed by VRSP-CZHTW is as follows:

Inbound truck 1, which brings customer orders 3 and 5, arrives at the cross dock yard at time 205.5 (inbound truck 1 arrival time is 205.5). The truck is assigned to inbound dock door 1, unloads its products at time 210.5 (the inbound truck 1 release time is 210.5).

The inbound truck 2, which brings customer orders 1, 2, and 4, arrives at cross dock yard at time 416 (arrival time of inbound truck 2 is 416). Then the inbound

truck 2 is assigned to inbound dock door 2 to unload its products. The inbound truck 2 release time is 421.

After the unloading is complete, the product consolidation decision of VRSP-CZHTW specifies that orders 1, 2, 3, 4, and 5 are consolidated together and loaded to outbound truck 1. The VRSP-CZHTW also determines the route of outbound truck 1 so that all the orders can be delivered on time and with minimum inventory and transportation costs.

Outbound truck 1, which is carrying orders 1, 2, 3, 4, and 5, loads the products, and then departs from cross-dock to customers at time 481. The outbound truck 1 route decision, based on VRSP-CZHTW, is CD-5-4-1-3-2-CD. So, after the outbound truck 1 departs from cross-dock at time 481, it then goes to customer 5 and completes the delivery at time 677. After that the truck goes to customer 4 and completes the delivery at time 846. Following that the truck goes to customer 1 to deliver the order and completes at time 974. Then the truck goes to customer 3 and completes the delivery at time 1082, following which the truck goes to customer 2 and completes the delivery at time 1269 before returning to cross-dock.

The outcome of VRSP-CZHTW model is slightly different from VRSP, but ensures that all the orders are delivered on time, inventory holding cost is \$ 156.2 and the transportation cost is \$ 673.2, with the total cost of \$ 829.4

4.4.2 Experiment Results of VRSP-CZHTW

By modifying the original model through the introduction of customer zones and hard time windows, the number of combinations that needs to be searched is

reduced tremendously, leading to much lower computation requirements. Next we conduct a series of experiments and solve the model for different number of vehicles, orders and dock-doors.

As already mentioned in the previous section, the VRSP-CZHTW is tested using the same data sets and problem parameters with the VRSP. Therefore, the numerical experiments of VRSP-CZHTW are performed using the data sets for small, medium, and large scale datasets, and problem parameters as shown in Chapter 3 and in Appendix-A.

As already mentioned in the previous section, firstly the clustering of customer zones needs to be determined. Customer clustering is done using K-Means clustering method in MATLAB. The elapsed run time for the K-Means clustering are reported in Table 4.1.

Table 4.1 The elapsed run time for K-Means clustering with MATLAB

Problem set	Elapsed run time (sec)
SS_1	0.10
SS_2	0.10
SS_3	0.13
SS_4	0.12
SS_5	0.12
SS_6	0.09
SS_7	0.05
SS_8	1.13
Problem set	Elapsed run time (sec)
MS_1	0.05
MS_2	0.06
MS_3	0.12
MS_4	0.12
MS_5	0.18
MS_6	0.18
MS_7	0.23
MS_8	0.23

**VEHICLE ROUTING AND SCHEDULING PROBLEM WITH
CUSTOMER ZONES AND HARD TIME WINDOWS**

Problem set	Elapsed run time (sec)
LS_1	9.14
LS_2	9.14
LS_3	17.25
LS_4	17.25
LS_5	31.62
LS_6	31.62
LS_7	156.55
LS_8	156.55

Thus K-Means clustering can be done in an acceptable time even for large scale problems. Based on the result of K-Means clustering, the customer zones are input into the VRSP-CZHTW coded in CPLEX Solver.

The numerical experiments are performed using the same computer platform as experiments of VRSP. The problems are solved using CPLEX Solver version 12.3. The results of the VRSP-CZHTW experiments are shown in Table 4.2-4.4.

Table 4.2 Experiment result of VRSP-CZHTW with CPLEX Solver
(small data sets)

Problem set	VRSP-CZHTW with CPLEX Solver	
	Total cost (\$)	Run Time (sec)
SS_1	376.8	0.02
SS_2	579.6	0.02
SS_3	661.8	0.30
SS_4	958.2	0.38
SS_5	636	0.42
SS_6	983.56	1.72
SS_7	829.4	1.41
SS_8	1402.8	0.47

**VEHICLE ROUTING AND SCHEDULING PROBLEM WITH
CUSTOMER ZONES AND HARD TIME WINDOWS**

Table 4.3 Experiment result of VRSP-CZHTW with CPLEX Solver
(medium data sets)

Problem set	VRSP-CZHTW with CPLEX Solver	
	Total cost (\$)	Run Time (sec)
MS_1	1456.5	3.69
MS_2	2048.16	26.56
MS_3	882.32	676.8
MS_4	1158.86	691.3
MS_5	4750.5	782.1
MS_6	7489.6	863.3
MS_7	7657.2	837.8
MS_8	12591.6	1824.8

Table 4.4 Experiment result of VRSP-CZHTW with CPLEX Solver
(large data sets)

Problem set	VRSP-CZHTW with CPLEX Solver	
	Total cost (\$)	Run Time (sec)
LS_1	16869.6	>7200
LS_2	15598.4	>7200
LS_3	24184.8	>7200
LS_4	38648.7	>7200
LS_5	No solution	
LS_6	No solution	
LS_7	No solution	
LS_8	No solution	

The experimental results show that the model can now be solved for fairly large and real-life problems in a reasonable amount of time. However for the large datasets with more than 500 customers, VRSP-CZHTW fails to find a solution within the giving global time limit of 7200 seconds. Therefore, the model shows promise in being used in real life for developing distribution strategies involving a cross dock warehouse.

4.5 SUMMARY OF VRSP-CZHTW MODEL

In this chapter, a simplification strategy incorporating customer zones and hard time windows is applied to the VRSP model developed in the previous chapter. The results of the VRSP-CZHTW experiments show that the simplification strategy is successful in simplifying the VRSP model. The modified model (VRSP-CZHTW), based on the experimental results, can solve the vehicle scheduling and routing problem from small to large-scale problems in acceptable time.

While the simplification strategy allows for solving the cross-dock vehicle scheduling and routing problem in reasonable time even for large size problems, it is based on assumptions that might not be met by all real-life systems.

For example, in this research, the original model (VRSP) uses soft time windows and does not enforce customer zones. As the consequence, the mathematical model (the objective function and the constraints) of VRSP-CZHTW does have some differences from the original model.

In the next chapter, further research is conducted and the original model is solved without enforcing the constraints considered in this chapter. Instead, a meta-heuristic approach is investigated that may be able to solve the model in reasonable time and provide close-to-optimal solutions.

As a limitation of the VRSP-CZHTW model, the degree of overlap between the time windows is not studied in this research. This topic will be an interesting topic for future research. This may become crucial in cases where customers

request tight time windows. But in cases where customer time windows are loose (deliver anytime on a specific day, or anytime in the afternoon), the degree of overlap should not have a big effect on the solution quality.

CHAPTER 5

**TABU SEARCH BASED ALGORITHM FOR VEHICLE ROUTING
AND SCHEDULING**

As discussed in Chapter 3, VRSP model can be solved using CPLEX Solver in an acceptable time for only small-scale problems; for medium size problems the time taken is rather long and for large size problems the system fails to find a solution. In chapter 4, an approach to solve the larger size VRSPs in a reasonable time, by reducing the solution space, was presented. That approach enforces customer delivery time windows and utilizes customer zones to limit the search space. However, for large (real-life) size problems, even that approach fails to solve the VRSP in a reasonable amount of time.

In this chapter, an alternate approach, based on meta-heuristics is employed to solve the original VRSP for real-life sized problems, which, as discussed earlier, are NP hard. Of all the approaches considered in the literature, Tabu Search seems to be the most promising to solve the VRSP and is explored in this research.

Firstly in section 5.1 a justification for employing TS as the meta-heuristics approach for solving VRSP is presented, followed by development of TS-based algorithm for solving VRSP. This is followed by a discussion of the experiments conducted with the TS-based algorithm, along with its comparison with the two methods presented earlier (VRSP and VRSP-CZHTW with CPLEX Solver) is

presented. The chapter ends with a summary of the TS algorithm and its efficacy in solving large-size VRSP.

5.1 TABU SEARCH

This section describes a basic TS algorithm, followed by a justification of the suitability of TS for scheduling and routing problems.

5.1.1 Basic TS Algorithm

TS was developed by Fred W. Glover in 1986. It is a metaheuristic method based on local search that is used for mathematical optimization. The basic principle of TS is to pursue local search when it encounters local optima by allowing non-improving moves. According to Glover (1990) and Glover et al. (1993), a chief mechanism for exploiting short term memory in TS is to classify a subset of the moves in a neighborhood as forbidden (tabu). It prevents the chosen moves to be reversed (repeated) in order to escape from local optima and prevent cycling at one particular iteration.

A tabu list T with a certain size is maintained to serve as a record of the tabu moves. A tabu list can be represented as a queue that works on the basis of "first in first out". Every time a new move enters the tabu list, it is placed at the head of the list. Once the length of the list reaches a predefined size, the oldest tabu move from the tail of the list is removed and the new move is placed at the head

of the list. Each tabu move is removed from the tail of the tabu list after certain number of iterations, determined by the tabu list size.

The size of a tabu list plays an important role in controlling the search process and affects the final search results. A small tabu list size implies that more work is desired on the local search around the current solution; however, this may trap the search routine in a cyclic pattern. A large tabu list size implies that more regions of the solution space can be explored; this, on the other hand, may deteriorate the quality of the solutions found within a reasonable cost and time.

When a tabu move is found to result in a solution better than the best solution found so far, its tabu classification can be overridden if it satisfies conditions called aspiration criteria. Such criteria are introduced in TS to determine when tabu restrictions can be overridden and temporarily release a solution from its tabu status so as to increase the flexibility of the algorithm.

At each iteration, during move selection, a move is considered forbidden (tabu) if it is already in the tabu list. If the best move, in terms of the objective function value, is not tabu or if it is tabu but satisfies the aspiration criteria, then it is selected, else the second best non-tabu move is. When a move is picked, it is also added to the tabu list so that it is not reversed or repeated over the next few iterations (Glover, 1990; Glover et al., 1993). The TS algorithm, in its simple form, is illustrated in Figure 5.1.

Figure 5.1 Simple tabu search (Glover, 1990)

For effective application of TS method, fine tunings of TS parameters, such as search strategy, stopping rules, initial solution, length of tabu list, maximum number of iterations and the level of aspiration, are required. In some applications, the short term memory of TS is sufficient to produce very high quality solutions. TS method can be made significantly stronger by including longer term memory which would ensure that the entire solution space is adequately explored before the search is completed. There are two strategies to employing longer term memory: intensification and diversification.

Intensification strategies are based on modifying choice rules to encourage move combinations and solution features historically found to be good. They focus the search in regions containing previously found good solutions. They may also initiate a return to attractive regions to search them more thoroughly.

Diversification strategies, on the other hand, are based on modifying choice rules to bring attributes into the solution that are infrequently used. They direct the search process to examine unvisited regions and to generate solutions that differ in significant ways from those seen before.

5.1.2 The TS Algorithm Selection

The literature review (presented in Chapter 2) shows that there have been at least two previous attempts in cross-dock warehouse domain that have used TS-based algorithm to solve the VRP problems: Lee et al. (2006) and Wen et al. (2009). It is also found that TS algorithm is widely used to solve scheduling problems (Widmer and Hertz, 1989; Taillard, 1990; Ben-Daya and Al-Fawzan, 1998) and assignment problems (i.e. Lim et al., 2006; Miao et al., 2009). Lim et al. (2006) and Miao et al. (2009) also show that their TS-based approach outperforms GA and CPLEX Solver and provides better results than GA and CPLEX Solver.

In addition, Fan and Machemehl (2008) use TS-based algorithm for transit route network design problem, showing that TS outperforms GA and provides better results.

Considering that VRSP is the integration of VRP and scheduling problems, in this research a tabu-search based algorithm is developed to solve large-size VRSP.

TS also provides an easier representation for the VRSP problem in comparison to other meta-heuristics such as GA or ACO. For example, consider a cross-dock

warehouse serving four customers. The cross-dock warehouse can be denoted as 0 and the four customers as 1, 2, 3, and 4. So an outbound truck route starting from cross-dock warehouse to customer 1, then going to customers 2, 3, 4 and returning back to cross-dock warehouse can be represented as 0-1-2-3-4-0.

TS based method begins with a potential solution, and generates other candidate solutions by neighborhood generation mechanism. The neighborhood generation mechanisms used in TS algorithm are: adjacent pairwise interchange, all pairwise interchange, last insertion, and all insertion (Baker and Trietsch, 2009). TS has a very simple mechanism among meta-heuristics methods to generate neighborhoods as the candidate solutions.

In addition, TS also has a simple mechanism of neighborhood selection so as to determine the best neighborhood which becomes the seed for the next iteration. A detailed explanation of TS mechanism for generating neighborhoods and neighborhood selection is explained in section 5.2.

