An Adaptive Large Neighborhood Search Algorithm for Selective and Periodic Inventory Routing Problem

by

Özge TÜNCEL

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

Koç University

August, 2013

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Özge TÜNCEL

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

____

Assoc. Prof. F. Sibel SALMAN (Advisor)

____

Assoc. Prof. Deniz AKSEN

____

Assoc. Prof. Semra AĞRALI

Date ________________________________

ABSTRACT

In this thesis we develop the first metaheuristic method for a selective and periodic inventory routing problem (SPIRP) that arises in reverse logistics. The problem concerns a biodiesel production company collecting used vegetable oil from restaurants and hotels which are the source nodes using and wasting vegetable oil in considerable amounts. The production facility reuses the waste oil as raw material to produce biodiesel and meets the raw material requirement for each day from daily collection, inventory and by purchasing oil. The manager needs to decide which of the present source nodes to include in the collection program, and which periodic routing schedule to repeat in every planning horizon to visit these nodes accumulating vegetable oil. His objective is to minimize the total collection, inventory and purchasing costs while the production requirements and operational constraints are met. Recently, a flow-based mixed integer linear programming (MILP) formulation was proposed for this problem, and solved on a real-world case with up to 40 source nodes. However, it was observed that the average optimality gap attained by the commercial MILP solver in three hours exceeds 10% when there are more than 25 nodes present. In order to solve large sized instances of SPIRP more effectively in a reasonable time, we develop an Adaptive Large Neighborhood Search (ALNS) algorithm by using a rich neighborhood structure comprised of 11 distinct moves. Some of these moves modify the visiting schedule and vehicle routes, while others change also the subset of visited source nodes. We test our algorithm on small size instances and compare the results with the MILP model. While our algorithm solves the small instances in several seconds, the MILP model runs for hours to find similar results. When the number of source nodes is 30 and more, our algorithm outperforms the MILP model. We also test our algorithm on larger instances with up to 100 nodes and present the related computational results. For the instances with 50 to 100 nodes, the problem is solved with around 10.7% gap.

ÖZETÇE

Bu çalışmada, tersine lojistik alanında karşımıza çıkan seçici ve periyodik envanter rotalama problemi için literatürdeki ilk sezgisel methodu geliştiriyoruz. Bu problemde, restoran ve otel gibi büyük miktarda bitkisel yağ tüketimi yapan ve İstanbul'un anadolu tarafına yayılmış bu kaynak noktalarından atık bitkisel yağ toplayan bir biyodizel üretim tesisini inceliyoruz. Toplanan atık yağlar bu tesiste biyodizel üretmek için hammadde olarak kullanılmaktadır. Üretim tesisinin yöneticisi mevcut kaynak noktalarından hangilerini atık toplama programına dahil edilmesi gerektiğine; hangilerinin her gün ziyaret edilmesi gerektiğine; sonsuz süre zarfında hangi periyodik rotalama çizelgesinin tekrarlanması gerektiğine karar vererek, üretim gereksinimleri ile operasyonel kısıtlar altında araç kullanımı, rotalama, envanter ve satın alma maliyetlerin toplamını minimize etmeyi amaçlamaktadır. Bu seçici ve periyodik envanter rotalama problemi için yakın geçmişte ilk olarak akış tabanlı bir doğrusal tamsayılı programlama (DTP) modeli geliştirildi ve 40 kaynak noktasına kadar gerçek hayat problemleri üzerinde test edildi. Kaynak nokta sayısı 25'i aştığında, bu modelin 3 saat limitli çözümlerinin uygunluk düzeyinin %10'u aştığı belirtildi. Bu problemi daha fazla kaynak sayısı ile uygun zaman limitleriyle daha etkili çözebilmek için 11 farklı komşu yapısından oluşan bir uyarlanmış geniş komşu arama sezgisel algoritması geliştirdik. Bazı komşu yapıları kaynak noktasının ziyaret programını ve araçların rotalamasını modifiye ederken, diğerleri ziyaret edilen kaynak nokta listesini de değiştirebiliyor. Algoritmamızın sonuçlarını DTP modeli ile karşılaştırdığımızda, bizim methodumuzun birkaç saniye içinde bulduğu çözümü DTP modelinin bir kaç saate bulduğunu gördük. Kaynak nokta sayısı 30'u geçtiğinde bizim algoritmamız, DTP modelinden daha iyi sonuçlar vermektedir. Aynı zamanda, algoritmamızı 100 kaynak noktasına kadar büyük problemlerde de test ettik. 50 ile 100 arasında kaynak noktasına sahip problemlerin çözümlerini hesapladığımız alt limitlerle karşılaştırdığımızda, algoritmamızın bu problemleri ortalama %10.7 uygunluk düzeyinde çözdüğünü gözlemledik.

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my supervisors, F. Sibel Salman, Onur Kaya, and Deniz Aksen. Their expertise, understanding and patience contributed considerably to my thesis and graduate experience. I consider myself very lucky to have advisors such as them and I am also very proud of being one of their students.

I am also very grateful to Semra Ağralı for taking part in my thesis committee and for their valuable suggestions and comments.

I would like to thank to TUBITAK (The Scientic and Technological Research Council of Turkey) for providing the graduate scholarship during my M.S. study.

I would like to thank to all of my friends for their support and kindness. I would like to thank specially to Pelin Doğan, Hadi Feyzollahi, and Maziar Kasaei and my brother Emre Tüncel for their tremendous help in my thesis.

And I would like to thank my parents and my brothers for believing in me, guiding me and supporting me through my whole life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

SPIRP	Selective and Periodic Inventory Routing Problem
MILP	Mixed Integer Linear Programming
ALNS	Adaptive Large Neighborhood Search
PVRP	Periodic Vehicle Routing Problem
IRP	Inventory Routing Problem
VRP	Vehicle Routing Problem
CVRP	Capacitated Vehicle Routing Problem
DCVRP	Distance-Constrained Capacitated Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Windows
VRPB	Vehicle Routing Problem with Backhauls
VRPBTW	Vehicle Routing Problem with Backhauls and Time Windows
VRPPD	Vehicle Routing Problem with Pickup and Delivery
VRPPDTW	Vehicle Routing Problem with Pickup and Delivery and Time Windows
MCVRP	Multi-Compartment Vehicle Routing Problem
MDPVRP	Multi-Depot Period Vehicle Routing Problem
PVRPIF	Periodic Vehicle Routing Problem with Intermediate Facilities
PVRP-SC	Periodic Vehicle Routing Problem with Service Choice
SIRP	Stochastic Inventory Routing Problem
LP	Linear Programming

VMI	Vendor-Managed Inventory
ML	Maximum Level Policy
OU	Order-Up-to Level Policy
VNS	Variable Neighborhood Search
NEARP	Node, Edge and Arc Routing Problem
GLS	Guided Local Search
SA	Simulated Annealing
IF	Intermediate Facilities
LNS	Large Neighborhood Search

Chapter 1

INTRODUCTION

In our study we propose the first heuristic method for a recently introduced routing and scheduling problem. The problem is based on a case study about a biodiesel production facility (the company) in Istanbul that collects waste vegetable oil from source nodes at different locations throughout the city to use the oil as input in biodiesel production. The source nodes include businesses that consume cooking oil in large volumes, such as restaurants, and hotels. The company makes an agreement with each source node and specifies on which days of the week the accumulated waste oil will be collected. Waste vegetable oil accumulates with different rates at the source nodes and uncollected amount at any day is kept till the next visitation day.