The algorithm proposed in this chapter is titled “Tabu Search-based Algorithm for Vehicle Routing and Scheduling Problem” (TSA-VRSP). Since VRSP is integration of two NP hard problems, it is a challenging to solve VRSP and requires the use of non-traditional optimization methodology, presented next.

5.2 DEVELOPMENT OF TABU-SEARCH BASED ALGORITHM FOR VRSP

This section describes the development of TSA-VRSP which uses only short-term memory as it is simpler than using both short and long term memories, and adequate enough to get good results. Thus, at the starting stage of TSA-VRSP development, the key features that are applied in TSA-VRSP are neighborhood generation (swap pair formulation) and tabu list. Glover (1989) and Ben-Daya and Al-Fawzan (1998) call this version of TS as “simple Tabu Search”.

In its simplest form, the TS algorithm has the following features: initial sequence, mechanism for generating some neighborhood of the current sequence, tabu list, and stopping criteria.

In applying TS-based algorithm, there are some tactical decisions and parameters of TS-based algorithm that need to be determined, since all of these influence the outcome of the algorithm. TS tactical decisions that need to be determined are (Baker and Trietsch, 2009),:

- the method of obtaining the initial solution (initial seed) at the initial phase,
- neighborhood generation strategy (mechanism),
- neighborhood selection criteria (selection of the new seed), and
- stopping criteria.

As per Fan and Machemehl (2008), TS parameters that need to be determined are: number of neighborhoods, number of iterations (generations), and tabu list

size. These parameters are determined using sensitivity analysis (Fan and Machemehl, 2008). Parameter selection of TSA-VRSP is explained in the following section.

5.2.1 Initial Phase

TSA-VRSP uses greedy algorithm to generate an initial outbound truck route. Greedy algorithm can be used to obtain an initial solution (Baker and Trietsch, 2009), which can then be followed by other heuristics to get optimal (near optimal) solution.

A greedy algorithm builds a schedule from scratch, by adding jobs to the schedule one at a time. It does not necessarily add the jobs in order from earliest to latest. For example, a greedy algorithm could choose the first-to-last job in sequence or the last –to-first structure in sequence (Baker and Trietsch, 2009).

In TSA-VRSP, the greedy algorithm is applied to generate the initial solution (initial route). The greedy algorithm is applied to determine the first-to-last node in outbound truck route based on the travel time between two nodes. A detailed explanation of greedy algorithm's application in TSA-VRSP follows.

The search starts by adding the first node to the first route by looking for the nearest node to the cross-dock warehouse. The second route is then determined by selecting the node (customer) that has the shortest time from the first route. The third route is also selected by selecting the node (customer) that has the shortest time from the second route. The route selection follows this way until

the last route is determined. An illustration of the greedy algorithm at the initial phase of TSA-VRSP follows.

Table 5.1 Example of travel time of 4 customers (in minutes)

	CD	1	2	3	4
CD	0	375	330	120	386
1	375	0	190	271	46
2	330	190	0	280	157
3	120	271	280	0	292
4	386	46	157	292	0

Note: *CD=cross-dock warehouse

Table 5.1 presents an example travel time data between nodes of 4 customers.

Using greedy algorithm, an initial route can be developed as follows:

- First step: select the customer with the shortest time from cross-dock.
This is customer 3.
- Second step: select the customer with the shortest time from customer 3.
This is customer 1.
- Third step: select the customer with the shortest time from customer 1.
This is customer 4.
- Fourth step: Select the last customer. The only remaining unselected customer at this stage is customer 2. Add that to the route.
- Fifth step: To complete the route generation, add the return of truck from the last customer back to the cross-dock.

So the initial route obtained using greedy algorithm is CD-3-1-4-2-CD: from cross-dock warehouse truck goes to customer 3, 1, 4, and 2, before returning back to cross-dock warehouse), with a total travel time of 924 minutes.

5.2.2 Neighborhood Generation Mechanism

In TS there are four mechanisms to generate neighborhoods: adjacent pairwise interchange, last insertion, all pairwise interchange, and all insertion (Baker and Trietsch, 2009). Since adjacent pairwise interchange (API) mechanism is simple in its application, compared to others, and provides good results, it is used in TSA-VRSP to generate the neighborhoods.

In TSA-VRSP, the API mechanism is used to generate n number of candidates from the current seed. In this mechanism, the number of neighborhoods generated is $(n-1)$, where n is the number of nodes. For example if there are 10 customers, by using API, at every iteration (of neighborhood generation), 9 neighborhoods are generated.

An illustration of API mechanism in TSA-VRSP is presented next. For example if there are 4 nodes and the current sequence (the current seed) is 1- 2- 3- 4, the neighborhood of this current sequence is:

2-1-3-4

1-3-2-4

1-2-4-3

Neighborhood 1 (swap pair = 1, 2)

Neighborhood 2 (swap pair = 2, 3)

Neighborhood 3
(swap pair = 3, 4)

⋮

Neighborhood n-3
(swap pair = n-3, n-2)

Neighborhood n-2
(swap pair = n-2, n-1)

Neighborhood n-1
(swap pair = n-1, n)

Figure 5.2 Adjacent pairwise interchange mechanism

These three sequences are called the neighborhood of the seed. The size of neighborhood is $(n-1)$ (or 3 in this case). The basic idea behind API is demonstrated in Figure 5.2.

Referring to Figure 5.2, for the seed 1, 2, 3, ..., n, the neighborhoods generated by API mechanism are

2,1,3,...,n-2,n-1,n;

1,3,2,...,n-2,n-1,n;

⋮

1,2,3,...,n-1,n-2,n;

1,2,3,...,n-2,n,n-1.

These sequences are called the neighborhood of the seed and the number of the sequences or size of neighborhood is $(n-1)$. In TS, the swap pair (1, 2) is the same with swap pair (2, 1), and swap pair (2, 4) is the same with swap pair (4, 2), and so on.

5.2.3 Neighborhood Selection

In TS method, the selected neighborhood becomes the seed for the following iteration. Thus, the neighborhood selection also implies the next seed selection. The selected neighborhood is a neighborhood that satisfies the following two conditions:

- 1) The neighborhood (candidate) which has the best objective function value or the neighborhood (candidate) which has the first improvement (the user selects one of these two criteria), and
- 2) The selected neighborhood must satisfy the tabu free status. The tabu value of the swap pair of the selected neighborhood must be zero (i.e. not in tabu status).

This is the fundamental difference between TS and other meta-heuristics approaches: in TS, the selected neighborhood (candidate) must satisfy the criterion that the swap pair of the selected neighborhood not be in tabu status (tabu value of zero). Since in TS algorithm, the swap pair (a, b) is the same as swap pair (b, a) , thus the value of swap pair (a, b) is the same as the value of swap pair (b, a) .

In TSA-VRSP, the neighborhood selection mechanism selects the candidate (neighborhood) with the best objective function value which also satisfies the tabu free status (tabu value of zero). Since TSA-VRSP is an algorithm developed to solve the VRSP, whose objective is to minimize the total cost, in TSA-VRSP, the objective function value chosen is the total cost and the best objective function is the one with the lowest total cost. Thus TSA-VRSP always selects the neighborhood with the lowest total cost and zero tabu value.

An illustration of the neighborhood selection mechanism of TSA-VRSP can be found in Appendix-B.

5.2.4 Stopping Criteria

The TSA-VRSP is stopped when one of the following two criteria has been met:

1. TSA-VRSP has been run for m iterations (max iterations set by the user),
or
2. For n consecutive iterations TSA-VRSP leads to the same result. Value
of n is also set by the user.

Once one of these two conditions is satisfied, TSA-VRSP stops. The rationale for setting the values of the two parameters, m and n , is provided in section 5.5.

5.3 THE FRAMEWORK OF TSA-VRSP

This section describes the TSA-VRSP framework. Since the TSA-VRSP is developed to solve the VRSP problem, the objective of TSA-VRSP is to minimize the (sum of) earliness, tardiness, inventory holding and transportation costs.

TSA-VRSP works similar to backward scheduling. In the initial phase, it starts with application of the greedy algorithm which determines the initial routing of outbound trucks. Based on the initial routing, the transportation cost and distance to customer are calculated. After that, scheduling of outbound and inbound trucks is performed. The outbound trucks' sequence and their assignment to the outbound dock doors is determined, followed by a computation of the outbound truck's departure time. That is followed by a determination of the outbound

truck's arrival time at the customer. Using the outbound truck's departure time, TSA-VRSP can determine the inbound truck release time and the sequence at each inbound dock door. With this info the arrival time of inbound truck to the cross-dock yard can be determined. Finally, the earliness, tardiness, and inventory costs are determined, leading to the calculation of the total cost, which is the sum of transportation, earliness, tardiness, and inventory costs.

After the initial phase, TSA-VRSP applies TS algorithm for generating a number of candidate solutions. By applying TS algorithm, TSA-VRSP generates several neighborhoods from the current solution (current seed), subjecting each neighborhood to the same process described above for the initial routing. Adjacent pairwise interchange is used to generate the neighborhoods from the current seed. The detailed mechanism of adjacent pairwise interchange is described in Appendix-B. For each neighborhood the transportation cost and distance to customer is calculated, followed by determination of the outbound and inbound truck scheduling, ending with the computation of earliness cost, tardiness cost, inventory handling cost, and total cost.

Based on the neighborhoods' total cost, the best one can be selected. The neighborhood selection mechanism is also explained in Appendix-B. The selected neighborhood becomes the seed for the next iteration. This procedure is repeated until one of the stopping criteria is met. The selected neighbourhood at the time the algorithm stops is the final solution. The workflow and the framework of TSA-VRSP is presented in Figures 5.3 and 5.4.

As shown in Figure 5.3, the initial solution is generated once, at the start of the application of TSA-VRSP, shown as a solid line in Figure 5.3. Thereafter, the following step generates new neighborhoods based on the best previous neighborhood, followed by selection of the next best neighborhood. These steps, shown as dashed line in Figure 5.3, are performed iteratively until one of two stopping criteria is met:

- (1) the number of iterations exceeds the maximum number of iterations, n , or
- (2) over the preceding m iterations, the solution does not change.

The parameters n and m numbers are set by the user.

Figure 5.3 TSA-VRSP workflow

Step 1. Initial Phase

- 1.1 Set up the default tabu list. All the pair values are zero.
- 1.2 Using greedy algorithm generate the initial outbound truck route; this is the initial seed (initial route).
- 1.3 Calculate the transportation cost.
- 1.4 Calculate the travel time to customer based on the outbound truck route.
- 1.5 Determine the outbound truck sequence.
- 1.6 Determine the outbound truck departure time.
- 1.7 Determine the arrival time to customer.
- 1.8 Determine the inbound truck release time.
- 1.9 Determine the inbound truck sequence.
- 1.10 Determine the inbound truck arrival time.
- 1.11 Calculate the earliness and tardiness costs.
- 1.12 Calculate the inventory holding cost.
- 1.13 Calculate the total cost.

Step 2 Apply TS-algorithm to obtain the candidate solutions

- 2.1 Based on the current route (current seed) generate n neighborhoods as n new routes with adjacent pairwise interchange mechanism.
- 2.2 Perform steps 1.3 to 1.13 for each new route.
- 2.3 Determine 'top k candidates' (k candidates with lowest total cost).
- 2.4 Select 1st best candidate (candidate with the lowest total cost).
- 2.5 Check its tabu list status.

Figure 5.4 The framework of TSA-VRSP

TSA-VRSP uses TS to generate several different combinations (solutions). Using some iterations, it can get near optimal solution for VRSP. Firstly, TSA-VRSP initiates routing using greedy algorithm to obtain the initial solution (initial stage $t = 0$). At step 0 (step initialization), the result from greedy algorithm is set as $Solution_Sofar^0$ and $Solution_Best^0$. Then the stage number is updated, $t = t + 1 = 1$.

At $t = 1$ Neighborhoods^t from $Solution_Sofar^0$ ($Solution_Sofar^0$ is the solution obtained from greedy algorithm) are generated. Next the best neighborhood is selected by checking its tabu list to ensure that the selected neighborhood is free

and not forbidden (its tabu value is zero). After checking the tabulist status, the $\text{Best_Neighborhoods}^t$ can be selected. The $\text{Best_Neighborhoods}^t$ is set as Solution_Sofar^1 .

At the next step, the tabu list status needs to be updated; this is done by updating the tabu value of the selected neighborhood with the maximum value of tabu list. After that the Solution_Best^1 is determined by comparing Solution_Best^0 with Solution_Sofar^1 . If the total cost of Solution_Sofar^1 is lower than Solution_Best^0 , Solution_Sofar^1 becomes Solution_Best^1 , else Solution_Best^0 becomes Solution_Best^1 . After each iteration, a check is performed to ensure the maximum number of iterations (M , set by the user) has not been exceeded. If t has not yet exceeded M , the process continues.

If t has not yet exceeded M , the stage number t is updated, $t = t+1$ and the process continued. The above steps are repeated iteratively until one of two stopping criteria is met. When the process terminates, the Solution_Best^t is the solution for VRSP.

5.4 TSA-VRSP PROCEDURE

The VRSP addresses integrated vehicle routing and scheduling problem in a cross-dock warehouse. Since the problem is NP hard and needs to be solved in a reasonable time, TSA-VRSP is proposed to solve the problem. Previous experiments with VRSP using CPLEX Solver (section 3.4) show that the VRSP problem can be solved in a reasonable time for only small size problems.

TSA-VRSP is a TS heuristics-based algorithm developed to solve the VRSP. TSA-VRSP works in a backward manner, starting from customer side to outbound side of cross-dock, then moving on to inbound side of cross-dock and ending at the supplier side. The general framework of TSA-VRSP has already been discussed in the previous section (section 5.3). This section will explain the detailed procedure of TSA-VRSP, in solving the VRSP.

The notation used in TSA-VRSP procedure is as follows.

- M = a very large number
- Z = outbound truck routing
- XY = outbound truck sequence and assignment at outbound dock door
- OTS = a set of decision variables for outbound truck schedule, consisting of outbound truck departure time (DT), outbound truck arrival time to customer location (S), earliness (E), and tardiness (L)
- XY and OTS form the outbound truck schedule
- ITS = determine inbound truck schedule, consisting of sequence and assignment of inbound truck to inbound dock door (x and y), inbound truck release time (RT), and inbound truck arrival time (AT)
- TC = transportation cost
- $ECLC$ = earliness and tardiness cost (EC is a function of Earliness and LC is a function of tardiness)

- IC = inventory holding cost (IC is a function of DT and RT)
- TTC = total cost ($TTC = TC + ECLC + IC$)
- t1 = iteration number for decision variable Z
- t2 = iteration number for decision variable OTS ($t2 = 1, 2, \dots, T2$).
- T2 = maximum number of iterations for decision variable OTS
- n1 = number of neighborhoods ($n1=1,2,\dots,N1$)
- N1 = maximum number of neighborhoods generated per iteration

TSA-VRSP starts by determining the outbound truck routing (Z), then determines the outbound truck schedule consisting of outbound truck sequence (X) and assignment (Y) at outbound dock door, outbound truck departure time (DT), outbound truck arrival time to the customer location (S), earliness (E), and tardiness (L).

After that TSA-VRSP goes on to determine the inbound truck schedule consisting of inbound truck sequence (x) and assignment (y) at inbound dock door, inbound truck release time (RT) and arrival time (AT). Based on these decision variables, the total cost can be determined, by adding up the transportation, earliness, tardiness, and inventory holding costs.

The objective of TSA-VRSP is to minimize the total cost consisting of transportation, earliness, tardiness, and inventory costs. TSA-VRSP has a set of decision variables consisting of {Z, X, Y, DT, S, E, L, x, y, RT, and AT} and is developed in two parts: first part is the main program, named TSA-VRSP main

program (Figure 5.5). The second part is a function called TTC calculation routine (Figure 5.6).

TTC calculation routine's only function is to calculate total cost (TTC). The output of the routine is the outbound truck and inbound truck schedules, along with its TTC. In summary, TTC routine determines:

- Outbound truck schedule consisting of outbound trucks' departure times, outbound trucks' arrival time at customer location, and associated earliness and tardiness.
- Earliness and tardiness costs.
- Inbound truck schedule consisting of sequence and assignment of inbound trucks to inbound dock doors, inbound truck release times, and inbound truck arrival times.
- Inventory holding cost, which is a function of departure times and release times.
- Total cost, which is the sum of transportation cost, earliness and tardiness cost, and inventory cost.

Figure 5.5 TSA-VRSP main program

Figure 5.6 TTC calculation routine

5.5 EXPERIMENTS WITH TSA-VRSP

Multiple experiments are performed to test and validate the TSA-VRSP's effectiveness in solving the VRSP. The experiments are performed for different

number of vehicles, orders, and dock doors. The algorithm is coded using MATLAB version 7.9.0.

It is difficult to benchmark the results of TSA-VRSP since a similar model integrating vehicle routing and scheduling problems into one model is not available. Instead a comparison with two previously developed methods (VRSP and VRSP-CZHTW using CPLEX Solver) is done.

5.5.1 TS Parameter Setting

Before any numerical experiments can be conducted with TSA-VRSP, appropriate parameter settings need to be determined. The parameters are: number of iterations and size of tabu list. In this research recommendation of Miao et al. (2009) is followed, which sets the number of iterations at 10^6 and tabu list size at 10.

5.5.2 Experiment Result of TSA-VRSP

The datasets used in the experiments was discussed in Chapter 3 and in Appendix-A. The algorithm is coded in MATLAB version 7.9.0 and running on Windows computer platform Intel® Celeron® CPU N2840 2.16 GHz. The results of experiments are presented in Table 5.2-5.4. The experiments with TSA-VRSP show that TSA-VRSP is able to solve the small scale problem in around 20-60 seconds, while for medium scale problem, TSA-VRSP takes around

hundreds seconds to one hour. For large scale problem, TSA-VRSP takes around two hours. Based on these experiment results TSA-VRSP is able to solve from small up to large scale problems in an acceptable time.

Table 5.2 TSA-VRSP on small dataset

Problem set	TSA-VRSP	
	Total cost (\$)	Run Time (sec)
SS_1	376.8	25.2
SS_2	579.6	17.4
SS_3	661.8	30.7
SS_4	958.2	26.3
SS_5	636	49.4
SS_6	1009.8	49.6
SS_7	865.76	61.1
SS_8	1425	56.9

Table 5.3 TSA-VRSP on medium dataset

Problem set	TSA-VRSP	
	Total cost (\$)	Run Time (sec)
MS_1	1225.8	384.3
MS_2	2107.2	325.5
MS_3	314.4	826.7
MS_4	555	857.8
MS_5	5139.6	1999
MS_6	9514.2	1896.3
MS_7	5523.9	3656.4
MS_8	3248.4	5390.1

Table 5.4 TSA-VRSP on large dataset

Problem set	TSA-VRSP	
	Total cost (\$)	Run Time (sec)
LS_1	9943.8	5614.8
LS_2	17280	5134.1
LS_3	19503.9	6877.2
LS_4	30918.9	6859.5
LS_5	46378.4	7118.9
LS_6	48697.3	7124.3
LS_7	72291.3	8273.9
LS_8	73268.1	8389.1

5.6 COMPARISON OF TSA-VRSP WITH VRSP AND VRSP-CZHTW

In order to validate the results of TSA-VRSP, it is next compared with two previously developed models of VRSP: VRSP (Chapter 3) and VRSP-CZHTW (Chapter 4). The comparison focuses on the solution quality and the run time of the algorithms. Tables 5.5-5.7 present the experiment results of the comparison. In addition to the cost and run time statistics for the three algorithms, the tables also present the difference in cost between VRSP, VRSP-CZHTW, and TSA-VRSP expressed as a % Gap. Gap 1 shows the percentage gap between VRSP-CZHTW (or TSA-VRSP) with VRSP, while gap 2 shows the percentage gap between VRSP-CZHTW and TSA-VRSP.

Table 5.5 Comparison of VRSP, VRSP-CZHTW, and TSA-VRSP
for small datasets

Problem set		SS_1	SS_2	SS_3	SS_4
VRSP	Total cost (\$)	376.8	579.6	661.8	958.2
	Run time (sec)	0.77	0.28	2.11	1.39
VRSP-CZHTW	Total cost (\$)	376.8	579.6	661.8	958.2
	Run time (sec)	0.02	0.02	0.3	0.38
	Gap 1	0.0%	0.0%	0.0%	0.0%
TSA-VRSP	Total cost (\$)	376.8	579.6	661.8	958.2
	Run time (sec)	25	17	30	26
	Gap 1	0.0%	0.0%	0.0%	0.0%
	Gap 2	0.0%	0.0%	0.0%	0.0%
Problem set		SS_5	SS_6	SS_7	SS_8
VRSP	Total cost (\$)	636	808.8	728.8	1402.8
	Run time (sec)	0.51	3.28	1.95	2.67
VRSP-CZHTW	Total cost (\$)	636	983.56	829.4	1402.8
	Run time (sec)	0.42	1.72	1.41	0.47
	Gap 1	0.0%	17.8%	12.1%	0.0%
TSA-VRSP	Total cost (\$)	636	1009.8	865.76	1425
	Run time (sec)	49	49	61	56
	Gap 1	0.0%	24.9%	18.8%	1.6%
	Gap 2	0.0%	2.7%	4.4%	1.6%

Table 5.6 Comparison of VRSP, VRSP-CZHTW, and TSA-VRSP
for medium datasets

Problem set		MS_1	MS_2	MS_3	MS_4
VRSP	Total cost (\$)	1054.8	1693.2	795.26	571.2
	Run time (sec)	1800	2057	>3600	>3600
VRSP-CZHTW	Total cost (\$)	1456.5	2048.16	882.32	1158.86
	Run time (sec)	3.69	26.56	676.8	691.3
	Gap 1	38.08%	20.96%	10.95%	102.88%
TSA-VRSP	Total cost (\$)	1225.8	2107.2	314.4	555
	Run time (sec)	384.3	325.5	826.7	857.8
	Gap 1	16.21%	24.45%	-60.47%	-2.84%
	Gap 2	-15.84%	2.88%	-64.37%	-52.11%
Problem set		MS_5	MS_6	MS_7	MS_8
VRSP	Total cost (\$)	4275.45	6890.432	No solution	No solution
	Run time (sec)	>3600	>3600		
VRSP-CZHTW	Total cost (\$)	4750.5	7489.6	7657.2	12591.6
	Run time (sec)	782.1	863.3	837.8	1824.8
	Gap 1	11.11%	8.70%	-	-
TSA-VRSP	Total cost (\$)	5139.6	9514.2	5523.9	3248.4
	Run time (sec)	1999	1896.3	3656.4	5390.1
	Gap 1	20.21%	38.08%	-	-
	Gap 2	8.19%	27.03%	-27.86%	-74.20%

Table 5.7 Comparison of VRSP, VRSP-CZHTW, and TSA-VRSP
for large datasets

Problem set		LS_1	LS_2	LS_3	LS_4
VRSP	Total cost (\$)	No solution	No solution	No solution	No solution
	Run time (sec)				
VRSP-CZHTW	Total cost (\$)	16869.6	15598.4	24184.8	38648.7
	Run time (sec)	>7200	>7200	>7200	>7200
	Gap 1	-	-	-	-
TSA-VRSP	Total cost (\$)	9943.8	17280	19503.9	30918.9
	Run time (sec)	5614.8	5134.1	6877.2	6859.5
	Gap 1	-	-	-	-
	Gap 2	-41.05%	10.78%	-19.35%	-20.00%
Problem set		LS_5	LS_6	LS_7	LS_8
VRSP	Total cost (\$)	No solution	No solution	No solution	No solution
	Run time (sec)				
VRSP-CZHTW	Total cost (\$)	No solution	No solution	No solution	No solution
	Run time (sec)				
	Gap 1	-	-	-	-
TSA-VRSP	Total cost (\$)	46378.4	48697.3	72291.3	73268.1
	Run time (sec)	7118.9	7124.3	8273.9	8389.1
	Gap 1	-	-	-	-
	Gap 2	-	-	-	-

Results in Tables 5.5-5.7 show that TSA-VRSP can solve up to large-scale problems quite fast (in a matters of minutes) and comes up with a good solution even for large datasets, comprising of hundreds of customer orders. VRSP model using CPLEX Solver, on the other hand, can only be solved for small size problems. For even medium size problems the time taken becomes longer and VRSP is unable to solve the problem for large data sets. For large datasets, even VRSP-CZHTW's performance, which is quite good for medium size datasets,

begins to suffer in comparison to TSA-VRSP, with no solution starting with five hundred customers of the large datasets.

So, the experiments conducted with various size datasets show that, in term of computation time and quality of solution, TSA-VRSP's performance is much better in comparison to VRSP and VRSP-CZHTW.

Thus it can be concluded that TSA-VRSP is a powerful tool for solving the integrated problem of scheduling and routing of vehicles at a cross-dock warehouse. The experiment results show that TSA-VRSP has much better performance than two previously developed integrated models (VRSP and VRSP-CZHTW) and has acceptable computation times for even large scale problems.

5.7 SUMMARY OF TSA-VRSP

In this chapter a new TS heuristic-based approach is proposed to solve the vehicle routing and scheduling problem, termed TSA-VRSP. The TS algorithm used in TSA-VRSP is simpler since only two of its short term memory attributes are used, namely (1) generating neighborhoods and (2) tabu list. However, TSA-VRSP with these two attributes is a good starting point for implementing TS in solving VRSP and obtaining a near optimal solution quickly. As shown in Table 5.2 to 5.4, TSA-VRSP can solve the VRSP for even large-scale problems very quickly, and leads to a good solution. In addition, a comparison with two previously developed methods, VRSP and VRSP-CZHTW solved with CPLEX

Solver, shows that the TSA-VRSP performs the best. In short, TSA-VRSP is the best decision support tool for cross-dock managers in managing the routing and scheduling of vehicles on a daily basis.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This chapter presents conclusions and summary of the research presented in the first five chapters of the thesis. Key contributions of this research are also presented. Finally, future directions this research could take are also presented to conclude the report.