The company has a predetermined daily production plan and needs to obtain the input materials to follow this plan. Thus, the production plan creates a necessity to obtain the daily input requirements for vegetable oil. The company can satisfy the vegetable oil needed for biodiesel production either by the waste vegetable oil collection or by purchasing virgin oil. Purchasing virgin oil is considered to be more costly, but waste vegetable oil collection also has a significant cost due to the utilized vehicles, drivers, fuel, etc. Thus, the manager needs to decide on how much waste vegetable oil to collect, if possible, from source points and how much to purchase on each day, depending on the available inventory at hand, in order to satisfy the input requirements for production. The manager also needs to decide on the route of each vehicle in order to make the collection at the minimum possible cost. Moreover, the amount of waste vegetable oil accumulated at the source nodes might be more than the amount needed for production or more than the production capacity. In such cases visiting all the source nodes will not be necessary. Hence, the manager has to decide with which of the source nodes they should make a collection agreement. The company can also keep an inventory at its production. The objective is to find the decisions considering which of the potential source nodes to include in the collection program, which of them to visit on each day, which periodic routing schedule to repeat over an infinite horizon and how many vehicles to operate such that the total collection, inventory and purchasing costs are minimized while the production requirements and operational constraints are met. This considerably hard routing and scheduling problem has been recently defined as the Selective and Periodic Inventory Routing Problem (SPIRP) by Aksen et al. [1] in 2012.

For this problem, we apply an Adaptive Large Neighborhood Search (ALNS) algorithm. As the problems in the literature get more complicated to deal with real life issues, the ability of the existing methods to escape local optima has become insufficient. Therefore, the emergence of more sophisticated solution methods became necessary. ALNS is in the class of large neighborhood search algorithms. Recent literature has shown us the effectiveness of the large neighborhood search mechanisms especially on routing and scheduling problems. Searching larger and more complicated neighborhoods that can escape local optima more effectively is the basic idea of these algorithms. Dealing with a larger neighborhood gives a chance to span a larger proportion of the solution space, which in return helps to find better objective values. This characteristic is a disadvantage in terms of the time it takes the method to perform. To avoid this problem, the use of these large neighborhoods is limited into a subset of the search space. Furthermore, the ALNS algorithm uses several moves interchangeably throughout the algorithm unlike many other metaheutistics. The chance of a move to be used for the next iteration depends on the past performance of the move itself. If a move updates the best or the current solution, the probability of that move to be chosen for the later iterations increases.

We present a wide literature review in Chapter 2. In Chapter 3 we give the problem definition and provide the mixed integer linear programming model. In Chapter 4 we focus on the heuristic method we apply to the problem; Adaptive Large Neighborhood Search algorithm. In Chapter 5 we apply the heuristic method to the problem. We generate 54 instances with 20 to 100 source nodes using real life data. We test our algorithm and compare its performance with that of MILP introduced by Aksen et. al. [1]. Finally, in Chapter 6 we give our concluding remarks and discuss brief directions for future work.

Chapter 2

LITERATURE REVIEW

Rapid advances and complexity in technology changed the shape of competition. The effects of traditional competition elements, quality and cost, are reduced. To survive the competition, the companies have to reduce cost while improving services by considering social and economic factors related to their supply chain. All aspects of the supply chain became the focus of attention to win the competition, from raw materials to recycling. Moreover, companies started to realize the importance of sustainable supply chain management and reverse logistics to increase quality and profitability. It is stated that the unit cost of remanufacturing can be about 40-60% of the unit manufacturing cost of an original product in some industries like transportation, automotive, and construction [2]. The next step for the efficient and comprehensive supply chain designs of the future now leans on the sustainable supply chain management and reverse logistics [3].

Besides the heating competition between the rival companies in all sectors, the other important motivation pushing companies towards reverse logistics and sustainable systems is limited natural resources. The need for energy is increasing due to increases in industrialization and population all over the world. The basic sources of this energy have been petroleum, natural gas, coal, hydro, and nuclear. Petroleum diesel is the major fuel source worldwide and almost 50% of the petroleum diesel is used in the transportation sector. However, the source of petroleum, the fossil fuel reserves are decreasing day by day. Moreover, petroleum diesel creates atmospheric pollution. Since the use of petroleum does not seem like the best choice anymore, the need for an alternative source of energy became unavoidable [4]. As the new source of energy for transportation, biodiesel can be a substitute for petroleum. Biodiesel is a nontoxic and biodegradable alternative fuel. While the cost of virgin oil used in the production of biodiesel constitutes 85% of the total production cost, Gonzalez et al. [5] and Predojevic [6] state that collecting and using waste vegetable oil costs almost half the price of using virgin vegetable oil in biodiesel production. Therefore, waste cooking oil is an economical choice for biodiesel production, because of its availability and low cost. The studies show that biodiesel obtained from waste oil gives better engine performance and less emission with respect to virgin oil when tested on commercial diesel engines. [4]

The studies show that using waste vegetable oil in the production of biodiesel is important for an environmentally and economically sustainable system [7]. In the USA, every year between 4.5 billion and 11.3 billion liters of cooking oil is used; and in Japan, the amount of waste cooking oil generated is between 400 and 600 thousand tons [8]. While worldwide 108 billion liters of waste oil is estimated to be generated annually, only 6 billion liters of the waste oil are collected to be used in the production of biodiesel [9]. In Turkey, every year almost 390 thousand tons of cooking oil is wasted, which could have been used to produce biodiesel that can meet 5% of total diesel fuel consumption in Turkey. This much biodiesel can help Turkey to save about 300 million dollars per year. Besides, collecting waste vegetable oil decreases the contamination of rivers, lakes or oceans. It is stated that one liter of waste oil poured down the drain can contaminate one million liters of water and cause serious damage to the environment and the ecological life [9].