6.1 CONCLUSIONS

One of the contemporary strategies to improve the efficiency of distribution is cross-dock warehousing. Cross-docking has been known as the logistics strategy in which products from several suppliers/sources can be consolidated into a single shipment, enabling shipments in FTL rather than LTL. The cross-dock warehouse ships the received goods directly to the customers with little or no storage in between thereby also reducing inventory holding cost.

Due to shorter product life cycles, many companies prefer to use cross-docking, rather than traditional warehousing, to distribute their products. The advantages of implementing cross-docking are well-known: it saves distribution time, shortens the delivery cycle time, improves customer satisfaction, and cost savings. Cross-docking can help to reduce warehouse space cost, inventory cost, transportation cost, and labour cost. These benefits have resulted in more and more companies switching from traditional to cross-dock warehousing.

Since a cross-dock facility handles many products and trucks in a day, an optimal schedule of unloading, loading and routing is a prerequisite to efficiently and effectively operate the cross-dock facility. A common practice is to optimize the truck schedules and the vehicle routes separately. One common way to schedule trucks in the cross-dock is to use FCFS rule, which has some clear disadvantages, such as large number of early and tardy jobs and long waiting time. The TMS software, on the other hand, is used for effective vehicle planning, scheduling and planning of delivery routes.

Good information flow between suppliers, cross dock facility and customers is very important to enhance the supply chain performance. IT support system is the second biggest barrier faced by companies implementing cross-docking, as found by Saddle Creek Corp. surveys in US in 2008 and 2011. Effective cross-docking depends heavily on the use of information technology tools and sophisticated planning software that coordinate the physical product and information flows. In addition, sophisticated and real time scheduling is important since it sequences delivery vehicles and orders in an efficient way on a day-to-day basis, smoothening the cross docking operations and reducing cost. Motivated by these observations, this research integrates supplier, cross dock facility, and customer information flows in one system and develops scheduling and vehicle routing policies at the cross dock distribution system in an integrated manner. This is done by developing a comprehensive optimization model for distribution of products through the cross-dock facility. The model considers the customer delivery time windows, product consolidation decisions, and the

routing of deliveries so that the products are received by customers on time, at minimum total cost.

This thesis focuses on the operations within a cross-dock facility and develops three models, in increasing complexity, to address the research problem:

- (1) Develop a model for vehicle routing and scheduling problem at cross-dock;
- (2) Develop a model for vehicle routing and scheduling problem at cross-dock with customer zones and hard time windows; and
- (3) Propose TS-based meta-heuristics algorithm to solve large scale vehicle routing and scheduling problem at cross-dock within acceptable time.

This thesis results in four key contributions. First key contribution of this research (presented in Chapter 2) is to establish a clear picture of the state-of-the-art of research into cross-dock operations. A comprehensive literature review in cross-dock research area was performed and critical research gaps identified. The key gap addressed in this research is the lack of integrated planning of vehicle scheduling and routing problem at cross docks resulting in sub-optimal performance. This research focuses on the delivery vehicle scheduling and routing at cross-docking facility in an integrated manner, also taking into account the customer delivery time windows and product consolidation decisions.

While developing the integrated scheduling and routing policies, this research takes into account both the customer and the cross dock manager's points of view. The model is developed to maximize customer satisfaction and minimize the total operation cost of the distribution process at cross-dock. These objectives

are represented as minimizing the total cost, consisting of penalty cost of earliness and tardiness (customer's focus) and inventory holding cost and transportation cost (cross-dock manager's focus).

Second key contribution of this research is the development of integrated vehicle scheduling and routing policies for the cross-dock facility. Towards this end, firstly a model for vehicle routing and scheduling at cross-dock, termed VRSP, and modelled as an MILP is presented in Chapter 3. The model considers the suppliers, inbound trucks, cross-dock, outbound trucks, and customers in one system and determines the vehicle routing and vehicle scheduling policies simultaneously. The objective of VRSP is to determine the schedule of inbound and outbound vehicles at the cross-dock and the routing of outbound vehicles to deliver products within customer specified time windows. The delivery strategy proposed by the model consists of (a) inbound truck arrival time and release time and its schedule at inbound dock door; (b) product consolidation; and (c) outbound truck route, its departure time, and its schedule at the outbound dock doors.

The objective of VRSP is to minimize total cost, consisting of four components: the earliness and tardiness penalty cost of the orders, the inventory holding cost of the products, and the transportation cost of deliveries. A total of 32 constraints are modeled in VRSP, and are divided into four sets. First set of constraints is related to inbound truck scheduling in inbound area of the cross-dock; the second set to outbound truck routing in delivering orders to customer locations; the third set to product consolidation, and the fourth set is related to outbound truck scheduling in outbound area of the cross-dock.

The performance of VRSP is tested using CPLEX Solver version 12.3, running on an Intel® Celeron® CPU N2840 2.16 GHz computer. The experiment results show that the model can be solved in a fair amount of time only for small size problems.

Most cross-dock warehouses have dozens of inbound and outbound dock-doors and handle hundreds of orders a day. As the experimental analysis of the VRSP shows that the model is not able to solve real-life size problems, an alternative solution is proposed (Chapter 4) to handle the real-life complexity inherent in cross-dock warehouse operations. In this work, a VRSP simplification strategy is proposed to facilitate the development of operations policies for real-life sized cross-dock warehouses. This is the contribution discussed in Chapter 4, which proposes a modified VRSP model incorporating customer zones and hard-time windows to eliminate late and early delivery costs and reduce the size of the solution space. The model is denoted as VRSP-CZHTW. Through experiments using CPLEX Solver version 12.3, running on Intel® Celeron® CPU N2840 2.16 GHz computer, it is shown that the modified model (VRSP-CZHTW) can solve medium up to large scale problems within an acceptable time.

While the simplification strategy allows for handling the cross-dock vehicle scheduling and routing problem in reasonable time even for large size problems, it is based on a simplification strategy that might not be applicable to all real-life systems. For example, in this research, the original model (VRSP) uses soft time windows and does not enforce customer zones.

There may be instances where VRSP-CZHTW cannot be solved, for example when the delivery time windows are tight, and no feasible solution can be found for delivery within a zone. Another limitation of VRSP and VRSP-CZHTW approaches is that all available outbound trucks must be used for delivery of customer orders, thus determining the optimal number of trucks is a separate problem that must be addressed.

As a final step towards solving a life-size VRSP (Chapter 5), the original model (VRSP) is solved without enforcing the constraints of customer zones and hard-time windows. However, in order to solve this NP-hard problem, a meta-heuristic approach is investigated that may be able to solve the model in reasonable time and provide close-to-optimal solutions. Based on further literature review, an alternative approach that can be employed to solve an NP hard problem effectively is tabu-search based algorithms.

As a final contribution of this research, a meta-heuristics method based on TS algorithm is developed (Chapter 5) to solve the VRSP. The algorithm is denoted as TSA-VRSP. The TS algorithm used in TSA-VRSP is simpler since it only uses short term memory attribute of TS, which are: (1) neighbourhood generation, and (2) tabu list. However, TSA-VRSP with these two attributes can be a good starting point to solve VRSP and obtain a near-optimal solution in a short time. Experiments are performed to validate and test the TSA-VRSP, for different number of vehicles, orders, and dock doors, on an Intel® Celeron® CPU N2840 2.16 GHz computer. The algorithm is coded using MATLAB version 7.9.0.

Based on the TSA-VRSP results of experiments, it is seen that VRSP can be solved in an acceptable time even for large scale problems. The experimental

results also show that TSA-VRSP has better performance compared to the two previously discussed methods (VRSP and VRSP-CZHTW using CPLEX Solver). Thus TSA-VRSP has the potential to be the foundation of a decision tool for cross-dock managers that can develop integrated vehicle routing and scheduling policies for effective cross-dock operations.

As the final conclusion, based on the experiment results of VRSP, VRSP-CZHTW, and TSA-VRSP, it can be drawn conclusion that if the decision maker faces a small scale problem of cross-dock warehouse, it is recommended that he uses VRSP solved with CPLEX Solver approach. While if the decision maker faces medium problem size of cross-dock warehouse, he can use whether VRSP-CZHTW solved with CPLEX Solver or TSA-VRSP approaches. Both approaches are a good decision tools for medium scale size. On the other hand, if the decision maker faces a large scale problem of cross-dock warehouse, it is recommended that the decision maker uses TSA-VRSP than the other two methods, since based in the experiment result for large scale problem, it is shown that TSA-VRSP outperformed the two other methods.

6.2 FUTURE WORK

This research has made a bold attempt to develop solution techniques for VRSP that can be the foundation of decision tool for cross-dock managers. The work can, however, be further extended as follows:

1. The TS algorithm can be further modified to increase its effectiveness. The current research only used short term memory of TS. In future, the long term

memory aspects of TS may be considered to improve the performance of the proposed algorithm. In addition, application of Adaptive Memory Procedure (AMP) in the algorithm may also be considered.

2. Besides applying long term memory to improve the TSA-VRSP, there are many other improvements that can also be made, such as applying another mechanism for neighborhood generation, another feature of TS to enrich the framework of TS, using aspiration criteria, and flexible tabu value.
3. Other meta-heuristics, including hybrid meta-heuristics, may be considered to solve the VRSP and a comparison done with TS-based approach.
4. Some exact methods may also can be investigated to solve the VRSP and compared against heuristics-based solutions.
5. Another challenging future work, is to develop a TS-based approach to solve the VRSP-CZHTW, thus developing TS-based algorithm to solve the vehicle routing and scheduling problem of cross-dock with hard time window and customer zones, and comparing it with the VRSP-CZHTW solved by CPLEX Solver and TSA-VRSP.
6. Other features of VRSP-CZHTW that can be investigated are: degree of overlap of time windows and the effect to the quality of solution, and the effect of the number of customer zones on the quality of solution and the computation time.

REFERENCES

Acar, K., Yalcin, A., and Yankov, D., 2012. Robust door assignment in less-than-truckload terminals. *Computers & Industrial Engineering*. Vol 63, pp. 729-738.

Adamuthe, A.C. and Bichkar, R.S., 2012. Tabu search for solving personnel scheduling problem. *International Conference on Communication, Information & Computing Technology (ICCICT)*. October, 19-20, Mumbai, India.

Al-Khayyal, F. and Hwang, S.J., 2007. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: Applications and model. *European Journal of Operational Research*. Vol. 176, pp. 106-130.

Alpan, G., Ladier, A.L., Larbi, R., and Penz, B., 2011. Heuristic solutions for transshipment problems in a multiple door cross docking warehouse. *Computers & Industrial Engineering*. Vol.61, pp. 402-408.

Alvarez-Pérez, G.A., González-Velarde, J.L., and Fowler, J.W., 2009. Crossdocking-Just in Time scheduling: an alternative solution approach, *Journal of the Operational Research Society*. Vol. 60, pp. 554-564.

Androutsopoulos, K.N. and Zografos, K.G., 2009. Solving the multi-criteria time-dependent routing and scheduling problem in a multimodal fixed scheduled network. *European Journal of Operational Research*. Vol. 192, pp. 18-28.

Apte, U.M., and Viswanathan, S., 2000. Effective cross docking for improving distribution efficiencies. *International Journal of Logistics Research and Applications*. Vol. 3, No. 3, pp. 291-302.

Arabani, A.B., Zandieh, M., and Ghomi, S.M.T.F., 2012. A cross-docking scheduling problem with sub-population multi-objective algorithms. *International Journal of Advanced Manufacturing Technology*. Vol. 58, pp. 741-761.

Arabani, A.B., Zandieh, M., and Ghomi, S.M.T.F., 2011. Multi-objective genetic-based algorithms for a cross-docking scheduling problem. *Applied Soft Computing*. Vol.11, pp. 4954-4970.

Arabani, A.R.B., Ghomi, S.M.T.F., and Zandieh, M., 2010. A multi-criteria cross-docking scheduling with just-in-time approach. *International Journal of Advanced Manufacturing Technology*. Vol. 49, pp. 741-756.