The economic and environmental importance of reverse logistics has attracted the attention of researchers and there are now lots of case studies in the literature for different real-life problems [10]. Fleischmann [11] analyzes logistic network design in a reverse logistic content and the article presents a generic facility location model. Teixeira [12] analyzes a case study of planning vehicle routes for the collection of urban recyclable waste and develops heuristic techniques to create collection routes for every day of the month while minimizing the operation cost. Repoussis [13] presents a web-based decision support system for efficiently and effectively managing waste lube oil collection and recycling operations. They apply their system to a real-life industrial environment and show improved productivity and competitiveness, proving the applicability of their method on real-life reverse logistic planning problems.

Our problem is another real-life application of reverse logistics. Periodic vehicle routing problem (PVRP) and inventory routing problem (IRP) are two of the research topics that are related to our study and widely studied in the literature. These problems are extended versions of the vehicle routing problem (VRP).

2.1 VRP

The Vehicle Routing Problem (VRP) is a combinatorial optimization and integer programming problem consisting in designing the optimal set of routes for a fleet of vehicles in order to serve a given set of customers. This well-known problem in the fields of transportation, distribution and logistics was introduced by Dantzig and Ramser in 1959 [14]. Real-life application of VRP is studied in many sectors such as garbage collection, street cleaning, school bus routing, mail delivery, task sequencing, and collection of household waste, gasoline delivery, goods distribution and snow plough. In fact, the studies of Maffioli [15], Toth and Vigo [16] declare that using VRP algorithms in distribution processes helps companies save between 5% and 20% in transportation cost.

The classical Vehicle routing problem (VRP) is defined on an undirected graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ where $\mathrm{V} = \{\mathrm{v}_0; \mathrm{v}_1; \ldots; \mathrm{v}_\mathrm{n}\}$ is vertex set and $\mathrm{E} = \{(\mathrm{v}_\mathrm{i}; \mathrm{v}_\mathrm{j}) : \mathrm{v}_\mathrm{i}; \mathrm{v}_\mathrm{j} \in \mathrm{V}; \mathrm{i} < j\}$ is an edge set. The depot is represented with v_0 and the other vertices represent customers. The depot houses a fleet of vehicles with capacity of Q. The distance or the travel time matrix is described with $\mathrm{C} = (\mathrm{c}_{\mathrm{ij}})$. Elements in C represent the travel cost or the travel time between each pair of customers and between the depot and the customers. It can be asymmetric or symmetric. Moreover, customers have demands and service times to be met.

The solution of VRP is defining a set of routes, each traversed by a single vehicle that starts and ends at the depot such that each customer is visited exactly once by exactly one vehicle, total demand of the route does not exceed the vehicle capacity, and total duration or length of any route does not exceed a preset bound; while requirements of the customers are fulfilled and the global transportation cost is minimized. The original graph is transformed into a complete graph, whose vertices are the customers and the depot.

The composition and size of the fleet of vehicles can be fixed or can be defined according to the requirements of the customers. Moreover, a fixed cost associated with the utilization of the vehicle can be added to the problem.

The problem is studied for more than 50 years since it was introduced by Dantzig and Ramser [14] with a real-world application concerning the delivery of gasoline to gas stations. They propose the first mathematical programming formulation and algorithmic approach for the solution of the problem. Soon after, an effective greedy heuristic algorithm was introduced by Clarke and Wright [17]. Their algorithm make improvements on the Dantzig-Ramser approach. Taking these papers as basis, several exact and heuristic methods were proposed finding the optimal and approximate solutions for several versions of VRP.

The first book devoted to the Vehicle Routing Problem was published in 1971 [18]. Since then, a number of books dealing with Vehicle Routing Problem have been published [19], [20], [21]. Moreover, several survey papers were published over the years; [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36].

2.1.1 Variants of Vehicle Routing Problem (VRP)

The variants of the VRP which have received great attention in the scientific literature are:

- Capacitated VRP (CVRP)
- Distance-Constrained CVRP (DCVRP)
- VRP with Time Windows (VRPTW)
- VRP with Backhauls (VRPB) and with Time Windows (VRPBTW)
- VRP with Pickup and Delivery (VRPPD) and with Time Windows (VRPPDTW)
- Multi-compartment Vehicle Routing Problem (MCVRP)
- Periodic Vehicle Routing Problem (PVRP)
- Inventory Routing Problems
- Stochastic VRP

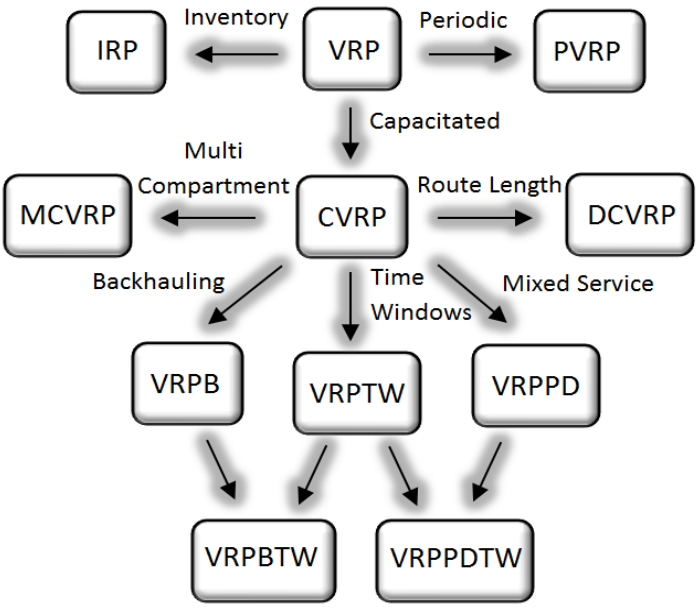

Figure 2. 1: The basic problems of the VRP class and their interconnections[21]
(An arrow moving from problem A to problem B means B is an extension of A)

2.1.2 Common Properties of the Vehicle Routing Problems (VRPs)

Routing and scheduling problems including all these VRP variants have some common characteristics such as:

- Size of available fleet; one or multiple,
- Type of available fleet; homogenous or heterogeneous,
- Housing of vehicles; single depot or multiple depots,
- Nature of demands; deterministic or stochastic or partial satisfaction of demands allowed,
- Underlying network; directed or undirected or mixed or Euclidean,
- Vehicle capacity restrictions; limited or unlimited,
- Maximum route times; same for all routes or different for all routes or not imposed,
- Operations; pickups only or deliveries only or mixed (pickups & deliveries) or split deliveries (allowed or disallowed),

- Planning Horizon; single period or multiple periods,
- Time windows; one sided or two sided or soft windows or hard windows,
- Costs; variable/routing cost or fixed operating/variable acquisition costs or common carrier cost (for unserviced demands),
- Objectives; single-objective or multi-objective.

2.1.3 Solution Methods for the Vehicle Routing Problem

Many researchers worked on exact solution methods on VRP. However, VRP is an NP-hard problem. To solve the problem to obtain optimal solution, all solution alternatives have to be enumerated. Thus, it is not expected to develop exact solution methods that can solve a VRP instance in reasonable amount of running time. Therefore; besides exact methods, lots of heuristic algorithms have been proposed. The most well-known proposed solution methods are:

Exact Solution Methods

Some methods developed and widely used as exact solution methods are Branch-and-Bound [37], [38], [39], [16], Branch-and-Cut [40], [41], [42], and Branch-and-Cut-and-Price [43], [44].

Heuristic Methods:

Simulated and deterministic annealing, tabu search, GRAPS, genetic algorithms, adaptive memory, ant colony optimization, neural networks, large scale neighborhood search and hybridizations of these methods.

2.2 PVRP

Most of the real problems which need pick-up and/or delivery operations, customers generally require frequent visits over a planning horizon. This creates demand to development of Periodic Vehicle Routing Problem (PVRP) also called allocation/routing problems. In classical VRPs the planning horizon is most of the time limited with very short time horizon, such as 8-10 hours or a day. In the case of the PVRP, the classical

VRP is generalized by extending the planning horizon to M days. (Different time units can used as well) PVRP is a more realistic and complex variation of VRP. A wide range of real life applications could be defined as PVRP such as courier services, elevator maintenance and repair, vending machine replenishment, the collection of waste and the delivery of interlibrary loan material [45].

In the Periodic VRP, vehicle routes are constructed over multiple days, assuming for each vehicle, one vehicle route represents a day. Within the planning period, each day a number of vehicles travels on their routes starting from and ending at the depot and visiting a number of customers in between. The objective of the problem is to minimize the total distance travelled within the planning horizon.

In the periodic vehicle routing problems deliveries are made to a set of customers over multiple time units during the period and optimizing these iterative operations can result in significant cost savings. According to the underlying complete graph $G = (N; A)$ we can find the distances among all arcs; therefore, by using these distances we can calculate the travel costs. All the nodes N, including depot and customers are visited with predetermined frequencies over the planning period. In most of the PVRP models, researchers propose a set of schedules which are a collection of time units within the planning period in which customers receive service. During the planning period by choosing one of these schedules customers can be visited several times and the visiting frequencies k for each customer may be in a predetermined interval $1 \leq k \leq M$.

In general, after creating a set of schedules, the PVRP is viewed as a multi-stage combinatorial optimization problem combining two defined problems: the assignment problem and the vehicle routing problem. PVRP involves three simultaneous decisions:

- Select a schedule from a candidate set of schedules for each node
- Assign a set of nodes to be visited by each vehicle on each day
- Route the vehicles for each day of the planning period

In the classic VRP, only the last two decisions are needed to be made, and over a single day only. In the PVRP, each customer needs to be visited several times with a frequency of f_i during the planning horizon. [61].

The first PVRP model was introduced in 1974 by Beltrami and Bodin for assigning hoist compactor trucks in municipal waste collection [46]. They propose heuristics to solve the PVRP, but do not present any model, just enlighten its complexity in comparison of the Classical VRP. Their heuristic method is based on cluster-first and route-second method.

The first formal definition of PVRP was introduced by Russell and Igo [47] in 1979 as “Assignment Routing Problem” and they introduced a Mixed Integer Linear Programming (MILP) model. They draw attention to the difficulties of choosing a schedule for each customer besides solving the routing problem.

The second formal definition was introduced by Christofides and Beasley [48] in 1984 as the generalization of vehicle routing problems over a planning horizon where each customer has a number of visit requirements over the horizon. They present an integer programming model which considers both assignments of schedules to the customers and routing of a vehicle at each time unit.

In summary, two viewpoints have emerged in defining the PVRP: Russell and Igo [47] and Tan and Beasley [49] approach the problem as an extension of the assignment problem with a routing component; Christofides and Beasley [48] formulate the PVRP as a routing problem with a selection decision involved.

2.2.1 Variants of the Periodic Vehicle Routing Problem

Mainly the literature includes three variants related to the Periodic Vehicle Routing Problems:

i. Multi-Depot PVRP
ii. PVRP with Time Windows
iii. PVRP with Service Choice

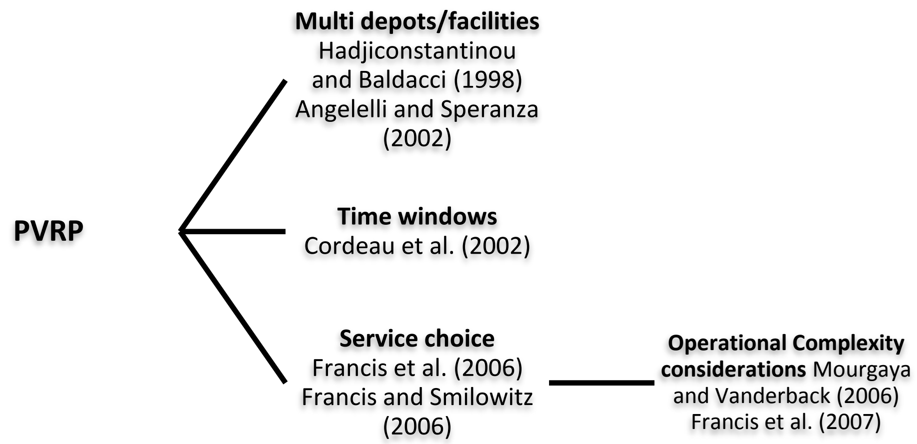

Figure 2. 2: Variants of the PVRP

In the Multi-Depot Vehicle Routing Problem (MDVRP) periodic deliveries are made using a fleet of vehicles that are based across a number of depots. Cordeau et al. [51] present a formulation of PVRP and show that the Multi-Depot Vehicle Routing Problem (MDVRP) is a special case of PVRP by associating depots with days. In their paper Hadjiconstantinou and Baldacci [52] combine the ideas of periodicity and multiple-depots, extending the PVRP to include multiple depots. This greatly increases the difficulty of the resulting problem as it involves the additional decisions of assigning vehicles to depots as well as customer nodes to depots. Their Multi-Depot Period Vehicle Routing Problem (MDPVRP) is the problem of designing a set of routes for each day of a given $D-$day planning period. Each route of day $d \in D$ must be executed by one of a homogenous fleet of K vehicles (service teams visiting customers) based at a certain depot (i.e., it must start and finish at its assigned depot).