- Arabani, A.R.B., Ghomi, S.M.T. F., and Zandieh, M., 2011. Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert Systems with Applications*. Vol. 38, pp. 1964-1979.
- Bachlaus, M., Pandey, M.K., Mahajan, C., Shankar, R., and Tiwari, M.K., 2008. Designing an integrated multi-echelon agile supply chain network: a hybrid taguchi-particle swarm optimization approach. *Journal of Intelligent Manufacturing*. Vol. 19, No. 6, pp. 747-761.
- Baker, K.R. and Trietsch, D., 2009. Principles of sequencing and scheduling. *John Wiley & Sons, Inc.* Hoboken, New Jersey.
- Barnes, J.W. and Carlton, W.B., 1995. Solving the vehicle routing problem with time windows using reactive tabu search. *INFORMS Conference*. New Orleans.
- Bartholdi, J.J., III and Gue, K.R., 2000. Reducing labor costs in an LTL crossdocking terminal. *Operations Research*. Vol. 48, No. 6, pp. 823-832.
- Bartholdi, J.J., III and Gue, K.R., 2004. The best shape for crossdock. *Transportation Science*. Vol. 38, No. 2, pp. 235-244.
- Belle, J.V., Valckenaers, P., Berghe, G.V., and Cattrysse, D., 2013. A tabu search approach to the truck scheduling problem with multiple docks and time windows. *Computers & Industrial Engineering*. Vol. 66, pp. 818-826.
- Ben-Daya, M and Al-Fawzan, M., 1998. A tabu search approach for the flow shop scheduling problem. *European Journal of Operational Research*. Vol. 109, pp. 88-95.
- Bolduc, M.C., Laporte, G., Renaud, J., and Boctor, F.F., 2010. A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars. *European Journal of Operational Research*. Vol. 202, pp. 122-130.
- Boysen, N., 2010. Truck scheduling at zero-inventory crossdocking terminals. *Computers and Operations Research*. Vol. 37, pp. 32-41.
- Boysen N. and Fliedner, M., 2010. Cross-dock scheduling: Classification, literature review and research agenda. *Omega*. Vol. 38, pp. 413-422.
- Boysen, N., Fliedner, M., and Scholl, A., 2010. Scheduling inbound and outbound trucks at cross docking terminals. *OR Spectrum*. Vol. 32, pp. 135-161.
- Bozer, Y.A., and Carlo, H.J., 2008. Optimizing inbound and outbound door assignments in less-than-truckload crossdocks. *IIE Transactions*. Vol. 40, pp. 1007-1018.

- Brandao, J., 2009. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*. Vol. 195, pp. 716-728.
- Carlton, W.B., 1995. A tabu search approach to the general vehicle routing problem. *Ph.D. dissertation*. The University of Texas at Austin.
- Chen, F. and Lee, C.Y., 2009. Minimizing the make-span in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*. Vol.193, pp. 59-72.
- Chen, F. and Song, K., 2009. Minimizing make-span in two-stage hybrid crossdocking scheduling problem. *Computers and Operations Research*. Vol. 36, pp. 2066-2073.
- Chen, J. and Chen, S., 2008. Optimization of vehicle routing problem with load balancing and time windows in distribution. *4th International Conference on Wireless Communications, Networking and Mobile Computing*. 12-14 October, Dalian, China, pp. 1-4.
- Chen, J.C., Hsieh, W. H., Cheng, C. H., and Chen, C. S., 2009. Hybrid meta-heuristics for vehicle routing problem with time window constraints. *6th International Conference on Service Systems and Service Management (ICSSSM)*. 8-10 June, Xiamen, China. pp. 766-771.
- Chen, L., Bostel, N., Dejax, P., Cai, J., Xi, L., 2007. A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *European Journal of Operational Research*. Vol. 181, Issue 1, pp. 40-58.
- Chen, P., Guo, Y., Lim, A., and Rodrigues, B., 2006. Multiple crossdocks with inventory and time windows. *Computers and Operations Research*. Vol. 33, pp. 43-63.
- Chen, R., Fan, B., and Tang, G., 2009. Scheduling problems in cross docking. *Lecture Notes in Computer Science*. Vol. 5573, pp. 421-429.
- Cheng, C.B. and Wang, K.P., 2009. Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. *Expert Systems with Applications*. Vol. 36, Issue 4, pp. 7758-7763.
- Chiang, W.C., Russell, R., Xu, X., and Zepeda, D., 2009. A simulation/metaheuristic approach to newspaper production and distribution supply chain problems. *International Journal of Production Economics*. Vol. 121, pp. 752-767.
- Choy, K.L., Chow, H.K.H., Poon, T.C., and Ho, G.T.S., 2012. Cross-dock job assignment problem in space-constrained industrial logistics distribution hubs

with a single docking zone. *International Journal of Production Research*. Vol. 50, No. 9, pp. 2439-2450.

Christiansen, M. and Fagerholt, K., 2002. Robust ship scheduling with multiple time windows. *Naval Research Logistics*. Vol. 49, pp. 611-625.

Chunyu, R., 2007. Multi-constraint vehicle routing problem with time windows of optimization for electronic commerce. *International Conference on Transportation Engineering (ICTE)*. 22-24 July, Southwest Jiaotong University, Chengdu, China. pp. 1647-1653.

Cordeau, J.F. and Laporte, G., 2001. A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR*. Vol. 39, No. 3, pp. 292-298.

Cordeau, J.F. and Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B*. Vol. 37, pp. 579-594.

Cordeau, J.F. and Maischberger, M., 2012. A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*. Vol. 39, pp. 2033-2050.

Cordeau, J.F., Gendreau, M., and Laporte, G., 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*. Vol. 30, pp. 105-119. John Wiley & Sons, Inc.

Cordeau, J.F., Laporte, G., and Mercier, A., 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*. Vol. 52, pp. 928-936.

Crainic, T.G., Gajpal, Y., Gendreau, M., 2012. Multi-zone multi-trip vehicle routing problem with time windows. *CIRRELT-2012-36*.

De Freitas, J.A., 2012. A tabu search metaheuristic algorithm for the multiple depot vehicle routing problem with time windows. *Bachelors of Industrial and Systems Engineering in the Faculty of Engineering, Built Environment and Information Technology*. University of Pretoria.

Derigs, U. and Reuter, K., 2009. A simple and efficient tabu search heuristic for solving the open vehicle routing problem. *Journal of the Operational Research Society*. Vol. 60, pp. 1658-1669.

Donaldson, H., Johnson, E.L., Ratliff, H.D., and Zhang, M., 1998. Scheduled-driven cross-docking networks. *Georgia Institute of Technology Research Report*. <http://www.isye.gatech.edu/research/files/misc9904.pdf>

- Dondo, R. and Cerdá, J., 2013. A sweep-heuristic based formulation for the vehicle routing problem with cross-docking. *Computers and Chemical Engineering*. Vol. 48, pp. 293-311.
- Dondo, R., Méndez, C.A., and Cerdá, J., 2011. The multi-echelon vehicle routing problem with cross docking in supply chain management. *Computers and Chemical Engineering*. Vol.35, pp. 3002-3024.
- Eswaramurthy, V.P. and Tamilarasi, A., 2007. Tabu search strategies for solving job shop scheduling problems. *Journal of Advanced Manufacturing Systems*. Vol. 6, No.1, pp. 59-75.
- Fagerholt, K. and Christiansen, M., 2000. A combined ship scheduling and allocation problem. *Journal of the Operational Research Society*. Vol. 51, pp. 834-842.
- Fagerholt, K. and Christiansen, M., 2000. A travelling salesman problem with allocation, time window and precedence constraints an application to ship scheduling. *International Transactions in Operational Research*. Vol. 7, pp. 231-244.
- Fakhrzad, M.B. and Esfahani, A.S., 2014. Modeling the time windows vehicle routing problem in cross-docking strategy using two meta-heuristic algorithms. *International Journal of Engineering Transactions A: Basics*. Vol. 27, No. 7, pp. 1113-1126.
- Fan, J., 2009. Dynamic vehicle scheduling problem with changeable time window. *Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. 14-16 August, Tianjin, China, pp. 276-280.
- Fan, W. and Machemehl, R.B., 2008. Tabu search based heuristic method for the transit route network design problem. *Computer-aided Systems in Public Transport (Lecture Notes in Economics and Mathematical Systems, editor: Mark Hickman, Pitu Mirchandani, Stefan Voß)*. pp387-408.
- Forouharfard, S. and Zandieh, M., 2010. An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems. *International Journal of Advance Manufacturing Technology*. Vol. 51, pp. 1179-1193.
- Fu, Z., Eglese, R., and Li, L., 2008. A unified tabu search algorithm for vehicle routing problems with soft time windows. *Journal of the Operational Research Society*. Vol. 59, pp. 663-673
- Glover, F., 1989. Tabu Search-Part I. *ORSA Journal on Computing*. Vol. 1, No.3, pp.190-206.

- Glover, F., 1990. Tabu Search-Part II. *ORSA Journal on Computing*. Vol. 2, No.1, pp. 4-32.
- Glover, F., 1990. Tabu Search: A Tutorial. *Interfaces*. Vol. 20, No. 4, pp. 74-94.
- Glover F., Taillard E., Laguna M., de Werra D., 1993. Tabu Search. *Annals of Operations Research*. Vol. 41.
- Gue, K.R., 1999. The effects of trailer scheduling on the layout of freight terminals. *Transportation Science*. Vol. 33, No. 4, pp. 419-428.
- Gümüş, M., and Bookbinder, J.H., 2004. Cross-docking and its implications in location-distribution system. *Journal of Business Logistics*. Vol. 25, No. 2, pp. 199-228.
- Han,Y., Guan, X., and Shi, L., 2007. Optimal supply location selection and routing for emergency material delivery. *Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering*. September 22-25, Scottsdale, AZ, USA, pp. 1039-1044.
- Harks, T., König, F.G., and Matuschke, J., 2013. Approximation algorithms for capacitated location routing. *Transportation Science*. Vol. 47, No. 1, pp. 3-22.
- Hauser, K. and Chung, C.H., 2006. Genetic algorithms for layout optimization in crossdocking operations of a manufacturing plant. *International Journal of Production Research*. Vol. 44, No. 21, pp. 4663-4680.
- Heragu, S.S., Du, L., Mantel, R.J., and Schuur, P.C., 2005. Mathematical model for warehouse design and product allocation. *International Journal of Production Research*. Vol. 43, No. 2, pp. 327-338.
- Ho, S.C. and Haugland, D., 2004. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*. Vol. 31, pp. 1947-1964.
- Hooker, J.N. and Natraj, N.R., 1995. Solving a general routing and scheduling problem by chain decomposition and tabu search. *Transportation Science*. Vol. 29, No. 1, pp. 30-44.
- Hosseini, S.D., Shirazi, M.A., and Karimi, B., 2014. Cross-docking and milk run logistics in a consolidation network: A hybrid of harmony search and simulated annealing approach. *Journal of Manufacturing Systems*. Vol. 33, pp. 567-577.
- Hu, Z.H., Zhao, Y., and Choi, T.M., 2013. Vehicle routing problem for fashion supply chains with cross-docking. *Mathematical Problems in Engineering*. Article ID 362980.

http://www.mmh.com/article/jcpenney_out_the_door_in_five_minutes.

Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., and Yagiura, M., 2005. Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation science*. Vol. 39, No. 2, pp. 206-232.

Isfahani, M.M., Moghaddam, R.T., and Naderi, B., 2014. Multiple cross-docks scheduling using two meta-heuristic algorithms. *Computers & Industrial Engineering*. Vol. 74, pp. 129-138.

Javanmard, S., Vahdani, B., and Moghaddam, R.T., 2014. Solving a multi-product distribution planning problem in cross docking networks: An imperialist competitive algorithm. *International Journal of Advanced Manufacturing Technology*. Vol. 70, pp. 1709-1720.

Jayaraman, V. and Ross, A., 2003. A simulated annealing methodology to distribution network design and management. *European Journal of Operational Research*. Vol. 144, pp. 629-645.

Jia, Y.J., Wang, C.J., Liu, Y., and Wang, X.F., 2009. Dynamic model and scheduling system for a real-time pickup and delivery problem with time windows. *Proceedings of the IEEE International Conference on Automation and Logistics*. Shenyang, China, pp. 2082-2086.

Jinhui, G., Ying, X., and Hongzhen, W., 2012. An improved TS for the open vehicle routing problem with soft time windows. *2012 Fifth International Symposium on Computational Intelligence and Design*. pp. 382-385.

Karimi, I.A., Sharafali, M., and Mahalingam, H., 2005. Scheduling tank container movements for chemical logistics. *American Institute of Chemical Engineers AIChE Journal*. Vol. 51, No. 1, pp. 178-197.

Ko, C.S., Lee, H.K., Choi, E.J., and Kim, T. 2008. A genetic algorithm approach to dock door assignment in automated cross-docking terminal with restricted layout. *International Conference on Genetic and Evolutionary Methods*. pp. 186-195.

Konur, D. and Golias, M.M., 2013. Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty. *Computers & Industrial Engineering*. Vol. 65, pp. 663-672.

Konur, D. and Golias, M.M., 2013. Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach. *Transportation Research Part E*. Vol. 49, pp. 71-91.

- Kreng, V.B. and Chen, F., 2008. The benefits of a cross-docking delivery strategy: a supply chain collaboration approach. *Production Planning & Control*. Vol. 19, No.3, pp. 229-241.
- Ladier, A.L. and Alpan, G., 2014. Crossdock truck scheduling with time windows: earliness, tardiness and storage policies. *Journal of Intelligent Manufacturing*. DOI 10.1007/s10845-014-1014-4.
- Larbi, R., Alpan, G., and Penz, B., 2009. Scheduling transshipment operations in a multiple inbound and outbound door crossdock. *International Conference on Computers and Industrial Engineering*. pp. 227-232.
- Larbi, R., Alpan, G., Baptiste, P., and Penz, B., 2007. Scheduling of transshipment operations in a single strip and stack doors crossdock. *19th International Conference on Production Research ICPR*.
- Larbi, R., Alpan, G., Baptiste, P., and Penz, B., 2011. Scheduling cross docking operations under full, partial and no information on inbound arrivals. *Computers & Operations Research*. Vol. 38, pp. 889-900.
- Lau, H.C. and Song, Y., 2002. Combining two heuristics to solve a supply chain optimization problem. *Proceeding of 15th European Conference on Artificial Intelligence*. 21-26 July, Lyon France, pp. 581-585.
- Lee, K., Kim, B.S., and Joo, C.M., 2012. Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance. *Expert Systems with Applications*. Vol. 39, pp. 12975-12983.
- Lee, Y. H., Jung, J. W., and Lee, K. M., 2006. Vehicle routing scheduling for cross-docking in the supply chain. *Computers & Industrial Engineering*. Vol. 51, pp. 247-256.
- Ley, S. and Elfayoumy, S., 2007. Cross dock scheduling using genetic algorithms. *The 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*. June 20-23, Jacksonville, FL, USA.
- Li, Y., Lim, A., and Rodrigues, B., 2004. Crossdocking-JIT scheduling with time windows. *Journal of the Operational Research Society*. Vol. 55, pp. 1342-1351.
- Li, Z., Low, M.Y.H., and Lim, R.Y.G., 2009. Optimal decision-making on product allocation for crossdocking and warehousing operations. *International Journal of Services Operations and Informatics*. Vol. 4, No. 4, pp. 352-365.
- Li, Z., Sim, C.H., Low, M.Y.H., and Lim, Y.G., 2008. Optimal product allocation for crossdocking and warehousing operations in FMCG supply chain. *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. pp. 2963-2968.

- Liao, C.J., Lin, Y., and Shih, S.C., 2010. Vehicle routing with cross-docking in the supply chain. *Expert Systems with Applications*. Vol. 37, pp. 6868-6873.
- Liao, T.W., Chang, P.C., Kuo, R.J., and Liao, C.-J., 2014. A comparison of five hybrid metaheuristic algorithms for unrelated parallel-machine scheduling and inbound trucks sequencing in multi-door cross docking systems. *Applied Soft Computing*. Vol. 21, pp. 180-193.
- Liao, T.W., Egbelu, P.J., and Chang, P.C., 2013. Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations. *International Journal of Production Economics*. Vol. 141, pp. 212-229.
- Liao, T.W., Egbelu, P.J., Chang, P.C., 2012. Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations. *Applied Soft Computing*. Vol. 12, pp. 3683-3697.
- Liaw, C.F., 1999. A tabu search algorithm for the open shop scheduling problem. *Computers & Operations Research*. Vol. 26, Issue 2, pp. 109-126.
- Liaw, C.F., 2003. An efficient tabu search approach for the two-machine preemptive open shop scheduling problem. *Computers & Operations Research*. Vol. 30, Issue 14, pp. 2081-2095.
- Lim, A., Ma, H., and Miao, Z., 2006. Truck dock assignment problem with time windows and capacity constraint in transshipment network through crossdocks. *Lecture Notes in Computer Science*. Vol. 3982 LNCS, pp. 688-697.
- Lim, A., Miao, Z., Rodrigues, B., and Xu, Z., 2005. Transshipment through crossdocks with inventory and time windows. *Naval Research Logistics*. Vol. 52, pp. 724-733.
- Lim, A., Rodrigues, B., Xiao, F., and Zhu, Y., 2002. Crane scheduling using tabu search. *Proceedings of the International Conference on Tools with Artificial Intelligence*. pp. 146-153.
- Liu, X. and Sun, Y., 2010. A modeling of cross dock location and its application to CD between parts suppliers and vehicle manufacturers. *ICLEM 2010: Logistics for Sustained Economic Development*. pp. 3414-3419.
- Louis, Y., Tsui, L.Y., and Chang, C.H., 1990. A microcomputer based decision support tool for assigning dock doors in freight yards. *Proceedings of the 12th Annual Conference on Computers & Industrial Engineering*. Vol. 19, No. 1-4, pp. 309-312.
- Madani, N., 2013. Application of tabu search to scheduling trucks in multiple doors cross-docking systems. *Master Thesis*. École Polytechnique de Montreal, Université de Montreal.

- Maknoon, M.Y. and Baptiste, P., 2009. Cross-docking: increasing platform efficiency by sequencing incoming and outgoing semi-trailers. *International Journal of Logistics Research and Applications*. Vol. 12, No. 4, pp. 249-261.
- Maknoon, M.Y., Kone, O., and Baptiste, P., 2014. A sequential priority-based heuristic for scheduling material handling in a satellite cross-dock. *Computers & Industrial Engineering*. Vol. 72, pp. 43-49.
- McWilliams, D.L., 2009. A dynamic load-balancing scheme for the parcel hub-scheduling problem. *Computers & Industrial Engineering*. Vol. 57, pp. 958-962.
- McWilliams, D.L., Stanfield, P.M., and Geiger, C.D., 2005. The parcel hub scheduling problem: a simulation-based solution approach. *Computers & Industrial Engineering*. Vol. 49, pp. 393-412.
- Mehrizi, H.A., Berman, O., and Baharnemati, M.R., 2014. Designing production-inventory-transportation systems with capacitated cross-docks. *Transportation Science*. Vol. 48, No. 1, pp. 121-135.
- Meng, Q., Lee, D.H., and Cheu, R.L., 2005. Multi objective vehicle routing and scheduling problem with time window constraints in hazardous material transportation. *Journal of Transportation Engineering*. Vol. 699, pp. 699-707.
- Meng Y. and Tang L., 2010. A tabu search heuristic to solve the scheduling problem for a batch-processing machine with non-identical job sizes. *2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM 2010)*. Vol. 3, pp. 1703-1707.
- Miao, Z., Fu, K., Fei, Q., and Wang, F., 2008. Meta-heuristic algorithm for the transshipment problem with fixed transportation schedules. *Lecture Notes in Computer Science*. Vol. 5027 LNAI, pp. 601-610.
- Miao, Z., Lim, A., and Ma, H., 2009. Truck dock assignment problem with operational time constraint within crossdocks. *European Journal of Operation Research*. Vol. 192, pp. 105-115.
- Moghadam, S.S., Ghomi, S.M.T.F, and Karimi, B., 2014. Vehicle routing scheduling problem with cross docking and split deliveries. *Computers and Chemical Engineering*. Vol. 69, pp. 98-107.
- Mooney, E.L., and Rardin, R.L., 1993. Tabu search for a class of scheduling problems. *Annals of Operations Research*. Vol. 41, pp. 253-278.
- Morais, V.W.C., Mateus, G.R., and Noronha, T.F., 2014. Iterated local search heuristics for the vehicle routing problem with cross-docking. *Expert Systems with Applications*. Vol. 41, pp. 7495-7506.

- Mousavi, S.M. and Moghaddam, R.T., 2013. A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain. *Journal of Manufacturing Systems*. Vol. 32, pp. 335-347.
- Mousavi, S.M., Moghaddam, R.T., and Jolai, F., 2013. A possibilistic programming approach for the location problem of multiple cross-docks and vehicle routing scheduling under uncertainty. *Engineering Optimization*. Vol. 45, No. 10, pp. 1223-1249.
- Mousavi, S.M., Moghaddam, R.T., and Siadat, A., 2014. Optimal design of the cross-docking in distribution networks: heuristic solution approach. *IJE TRANSACTIONS A: Basics*. Vol. 27, No. 4, pp. 533-544.
- Mousavi, S.M., Moghaddam, R.T., Siadat, A., and Vahdani, B., 2013. A hybrid simulated annealing algorithm for location of cross-docking centers in a supply chain. *Lecture Notes in Computer Science*. LNCS Vol. 7919, pp. 12-21.
- Mousavi, S.M., Vahdani, B., Moghaddam, R.T., and Hashemi, H., 2014. Location of cross-docking centers and vehicle routing scheduling under uncertainty: A fuzzy possibilistic stochastic programming model. *Applied Mathematical Modelling*. Vol. 38, pp. 2249-2264.
- Mula, J., Peidro, D., Madronero, M.D., and Vicens, E., 2009. Mathematical programming models for supply chain production and transport planning. *European Journal of Operation Research*. Vol. 204, No. 3, pp. 377-390.
- Musa, R., Arnaout, J.P., and Jung, H., 2010. Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Computers & Industrial Engineering*. Vol. 59, pp. 85-92.
- Naso, D., Surico, M., and Turchiano, B., 2007. Reactive scheduling of a distributed network for the supply of perishable products. *IEEE Transaction on Automation Science and Engineering*. Vol. 4, No. 4, pp. 407-423.
- Nassief, W., Contreras, I., and As'ad, R., 2015. A mixed-integer programming formulation and Lagrangean relaxation for the cross-dock door assignment problem. *International Journal of Production Research*. DOI:10.1080/00207543.2014.1003664.
- Oh, Y., Hwang, H., Cha, C.N., and Lee, S., 2006. A dock-door assignment problem for the Korean mail distribution center. *Computers & Industrial Engineering*. Vol. 51, pp. 288-296.
- Pankratz, G., 2005. Dynamic vehicle routing by means of a genetic algorithm. *International Journal of Physical Distribution & Logistics Management*. Vol. 35, No. 5, pp. 362-383.

- Petersen, H.L. and Ropke, S., 2011. The pickup and delivery problem with cross-docking opportunity. *ICCL 2011, LNCS*. Vol. 6971, pp. 101-113.
- Potvin, J.Y., Kervahut, T., Garcia, B.L., 1996. The vehicle routing problem with time windows-Part I: Tabu Search. *INFORMS Journal of computing*. Vol. 9, pp. 158-164.
- Ponnambalam, S.G., Sankar, G.S. and Karthikeyan, R., 2002. Scheduling flexible manufacturing system using tabu search method. *IEEE International Conference on Industrial Technology (IEEE ICIT'02)*. Vol. 2, pp. 1043-1048, Bangkok, Thailand.
- Ratliff, H.D., Vate, J.V., and Zhang, M., 1998. Network design for load-driven cross-docking systems. *Georgia Institute of Technology Research Report*. <http://www.isye.gatech.edu/research/files/misc9914.pdf>
- Rohrer, M., 1995. Simulation and cross docking. *Proceedings of the 1995 Winter Simulation Conference*. pp. 846-849.
- Ross, A. and Jayaraman, V., 2008. An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design. *Computers & Industrial Engineering*. Vol. 55, pp. 64-79.
- Saddle Creek Corp., 2008. Cross-Docking Trends Report. *Saddle Creek Corporation White Papers Series*. pp. 1-24.
- Saddle Creek Corp., 2011. Cross-Docking Trends Report. *Saddle Creek Corporation White Papers Series*. pp. 1-12.
- Santos, F.A., Mateus, G.R., and Cunha, A.S., 2013. The pickup and delivery problem with cross-docking. *Computers & Operations Research*. Vol. 40, pp. 1085-1093.
- Santos, F.A., Mateus, G.R., Cunha, A.S., 2011. A branch-and-price algorithm for a vehicle routing problem with cross-docking. *VI LAGOS - Latin American Algorithms, Graphs and Optimization Symposium*. Vol. 37, pp. 249-254.
- Santos, F.A., Mateus, G.R., Cunha, A.S., 2011. A novel column generation algorithm for the vehicle routing problem with cross-docking. *INOC 2011, LNCS*. Vol. 6701, pp. 412-425.
- Schaffer, B., 1998. Cross docking can increase efficiency. *Automatic I.D. News*. Vol. 14, Issue 8.
- Schulze, J. and Fahle, T., 1999. A parallel algorithm for the vehicle routing problem with time window constraints. *Annals of Operations Research*. Vol. 86, pp. 585-607.