The PVRP with Intermediate Facilities (PVRPIF) is similar to the MDPVRP. While Angelelli and Speranza [53] do not allow multiple vehicle depots, they do use the idea of "drop-off points", or intermediate facilities, at which vehicle can stop along their vehicle routes, allowing them to replenish their capacities. Vehicles start and end their routes at their own depots, but visit these intermediate facilities along the way. Such problems arise in applications like waste collection with recycling facilities or goods collection with warehouse facilities. The authors solve the resulting extended PVRP problem using a Tabu search method.

Another variant of PVRP, the PVRPTW is the problem of designing K different vehicle routes such that all customers are visited with their desired service frequency over the planning period, and each visit lies within a specified time interval. It was introduced by Cordeau et al. [54] as the extension of the earlier work by Cordeau et al. [51] including time-windows. The authors modify the Tabu search heuristic presented in Cordeau et al. [51]. The change to the heuristic is minor, principally requiring an additional penalty term to be added to the objective function for violations of time window constraints.

Francis et al. [55] extend the PVRP to make visit frequency a decision of the problem. The extended problem is called the PVRP with Service Choice (PVRP-SC). The problem concerns customers who have a minimum requirement for visits over the period but are willing to accept higher visit frequency as well. This property changes the problem in terms of arrangement of visit frequencies for each customer in a flexible way and this may decrease in the routing costs. This increases the difficulty of solving the problem in two ways: first, there is the added complexity of determining the service frequency; second, the vehicle capacity requirement when visiting a node also becomes a decision of the model.

2.2.2 Common Properties of the Periodic Vehicle Routing Problems

The common properties of the PVRP are defined in general as follows:

Given: A complete network graph $G = (N, A)$ with known arc costs c_{ij}, $\forall(i,j) \in A$, a planning period of D days indexed by d; a depot node indexed $i = 0$; a set of customer nodes $N' = N/\{0\}$ with each node $i \in N'$ having a total demand of W_i over the planning period, and requiring a fixed number of visits f_i; a set of vehicles K each with capacity C; a set of schedules S.

Find: An allocation of customer nodes to schedules such that each node is visited the required number of times; a routing of vehicles for each day to visit the selected nodes during that day; with

Objective: Minimum cost of visiting the nodes. [61].

2.2.3 Solution Methods for Periodic Vehicle Routing Problem

The evaluation of the problem definition and solution methods for the PVRP is explained by Francis, Smilowitz, and Tzur in [61]. The figure below shows the evaluation of the PVRP solution approaches.

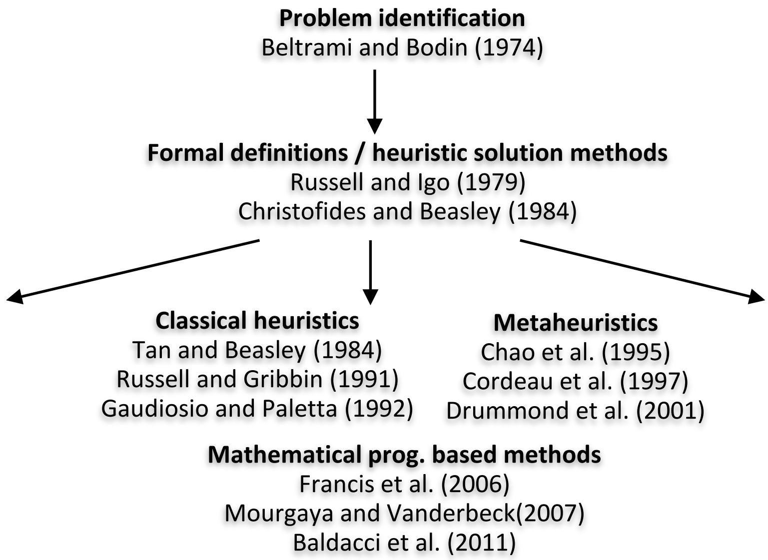

Figure 2. 3: Evaluation of models and solution methods for the PVRP [61].

Both the first definition of the problem and the first heuristic method were presented by Beltrami and Bodin in 1974 [62]. In the paper, Beltrami and Bodin [62] adopt a cluster-first, route-second approach since the agency operating the vehicles "decided a priori the day assignment for each site". After such an a priori assignment, the nodes to be visited on each day of the week are known and independent VRPs are solved for each day of the week.

Two-phase solution methods similar to that of Beltrami and Bodin [46] are commonly found in early heuristics for the PVRP. Recent PVRP literature has focused on metaheuristic methods of solving the problem that can escape the trap of local optimality that plagues conventional heuristics. In this section, we review the classical heuristics, the metaheuristics, as well as recent mathematical programming based approaches to solving the PVRP.

Classical Heuristics

Besides providing a formal definition of PVRP as the "Assignment Routing Problem", Russell and Igo [47] propose three heuristics; an improvement heuristic, and two construction heuristics. The viewpoint presented in Russell and Igo [47] is that the problem is one of picking a valid day combination for a specified service frequency. The first heuristic involves creating route clusters for all days using nodes whose day assignments are fixed. Then, the remaining unallocated nodes are assigned in descending order of required visit frequency. After initial construction, an improvement phase attempts to reassign nodes to other schedules. Their second heuristic is an improvement heuristic that reoptimizes the allocation and routing of nodes. It is a modified version of the MTOUR heuristic for VRP [57]. The third heuristic is an implementation of the Clarke-Wright savings method with additional conditions to ensure that any proposed savings move results in a feasible allocation of nodes to days.

Christofides and Beasley [48] do not attempt to solve PVRP to optimality given the complexity of the problem. They propose a two-stage heuristic method: first, they allocate nodes to days; second, they attempt node exchanges with the aim of minimizing the vehicle routing costs. They have a merit order of nodes according to which they make initial allocations. The idea is to reduce the possibility of infeasible solutions.

Tan and Beasley [49] summarize the results of Beltrami and Bodin [46], Russell and Igo [47], and Christofides and Beasley [48] and propose a problem that can be solved more simply than the PVRP itself. Given the difficulty of solving this problem, Tan and Beasley [49] suggest that the assignment of nodes to vehicles be neglected to reduce the size of the problem. They make the decision of allocating nodes to days in the first phase and the routing decision for each day in the second phase.

Russell and Gribbin [58] propose a solution method that consists of an initial route design using a network approximation, followed by three improvement phases. Their network flow model is similar to the formulation of Tan and Beasley [49]. The first improvement heuristic uses the interchange method of Christofides and Beasley [48] to make improvements in individual tours. The second heuristic applies this interchange idea at the vehicle routing level. Finally, the authors propose a binary integer program to further refine the proposed solution.

Gaudioso and Paletta [59] suggest an alternative heuristic for the tactical problem of minimizing fleet size, rather than the operational problem of reducing distance. They impose constraints on the maximum route duration as well as the vehicle capacity. Gaudioso and Paletta [59] do not impose a schedule set from which to choose day combinations, but instead place restrictions on the minimum and maximum number of days between visits for each node. They note that the distance cost of their solution is usually greater than other PVRP solution methods for two possible reasons: one, their objective is to minimize fleet size and not distance; and, two, they use a simple algorithm to solve the embedded TSP to optimize the routes after nodes have been allocated to delivery combinations.

Metaheuristics

Later, some metaheuristics are introduced starting with Chao et al. [60]. The method uses a relaxation of Christofides and Beasley [48] formulation to find an initial solution. Then, an improvement heuristic is used, moving a node from one schedule to another. If the total distance is reduced, the move is accepted immediately. If not, the new solution is accepted if total distance is under a threshold value. As the algorithm repeats, the threshold value is decreased. Algorithm stops when no more improving move is found.

Cordeau et al. [51] introduce a Tabu search method to solve PVRP. The algorithm gives comparable results with the algorithm of Chao et al. [60]. Their Tabu search method has been modified to use specific insertion and route improvement techniques developed by the authors; however, there is no significant change to the core of the Tabu search technique that is specific to PVRP.

Later, a hybrid metaheuristic combining genetic algorithm and Local Search is introduced by Drummond et al. [61]. It is a combination of genetic algorithm and local search heuristics. The days customers are assigned and accumulated demands are kept as chromosome and fitness value of each chromosome is calculated solving savings method for each individual day. They present a numerical study, comparing their solutions to those of Cordeau et al. [51], providing improved solutions to some problem instances.

A different heuristic approach was introduced in 2007 by Alegre et al [56] based on a real case of Periodic Pick-up of Raw Materials Problem for a manufacturer of auto parts. They state real life problems have longer horizon even as long as 90 days with respect to data sets in the literature. That’s why, their algorithm is designed to perform well especially on PVRPs with longer planning horizons. They adapt scatter search which is an evolutionary algorithm. Scatter search differs from other evolutionary algorithm due to its use of randomness less than others. Their method is also a two-phase method. Even though their initial aim to solve problems with longer planning horizons, they apply the algorithm to the problem instances in the literature as well to evaluate the performance of the algorithm. Their algorithm turns out to perform well and give competitive results with Cordeau et al. [51] and Chao et al. [60]. However, they do not compare their results with Drummond et al. [61] because they state that there are evident errors.

Later, a more sophisticated heuristic method, Variable Neighborhood Search, was applied to PVRP by Hemmelmayr et al. [62]. The method defines several neighborhoods in order to be used in the search and systematically change the neighborhoods in a local search procedure. After determination of initial solution, three steps are performed repetitively. The first step is Shaking, meaning randomly selecting a solution from the first neighborhood. The second step is applying a Local Search algorithm to the solution and finding the local optimum. The last step is deciding whether to move solution by checking if it is better than the current solution. If a new acceptable solution is not found, the steps are applied to the next neighborhood until the solution is found. When new solution is found, the algorithm returns to the first neighborhood and repeats the procedure until stopping criteria is met [62]. They compare their results with literature and state that their results are competitive with the existing solutions. The method performs faster for large instances even though this is not the case for others.

Very recently, Vidal et al. [63] proposed a hybrid genetic algorithm: the population consists of individuals representing feasible and infeasible solutions; the population evolves by applying different operators with the aim of having high quality solutions while maintaining diversity. They test the algorithm on benchmark instances for PVRP and MDVRP and on a new set of instances for the multi-depot periodic vehicle routing problem (MDPVRP), providing very good solutions.

Another very recent approach was proposed in Pacheco et al. [64]; the authors study the problem of a bakery company in northern Spain. In order to minimize the total distance traveled for the daily routes over the week, the bakery company allows some flexibility in the dates of delivery. The authors propose a mixed-integer linear programming model and solve the problem through a two phase algorithm. In the first phase, a set of good and diverse solutions is generated, based on GRASP. In the second phase, path-relinking is applied to improve the solutions. Computational experiments are performed on real-data-based instances. In addition, the authors apply the necessary modifications to treat the problem as a PVRP, in order to compare their algorithm with state-of-the-art algorithms for PVRP [65].

Lastly, in 2013, Cacchiani et al. [65] presented a hybrid optimization algorithm for mixed-integer linear programming, embedding both heuristic and exact components. Their algorithm is based on the linear programming (LP) relaxation of a set-covering-like integer linear programming formulation of the problem, with additional constraints. The LP-relaxation is solved by column generation, where columns are generated heuristically by an iterated local search algorithm. The whole solution method takes advantage of the LP-solution and applies techniques of fixing and releasing of the columns as a local search, making use of a tabu list to avoid cycling. They show the results of the proposed algorithm on benchmark instances from the literature and compare them to the state-of-the-art algorithms, showing the effectiveness of our approach in producing good quality solutions.

Mathematical Programming Based Approaches

Furthermore, mathematical programming based models are implemented by some authors such as [55], [67] and [50].

Francis et al. [55] develop an exact solution method based on Lagrangian relaxation of an integer programming formulation of the PVRP. Their Lagrangian relaxation phase removes the constraints that link the two sets of decision variables, and the problem decomposes into a capacitated assignment subproblem and a number of prize-collecting traveling salesman subproblems. Then, they apply a branch-and-bound phase to reach the optimality. This provides the first known exact solution method, a heuristic method with a bounded gap, and a lower bound for the PVRP class of problems.

Mourgaya and Vanderbeck [67] propose an algorithm which schedules visits and assigns these visits to vehicles but they disregarded sequencing customers will be visited within each time unit for each vehicle. In the algorithm they have two objectives; one is regionalization which is clustering customers geographically for tour lengths and the other one is workload balancing among vehicles. The authors use truncated column generation method with rounding heuristic to solve the model. With this method they solve the instances with 50 - 80 customers with 5 day planning period but not into optimality. They state this range of instances is solved by using metaheuristics in most of the PVRP literature.

In 2011 a new exact algorithm is proposed by Baldacci et al.[50] . It is based on a set partitioning integer linear programming formulation of the problem and on three different relaxations used to derive powerful lower bounds.