- Semet, F. and Taillard, É. D., 1993. Solving real-life vehicle routing problems efficiently using taboo search. *Annals of Operations Research*. Vol. 41, pp. 469-488.
- Schwind, G.F., 1996. A systems approach to docks and cross docking. *Material Handling Engineering*. Vol. 51, Issue 2, pp. 59-62.
- Shakeri, M., Low, M. Y. H., Li, Z., and Lee, E. W., 2010. Two efficient constructive heuristics for scheduling trucks at crossdocking terminals. *IEEE International Conference on Service Operations and Logistics and Informatics*. QingDao, China, pp. 177-182.
- Shakeri, M., Yoke, M., Hean Low, M.Y., and Li, Z., 2008. A generic model for crossdock truck scheduling and truck-to-door assignment problems. *IEEE International Conference on Industrial Informatics*. July 13-16, pp. 857-864.
- Shi, W., Liu, Z., Shang, J., and Cui, Y., 2013. Multi-criteria robust design of a JIT-based cross-docking distribution center for an auto parts supply chain. *European Journal of Operational Research*. Vol. 229, pp. 695-706.
- Shiguemoto, A.L., Netto, U.S.C., and Sergio, G.H., 2014. An efficient hybrid meta-heuristic for a cross-docking system with temporary storage. *International Journal of Production Research*. Vol. 52, No. 4, pp. 1231-1239.
- Solimanpur, M. and Elmi, A., 2013. A tabu search approach for cell scheduling problem with makespan criterion. *International Journal of Production Economics*. Vol. 141, Issue 2, pp. 639-645.
- Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*. Vol. 35, No. 2, pp. 254-264.
- Solomon, M.M. and Desrosiers, J., 1988. Time window constrained routing and scheduling problems. *Transportation Science*. Vol. 22, No. 1, pp. 1-13.
- Song, K. and Chen, F., 2007. Scheduling cross docking logistics optimization problem with multiple inbound vehicles and one outbound vehicle. *Proceeding of the IEEE International Conference and Automation and Logistics*. August 18-21, Jinan, China, pp. 3089-3094.
- Sung, C.S. and Song, S.H., 2003. Integrated service network design for a cross-docking supply chain network. *The Journal of the Operational Research Society*. Vol. 54, No. 12, pp. 1283-1295.
- Sung, C.S. and Yang, W., 2008. An exact algorithm for a cross-docking supply chain network design problem. *Journal of the Operational Research Society*. Vol. 59, pp. 119-136.

- Syarif, A., Yun, Y.S, and Gen, M., 2002. Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach. *Computers & Industrial Engineering*. Vol. 43, pp. 299-314.
- Taillard, E., 1990. Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*. Vol. 47, pp. 65-74.
- Taillard, E., Badeau, P., Gendreau, M., Guertin, F., and Potvin J.Y., 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*. Vol. 31, No. 2, pp. 170-186.
- Tang, S.L. and Yan, H., 2010. Pre-distribution vs. post-distribution for cross-docking with transshipments. *Omega*. Vol. 38, pp. 192-202.
- Tan, K.C., Chew, Y.H., and Lee, L.H., 2006. A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*. Vol. 172, pp. 855-885.
- Tarantilis, C.D., 2013. Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking. *Optimization Letters*. Vol. 7, pp. 1583-1596.
- Tootkaleh, S.R., Shirazi, M.A., Ghomi, S.M.T.F., and Hosseini, S.D., 2014. Truck capacity analysis in a cross-dock transportation network considering direct shipment. *Journal of Advanced Transportation*. Vol. 48, pp. 891-901.
- Tsui, L.Y. and Chang C-H., 1990. A Microcomputer based decision support tool for assigning dock doors in freight yards. *Computer & Industrial Engineering*. Vol. 19, Nos. 1-4, pp. 309-312.
- Tsui, L.Y. and Chang C-H., 1992. An optimal solution to dock door assignment problem. *Computer & Industrial Engineering*. Vol. 23, Nos. 1-4, pp. 283-286.
- Vahdani, B. and Zandieh, M., 2010. Scheduling trucks in cross-docking systems: Robust meta-heuristics. *Computers & Industrial Engineering*. Vol. 58, pp. 12-24.
- Vahdani, B.,Moghaddam, R.T., Zandieh, M., and Razmi, J., 2012. Vehicle routing scheduling using an enhanced hybrid optimization Approach. *Journal of Intelligent Manufacturing*. Vol. 23, pp. 759-774.
- Vahdani, B.,Soltani, R., and Zandieh, M., 2010. Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics. *International Journal of Advance Manufacturing Technology*. Vol. 46, pp. 769-783.
- Vanhoucke, M., 2002. Optimal due date assignment in project scheduling. *Vlerick working papers*. Vlerick Leuven Gent Management School. Vol. 19, pp. 1-24.

- Vis, I.F.A. and Roodbergen, K.J. 2008. Positioning of goods in a cross-docking environment. *Computers & Industrial Engineering*. Vol. 54, pp. 677-689.
- Vis, I.F.A. and Roodbergen, K.J., 2011. Layout and control policies for cross docking operations. *Computers & Industrial Engineering*. Vol. 61, pp. 911-919.
- Wei, J., Leung, S.C.H., Lee, S.C., and Kwok, B., 2009. Simulation of RFID-enabled loading strategy for outbound logistics: a case study in Hong Kong. *International Conference on Computers & Industrial Engineering (CIE 2009)*. pp. 1769-1774.
- Wen, M., Larsen, J., Clausen, J., Cordeau, J.F., and Laporte, G., 2009. Vehicle routing with cross-docking. *Journal of the Operational Research Society*. Vol. 60, pp. 1708-1718.
- Widmer, M and Hertz, A., 1989. A new heuristic method for the flow shop sequencing problem. *European Journal of Operational Research*. Vol. 41, pp. 186-193.
- Witt, C.E., 1998. Crossdocking: Concepts demand choice. *Material Handling Engineering*. Vol. 53, Issue 7.
- Wong, C.S., 2001. Tabu search-based techniques for clustering data set. *Thesis of Master of Philosophy*. The University of Hong Kong.
- Wu, Y., Dong, M., and Yang, D., 2011. Cross-docking centre operation optimization using simulation-based genetic algorithm. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*.
- Yan, X., Guozhong, M., Qiuhong, J., Jian, L. and Yun, W., 2008. Research on the optimal scheduling model of logistics distribution vehicles. *The 8th International Conference of Chinese Logistics and Transportation Professionals (ICCLTP)*. 8-10 October, Chengdu, China, pp. 2401-2406.
- Yanchang, L. and Min, D., 2009. Discussion of cross-docking design in modern logistics systems. *WRI World Congress on Computer Science and Information engineering*. Vol. 3, pp. 149-151.
- Yao, Q.M., Xin, M.L., Le, Z., and Yu, X.H., 2008. A new tabu search heuristic algorithm for the vehicle routing problem with time windows. *15th International Conference on Management Science & Engineering*. September 10-12, Long Beach, USA, pp. 1648-1653.
- Yu, W. and Egbelu, P.J., 2008. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research*. Vol. 184, pp. 377-396.

Zarandi, M. H. F., Khorshidian, H., and Shirazi, M.A., 2014. A constraint programming model for the scheduling of JIT cross-docking systems with pre-emption. *Journal of Intelligent Manufacturing*. Vol. 01/2014.

Zhang, Q. and Zhang, Q., 2008. An improved ant colony algorithm for the logistics vehicle scheduling problem. *Proceeding of Second International Symposium on Intelligent Information Technology Application*. 20-22 December, Shanghai, China, pp. 55-59.

Zografos, K.G. and Androutsopoulos, K.N., 2004. A heuristic algorithm for solving hazardous materials distribution problems. *European Journal of Operational Research*. Vol. 152, pp. 507-519.

Appendix-A

Data Sets for Small Size Problems (SS_1 to SS_8)

1. SS_1 (2/2/2/1/1)

Data for VRSP and TSA-VRSP:

Data for travel time:

	CD	1	2
CD	0	211	116
1	211	0	149
2	116	149	0

Data for Delivery time window:

a_i	b_i
733	1205
556	1033

Data for u_{il} and demand quantities:

u_{il}	d_i
1	20
2	18

Data for VRSP-CZHT (only one customer zone):

CZ1:

Data for travel time CZ1:

	CD	1	2
CD	0	211	116
1	211	0	149
2	116	149	0

Data for delivery time window CZ1:

a_i	b_i
733	1205
556	1033

Data for demand and U_{il} CZ1:

u_{il}	d_i
1	20
2	18

2. SS_2 (2/2/2/2/2)

Data for VRSP and TSA-VRSP:

Data for travel time:

	CD	1	2
CD	0	211	116
1	211	0	149
2	116	149	0

Data for delivery time window:

a_i	b_i
733	1205
556	1033

Data for U_{il} and demand:

u_{il}	d_i
1	20
2	18

Data for VRSP-CZHTW (there are 2 customer zones):

Data for CZ1:

Data for travel time

	CD	1
CD	0	211
1	211	0

Data for U_{il} , demand, and time window

u_{il}	d_i	a_i	b_i
1	20	733	1205

Data for CZ2:

Data for travel time:

	CD	2
CD	0	116
2	116	0

Data for U_{il} , demand, and time window

u_{il}	d_i	a_i	b_i
2	18	556	1033

3. SS_3 (3/2/2/1/1)

Data for VRSP and TSA VRSP:

Data for travel time:

	CD	1	2	3
CD	0	357	302	237
1	357	0	285	131
2	302	285	0	194
3	237	131	194	0

Data for u_{il} and demand:

u_{il}	d_i
1	11
2	20
1	6

Data for time window:

a_i	b_i
725	1234
752	1150
518	989

Data set for CZHTW (only one customer zone)

Data for CZ1:

Data for travel time:

	CD	1	2	3
CD	0	357	302	237
1	357	0	285	131
2	302	285	0	194
3	237	131	194	0

Data for u_{il} and demand:

u_{il}	d_i
1	11
2	20
1	6

Data for time window:

a_i	b_i
725	1234
752	1150
518	989

4. SS_4 (3/2/2/2/2)

Data for VRSP and TSA-VRSP:

Data for travel time:

	CD	1	2	3
CD	0	357	302	237
1	357	0	285	131
2	302	285	0	194
3	237	131	194	0

Data for u_{il} and demand:

u_{il}	d_i
1	21
2	30
1	16

Data for time window:

a_i	b_i
725	1234
752	1150
518	989

Data set for CZHTW (there are two customer zones):

CZ1:

Data for travel time

	CD	1	3
CD	0	357	237
1	357	0	131
3	237	131	0

Data for u_{il} , demand, and time window

u_{il}	d_i	a_i	b_i
1	21	725	1234
1	16	518	989

CZ2:

Data for travel time:

	CD	2
CD	0	302
2	302	0

Data for u_{il} , demand, and time window

u_{il}	d_i	a_i	b_i
2	30	752	1150

5. SS_5 (4/2/2/1/1)

Data for VRSP and TSA-VRSP:

Data for travel time:

	CD	1	2	3	4
CD	0	375	330	120	386
1	375	0	190	271	46
2	330	190	0	280	157
3	120	271	280	0	292
4	386	46	157	292	0

Data for u_{il} and demand

u_{il}	d_i
1	5
2	12
1	7
2	10

Data for time window

a_i	693	707	563	684
b_i	1008	995	1110	1248

Data for CZHTW (only one customer zone):

CZ1:

Data for travel time:

	CD	1	2	3	4
CD	0	375	330	120	386
1	375	0	190	271	46
2	330	190	0	280	157
3	120	271	280	0	292
4	386	46	157	292	0

Data for u_{il} and demand

u_{il}	d_i
1	5
2	12
1	7
2	10

Data for time window

a_i	693	707	563	684
b_i	1008	995	1110	1248

6. SS_6 (4/2/2/2/2)

Data for VRSP and TSA-VRSP:

Data for travel time:

	CD	1	2	3	4
CD	0	375	330	120	386
1	375	0	190	271	46
2	330	190	0	280	157
3	120	271	280	0	292
4	386	46	157	292	0

Data for u_{il} and demand:

u_{il}	d_i
1	5
2	20
1	17
2	8

Data for time window

a_i	693	707	563	684
b_i	808	995	1110	848

Data for CZHTW (there are two customer zones)

CZ1:

Data for travel time

	CD	1	2	4
CD	0	375	330	386
1	375	0	190	46
2	330	190	0	157
4	386	46	157	0

Data for u_{il} , demand, and time window

u_{il}	d_i	a_i	b_i
1	5	693	808
2	20	707	995
2	8	684	848

CZ2:

Data for travel time:

	CD	3
CD	0	120
3	120	0

Data for u_{il} , demand, and time window

u_{il}	d_i	a_i	b_i
1	17	563	1110

7. SS_7 (5/2/2/1/1):

Data for VRSP and TSA-VRSP:

Data for travel time:

	CD	1	2	3	4	5
CD	0	334	334	441	225	196
1	334	0	144	108	128	199
2	334	144	0	187	121	282
3	441	108	187	0	226	301
4	225	128	121	226	0	169
5	196	199	282	301	169	0

Data for time window

a_i	b_i
693	1008
707	1400
563	1110
684	1248
677	1062

Data for u_{il} and demand

u_{il}	d_i
2	8
2	5
1	2
2	7
1	8

Data for CZHTW (only one customer zone):

CZ1:

Data for travel time:

	CD	1	2	3	4	5
CD	0	334	334	441	225	196
1	334	0	144	108	128	199
2	334	144	0	187	121	282
3	441	108	187	0	226	301
4	225	128	121	226	0	169
5	196	199	282	301	169	0

Data for time window

a_i	b_i
693	1008
707	1400
563	1110
684	1248
677	1062

Data for u_{il} and demand

u_{il}	d_i
2	8
2	5
1	2
2	7
1	8

8. SS_8 (5/2/2/2/2):

Data for VRSP and TSA-VRSP:

Data for travel time:

	CD	1	2	3	4	5
CD	0	334	334	441	225	196
1	334	0	144	108	128	199
2	334	144	0	187	121	282
3	441	108	187	0	226	301
4	225	128	121	226	0	169
5	196	199	282	301	169	0

Data for time window

a_i	b_i
693	1008
707	1400
563	1110
684	1248
677	1062

Data for u_{il} and time window

u_{il}	d_i
2	10
2	5
1	20
2	17
1	8

Data for CZHTW:

CZ1:

Data for travel time:

	CD	1	2	3	4
CD	0	334	334	441	225
1	334	0	144	108	128
2	334	144	0	187	121
3	441	108	187	0	226
4	225	128	121	226	0

Data for u_{il} , demand, and time window

u_{il}	d_i	a_i	b_i
1	10	693	1008
1	5	707	995
2	20	563	1110
1	17	684	1248

CZ2:

Data for travel time:

	CD	5
CD	0	196
5	196	0

Data for u_{il} . Demand, and time window

u_{il}	d_i	a_i	b_i
2	8	677	1062

Appendix-B

In this appendix, the application of TSA-VRSP neighbourhood selection mechanism is illustrated. Assume there are 5 destinations (customers) 1,2,3,4, and 5, and a minimum transportation cost route to cover these destinations needs to be determined. The TS rules employed by the algorithm are as follows:

- Top 3 candidates are used.
- The tabu list size is 4.
- Adjacent pairwise interchange is used to generate the neighbourhood.
- The neighborhood selected is the one with the lowest cost and zero tabu value.
- The maximum number of iterations (generations) is 3.