The algorithm consists of (i) computing a near-optimal dual solution of the LP-relaxation of the formulation strengthened by valid inequalities, (ii) using this dual solution to generate a reduced integer problem containing all optimal solutions, and (iii) solving the resulting problem using an integer programming solver. They solve the most well-known PVRP instances in the literature. For the instances with up to 76 source nodes, they state they find the optimal solutions. For some instances with higher source nodes, they outperform the best values in the literature as well.

To summarize, the PVRP literature relating to solution methods recognizes that the problem is computationally hard. Research in this area has focused on heuristics for the PVRP. Of the heuristics reviewed, the classical heuristics tend to solve the assignment and routing decisions sequentially. More recent work has focused on metaheuristics and mathematical programming based approaches, recognizing the need to take an integrated approach to the PVRP problems [61].

2.3 IRP

Most of the logistic activities focus on material flow among the companies and processes. Those activities require relating different quantity decisions, such as Inventory Management and Vehicle Routing. The intersections between these areas make the problems of such logistic activities harder. These problems take place in the framework of Vendor-Managed Inventory (VMI), a business activity designed for decreasing logistics expenses and increasing business value. VMI requires supplier assessments on replenishment issues for transporting the goods to consumer. Furthermore decisions of the supplier must rely on unique inventory and supply chain policies. These problems are defined as Inventory Routing Problem (IRP) in the literature. Lately, serious researches are conducted on this topic; [94], [95], [96], [97].

IRPs differ from VRPs greatly. VRPs happen when customers set orders and the delivery company, on a randomly selected day, allocates the orders for the selected day for trucks routes. Furthermore according to the IRP applications, decisions about amount of goods will be delivered, are made by the delivery company rather than the customer. Also customer orders are not placed in IRP applications. As an alternative, the delivery company manages according to the principle that the customers never be run out of product. Also VRPs and IRPs differentiate in the planning horizon. Generally VRPs run for one day, and the single requisite is that all orders must be delivered before the end of the day, whereas IRPs are used for longer horizon. On a daily basis the delivery company decides on which customers to visit and amount of good to be delivered. The main purpose is minimum total cost and customers never be run out of product. For reducing distribution costs the time and volume of the delivery can be vary. Nevertheless, this flexibility as well makes harder to determine a product, approximately an optimal, cost-effective distribution plan. The options are endless because both the customers to serve and amount of good are variables [98].

IRP applications are widely used in different industries such as: The gas distribution industry [99], the petrochemical industry, suppliers of supermarkets [100], [101], department store chains, including Walmart [102], home products, such as Rubbermaid [103], the clothing industry, where vendor managed resupply (VMR) is supported by the

American Apparel Manufacturers Association [104], and the automotive industry (parts distribution) [105], chemical components industry [106], [107] and in the oil and gas industries [108], [109], [110], [111], [112], [113], [114].

2.3.1 Problem Definition

IRPs construct a large class problems and the number of solution approaches that have been suggested forms a larger class. Nevertheless, IRPs all have some basic characteristics. According to the definition, the IRP tackles with the repetitive allocation of one good from one facility to a set of N customers over a known length of planning horizon T. T is infinity for most cases. Customer i uses the good at a given rate of volume U_i (volume per day) and has the competence to retain a local inventory of the good up to a maximum value of C_i. The inventory at customer i is I_{0i} at time 0. A fleet of M homogeneous vehicles, with capacity Q, is presented for the allocation of the goods.

The objective is to minimize the average shipping costs in the planning period without causing stock-outs at any of the customers. Three decisions have to be made:

i. When to serve a customer,
ii. How much to deliver to a customer when served, and
iii. Which delivery routes to use.

Even the basic version of the IRP has some variety of characteristics depending of a particular inventory routing problem. They are presented in Table 2.1.

Table 2.1: Different characteristics of most studied IRPs in the literature [115].

Criteria	**Possible Options**		
Time horizon	Finite	Infinite	
Structure	One-to-one	One-to-many	Many-to-many
Routing	Direct	Multiple	Continuous
Inventory policy	Maximum level	Order-up-to level	
Inventory decisions	Lost sales	Back-order	Non-negative
Fleet composition	Homogeneous	Heterogeneous	
Fleet size	Single	Multiple	Unconstrained

In Table 2.1, time refers to the horizon taken into account by the IRP model. Time can be finite or infinite.

The number of customers and suppliers are variable so the configuration preserve to be one-to-one in when a single supplier supplying to a single customer and one-to-many when single supplier and more than one customers, or seldom, many-to-many for more than one suppliers and more than one customers.

Routing splits in three: direct routing (one customer for each route), multiple (a number of customers in one route) and continuous (without central depot).There is no central depot in most maritime applications.

To decide when to make replenishment at customers, the inventory policies that will be used is also decided beforehand in IRPs. These policies establish how the inventory management is modeled. In the literature, the two most studied inventory policies are the maximum level (ML) policy and the order-up-to level (OU) policy. Under an ML inventory policy, the replenishment level is flexible but bounded by the capacity available at each customer. Under an OU policy, whenever a customer is visited, the quantity delivered is that to fill its inventory capacity.

Another characteristic of IRPs is option of back-ordering. If the inventory level at the customer can be negative, then back-ordering becomes an option, meaning the demand can be met after it is placed. If back-ordering is not an option, then the extra demand is considered as lost sales. In both cases there may exist a penalty for the stock out. In deterministic contexts, the inventory can be forced to be non-negative.

The last two criteria refer to fleet composition and size as in VRPs. The fleet can either be homogeneous or heterogeneous, and the number of vehicles available may be fixed at one, fixed at many, or be unconstrained [114].

2.3.2 Origins of the Inventory-Routing Problem

The studies on IRP first started as variations of models designed for the VRP and heuristics developed for VRP taking inventory costs into consideration. Its study is rooted in the paper of Bell et al. [116] published 30 years ago. The paper deal with the case where only transportation costs are included, demand is stochastic and customer inventory levels must be met. This was followed by a number of variants of the problem defined by the same authors. Later, Federgruen and Zipkin [117] modify the VRP heuristic of Fisher and Jaikumar [118] to accommodate inventory and shortage costs in a random demand environment; Blumenfeld et al. [119] consider distribution, inventory and production set-up costs; Burns et al. [120] analyze trade-offs between transportation and inventory costs, using an approximation of travel costs; Dror et al. [121] study short term solutions. The latter study was extended to stochastic demand by Dror and Ball [122]. The paper of Dror and Levy [123] adapts earlier VRP heuristics to the solution of a weekly IRP, while Anily and Federgruen [124] propose the first clustering algorithm for the IRP. Most of these papers assume that the consumption rate at the customer locations is known and deterministic.