In the initial stage, the initial solution is 1-2-3-4-5 with cost 20. The tabu list is set up as default, and all values are zero. Then, based on the initial route (initial seed), and using adjacent pairwise interchange mechanism the neighborhood of the seed is generated, which is:

2-1-3-4-5 (cost=19; swap=1, 2)

1-3-2-4-5 (cost=17; swap=2, 3)

1-2-4-3-5 (cost=16; swap=3, 4)

1-2-3-5-4 (cost=18; swap=4, 5)

Next the neighborhoods are sorted in descending order (of cost) and the top 3 candidates are selected. The top 3 candidates are:

1-2-4-3-5 (cost=16)

1-3-2-4-5 (cost=17)

1-2-3-5-4 (cost=18)

Of the top 3 candidates, the neighborhood with the lowest cost is 1-2-4-3-5 with cost of 16 and the swap pair (3, 4). Since this neighborhood satisfies the zero tabu value requirement, it is selected to become the seed in the next iteration.

Stage: Initial

						cost	swap
seed						20	
Neighborhood							
1	2	1	3	4	5	19	(1,2)
2	1	3	2	4	5	17	(2,3)
3	1	2	4	3	5	16	(3,4)
4	1	2	3	5	4	18	(4,5)

Top 3 candidates

No	Neighborhood	Cost	Swap
1	12435	16	(3,4)*
2	13245	17	(2,3)
3	12354	18	(4,5)

Tabu List

	1	2	3	4	5
1	0	0	0	0	0
	2	0	0	0	0
		3	0	0	0
			4	0	0
				5	0

At the next iteration, the seed for the stage is 1-2-4-3-5. The tabu list must also be updated, so in the tabu list, the pair of (3, 4) must be valued at 4 (the maximum tabu value, as decided by the user), while the other pairs are still kept at zero. Then from this seed, newer neighborhoods are generated as follows:

2-1-4-3-5 (cost=14, swap =1, 2)

1-4-2-3-5 (cost=13, swap =2, 4)

1-2-3-4-5 (cost=20, swap =4, 3)

1-2-4-5-3 (cost=15, swap =3, 5)

Once again the neighborhoods are sorted in descending order of cost and the top

3 candidates selected. The top 3 candidates are:

1-4-2-3-5 (cost=13)

2-1-4-3-5 (cost=14)

1-2-4-5-3 (cost=15)

From the top 3 candidates, the neighborhood with the lowest cost is 1-4-2-3-5 with cost of 13 and the swap (2, 4). Since this neighborhood satisfies the zero tabu value requirement, it is selected as the best neighborhood and becomes the seed for the next iteration.

Stage: Iteration 1

						cost	swap
Next seed:						16	
Neighborhood:							
1	2	1	4	3	5	14	(1,2)
2	1	4	2	3	5	13	(2,4)
3	1	2	3	4	5	20	(4,3)
4	1	2	4	5	3	15	(3,5)

Top 3 candidates:

No	Neighborhood	Cost	Swap
1	14235	13	(2,4)*
2	21435	14	(1,2)
3	12453	15	(3,5)

Tabu List:

	1	2	3	4	5
1	0	0	0	0	0
	2	0	0	0	0
		3	0	4	0
			4	0	0
				5	0

During the second iteration, the seed is 1-4-2-3-5. The tabu list also must be updated, so in the tabu list, the pair of (2,4) is valued at 4 (the maximum tabu value), the tabu value of pair (3,4) is 3, while the other pairs are still valued at zero. From this seed, the generated neighborhoods are:

4-1-2-3-5 (cost=15, swap =1, 4)

1-2-4-3-5 (cost=17, swap =4, 2)

1-4-3-2-5 (cost=12, swap =2, 3)

1-4-2-5-3 (cost=13, swap =3, 5)

The neighborhoods are again sorted in descending order and the top 3 candidates selected, which top are:

1-4-3-2-5 (cost=12, swap =2, 3)

1-4-2-5-3 (cost=13, swap =3, 5)

4-1-2-3-5 (cost=15, swap =1, 4)

From the top 3 candidates, the neighborhood with the lowest cost is 1-4-3-2-5 with a cost of 12 and the swap of (2,3). Since this neighborhood satisfies the zero tabu value requirement, this neighborhood is selected as the best neighborhood and becomes the seed for the next iteration.

Stage: Iteration 2

						cost	swap
Next seed:						13	
Neighborhood:							
1	4	1	2	3	5	15	(1,4)
2	1	2	4	3	5	17	(4,2)
3	1	4	3	2	5	12	(2,3)
4	1	4	2	5	3	13	(3,5)

Top 3 candidates:

No	Neighborhood	Cost	Swap
1	14325	12	(2,3)*
2	14253	13	(3,5)
3	41235	15	(1,4)

Tabu List:

	1	2	3	4	5
1	0	4	0	0	0
	2	0	0	4	0
		3	0	3	0
			4	0	0
				5	0

For the third iteration, the seed used is 1-4-3-2-5 (lowest cost neighbourhood from the previous iteration). The tabu list also must be updated, with the pair (2,3) valued at 4 (the maximum tabu value), the tabu values of pairs (2,4) and (3,4) are valued at 3 and 2 respectively, while the other pairs are still valued at zero. Then from this seed, the following new neighborhoods are generated:

4-1-3-2-5 (cost=11, swap =1, 4)

1-3-4-2-5 (cost=9, swap =4, 3)

1-4-2-3-5 (cost=10, swap =3, 2)

1-4-3-5-2 (cost=12, swap =2, 5)

Applying the same procedure as before and selecting the top 3 lowest cost neighborhoods leads to:

1-3-4-2-5 (cost=9, swap =4, 3)

1-4-2-3-5 (cost=10, swap =3, 2)

4-1-3-2-5 (cost=11, swap =1, 4)

From the top 3 candidates, the neighborhood with the lowest cost is 1-3-4-2-5 with cost of 9 and the swap (4,3). However, this neighborhood does not satisfy the zero tabu value requirement since the pair is valued at 2. The neighborhood selection next turns to the second lowest cost sequence, which is 1-4-2-3-5 with cost of 10 and the swap (3, 2). But this neighborhood also does not satisfy the zero tabu value requirement since the pair is valued at 4. Next the neighborhood considers the third lowest cost sequence 4-1-3-2-5 with cost of 11 and the swap (1,4). Since this neighborhood satisfies the zero tabu value requirement, it is selected as the best neighborhood. As the number of iterations by this time has reached the maximum number (set at 3), the search terminates. The last selected neighbourhood becomes the solution of the TS algorithm, which is route 4-1-3-2-5 with cost of 11.

Stage: Iteration 3

						cost	swap
Next seed:						12	
Neighborhood:							
1	4	1	3	2	5	11	(1,4)
2	1	3	4	2	5	9	(4,3)
3	1	4	2	3	5	10	(3,2)
4	1	4	3	5	2	12	(2,5)

Top 3 Candidates:

No	Neighborhood	Cost	Swap
1	13425	9	(4,3)
2	14235	10	(3,2)
3	41325	11	(1,4)*

Tabu List:

	1	2	3	4	5
1	0	3	0	0	0
	2	0	4	3	0
		3	0	2	0
			4	0	0
				5	0

Search terminated. The solution of TS is the route 4-1-3-2-5 with cost of 11.

TSA-VRSP

Step 1: Initial Phase

1.1 Set $t1=0$

1.2 Generate the initial outbound truck route ($Z_{t1} = Z_0$) using Greedy algorithm.

Set Z_0 as Z_0^{best} and $Z_0^{current}$.

1.3 Calculate the transportation cost of initial outbound truck route (TC_{t1}). Set

TC_{t1} as $TC_{t1}^{current}$ and TC_{t1}^{best} .

1.4 Determine the outbound truck schedule (XY_{t1}), this is the outbound truck sequence and assignment to the outbound dock door.

1.5 Go to TTC calculation routine

1.6 Set $TTC_{t1}^{current}$ as TTC^{best}

1.7 Put the initial solution to *Solution_best*

Step 2: Applying Tabu Search in TSA-VRSP

2.1 set $t1=t1+1$

2.2 Apply adjacent pairwise interchange mechanism to generate $N1$ number of neighborhoods

from $Z_{t1-1}^{current}$ (the seed). Get $Z_{t1}^1, Z_{t1}^2, \dots, Z_{t1}^{N1}$.

2.3 Set $n1=1$

2.4 Calculate the transportation cost of Z_{t1}^{n1} , this is TC_{t1}^{n1} .

2.5 Determine the outbound truck schedule (XY_{t1}^{n1}), this is the outbound truck sequence and assignment to outbound dock door.

2.6 Go to TTC calculation routine

2.7 Put to set of neighborhoods solutions (NS_{t1}^{n1}).

2.8 Check whether $n1 > N1$?

If no: Set $n1 = n1 + 1$. Then go to step 2.4.

If yes: go to step 2.9.

2.9 Get a set of Neighborhoods solutions ($NS_{t1}^1, \dots, NS_{t1}^{N1}$). Then sorts in descending orders based on the total cost, since TSA-VRSP is a lowest cost problem.

2.10 Determine the top k candidate solutions

2.11 Set $i = 1$ (i is the ranking of candidate solution that start from 1, this is the candidate with the lowest total cost).

2.12 Check whether $(Z_{t1}^{n1})^i$ is tabu?

If yes, go to step 2.13

If not, go to step 2.14

2.13 set $i = i + 1$ (go to the candidate with lower cost). Then go to step 2.12 again.

2.14 Set $(NS_{t1}^{n1})^i$ as $NS_{t1}^{current}$

2.15 Update tabu list

2.16 Check whether $TTC_{t1}^{current} < TTC^{best}$

If no: $TTC^{best} = TTC^{current}$

$Solution_best = Solution_current$

If Yes: $TTC^{best} = TTC_{t1}^{current}$

$Solution_best = NS_{t1}^{current}$

2.16 Stopping criteria is met?

If no: go to (repeat) step 2.1

If yes: get solution for VRSP this is $Solution_best$ and go to step 3.

Step 3: Stop TSA-VRSP

Figure B.1 The pseudo-code of TSA-VRSP

The pseudo-code of TTC calculation routine is described below.

TTC calculation routine

Step 1: Set $t2=0$

Step 2: Set $ECLC_{t1}^{best} = M$ and $OTS_{t1}^{best} = M$

Step 3: Set $t2=t2+1$

Step 4: Determine $OTS_{t1\ t2}$; Calculate $ECLC_{t1\ t2}$

Step 5: Check whether $ECLC_{t1\ t2} < ECLC_{t1}^{best}$?

If No: set $ECLC_{t1}^{best} = ECLC_{t1-1}^{best}$; set $OTS_{t1}^{best} = OTS_{t1-1}^{best}$; Go to step 6

If Yes: set $ECLC_{t1}^{best} = ECLC_{t1\ t2}$; set $OTS_{t1}^{best} = OTS_{t1\ t2}$; Go to step 6

Step 6: Check whether $t2 > T2$?

If No: Go to (repeat) step 3

If Yes: Determine ITS_{t1} and calculate IC_{t1} ; calculate $TTC_{t1}^{current}$; Go to

step 7

Step 7: Stop TTC calculation routine

Figure B.2 The pseudo-code of TTC calculation routine in TSA-VRSP