2.3.3 Solution Methods

The inventory routing problem (IRP) is a challenging and intriguing problem that provides a good starting point for studying integration of different components of the logistics value chain, i.e., inventory management and transportation. For the inventory routing problem, several matheuristic and metaheuristic approaches have been developed. Several extensions of the IRP have been introduced. With the new variants of the IRP, several new solution methods were proposed. Coelho et al. [114] define the type of IRPs on which most of the research effort has focused on as the basic versions. The more sophisticated versions are defined as extensions of the basic versions.

In this section first, the solution methods introduced for the basic versions of the IRP are presented. Later, the solution methods for the relatively more well-known variations of the IRP are explained.

Basic Versions

Since the basic IRP is an NP-hard problem, most papers propose heuristics for its solution, but also there are some exact algorithms trying to get the exact results. In the following we will describe the exact and heuristic methods.

i. Exact Algorithms

The first branch-and-cut algorithm for a single-vehicle IRP is proposed in Archetti et al. [125], which are able to solve IRPs with both order-up-to level (OU) and maximum level (ML) policies. They introduce a general model by incorporating both inventory holding cost at customers and supplier. These authors solve instances with up to 50 customers in a three-period horizon and 30 customers in a six-period horizon within two hours of computing time.

After them, Solyalı and Süral [126] improve the model by using stronger formulation with the shortest path networks representing customer replenishments. With the new model, they solve larger problems, such as 15 customers and 12 periods or 12 customers and 9 periods. They also just consider the OU policy in their model.

Besides, rather than just considering OU and ML policies, solving multi vehicle version is proposed recently by Coelho and Laporte [127] and Adulyasak et al. [128]. By using branch-and-cut fashion in the algorithm, they solve instances with up to 45 customers, three periods and three vehicles to optimality with CPLEX.

ii. Heuristic Algorithms

Simple heuristics have been used in early papers on IRP to explore the solution space and decompose it into hierarchical sub-problems, in a way that after finding the solution of one sub-problem, its solution is used in the next step. In this regard, assignment heuristic [121], an interchange algorithm [123], trade-offs based on approximate routing costs [120] are such examples.

Nowadays, the new heuristic algorithms are able to obtain high quality solutions to difficult optimization problems. In the heuristics proposed by Gendreau and Potvin [129] to avoid local optima and thorough evaluation performance, the concept of metaheuristics and local search procedures is used. Raidl et al. [130] by using hybridization of different metaheuristic concepts create even more powerful algorithms. Additionally, matheuristic which is the hybridization of a heuristic and of mathematical programming algorithm increased the quality of the results even further [131].

Bell et al. [116] study a case where only transportation costs are included and also inventory levels must be met at the customers. Dror et al. [121] offer the first algorithmic comparison for IRP with a case that OU policy is applied and customers are only visited once during the planning period. For a weekly IRP, Dror and Levy [123], propose a vertex interchange algorithm. They generate initial solutions to a VRP and try to improve the results using the idea by Dror et al. [121].

Burns et al. [120] propose formulas based on the trade-offs between transportation and inventory costs and prove that under direct shipping the optimal delivery size is the economic order quantity.

Anily and Federgruen [124] and Campbell and Savelsbergh [132] propose the clustering heuristics for IRP. Gallego and Simchi-Levi [133] study the direct deliveries and their long-term effectiveness. In addition, to allow the vehicles to perform more than one route per period Aghezzaf et al. [134] propose the new heuristic algorithm which uses heuristic column generation proposed by Anily and Federgruen [124]. As an extension to their work Raa and Aghezzaf [135] add driving time constrain to the model. In this regard, for a problem with heterogeneous fleet Chien et al. [136] proposed the improvement and construction heuristics.

Abdelmaguid [137] propose a construction heuristic which also considers backlogging and using the genetic algorithm Abdelmaguid and Dessouky [138] get better results for this model. Abdelmaguid et al. [139] review the heuristics for IRP with backlogging.

For a case that a single producer cannot usually meet the demand of the customers, Savelsbergh and Song [140] formulate and solve the problem with several suppliers and trips lasting longer than one period. This problem which is called IRP with continuous moves is solved by using initial solutions which are generated by a randomized greedy heuristic and a local search algorithm is applied on that.

Raa and Aghezzaf [141] develop an algorithm allowing multiple tours which also considers a cyclic planning approach and a long term-distribution pattern. In their algorithm customers are partitioned over vehicles and then for each vehicle, the set of customers assigned to it is partitioned over different tours. To check feasibility, a delivery schedule is made for each partition of customers over tours and each combination of tour frequencies.

Geiger and Sevaux [142] consider identifying pareto-optimal solution by comparing different solutions with respect to the two opposing terms in the objective function. According to the authors, when they change some of the parameters, customer visiting frequency becomes more important, since it's inventory cost becomes low but routing cost becomes expensive, and vice versa.

Michel and Vanderbeck [143] propose a heuristic column generation algorithm to solve a tactical IRP in which customer demands are deterministic and are clustered to be served by different vehicles. This heuristic yields solutions that deviate by approximately 6% from the optimum and increase upon industrial practice by 10% with respect to travel distances and the number of vehicles used.

Campbell et al. [144] propose a two-phase heuristic based on a linear programming model. The model calculates the exact visiting period and quantity to be delivered to each customer and after that customers are sequenced into vehicle routes. It also considers time constraints explicitly but does not include any consideration for the inventory holding costs. Because of the high number of possible routes, the model becomes difficult to solve. An increase in the length of the planning horizon also increases the difficulty of the problem. By limiting the model into small set of routes and aggregating periods toward the end of the horizon they find more reasonable results. The output of first phase shows how much to deliver to each customer in each period of the planning horizon. Then the information from it becomes input for the second phase. Because the result of the second phase is related to the first phase, and the decisions in the phases are taken separately, the second phase can only be optimal with respect to the solution obtained from the first phase.

For a single-vehicle case in which an OU inventory policy is applied by Bertazzi et al. [145]. They propose a fast local search algorithm which decreases the flexibility of the decision maker and restricts the set of possible solutions. They also consider both inventory and transportation costs in their model. Then, by using heuristics they solve the simplified problem. A first step creates a feasible solution, and a second one is applied as long as a given minimum improvement is made to the total cost function. While in this heuristic the optimality gap is larger than 5% it is extremely fast.

By combining tabu search and exact solution of mixed integer linear programs (MILPs) Archetti et al. [146] propose a more involved heuristic to approximate routing decisions. A combination of a tabu search heuristic with four neighborhood search operator and two MILPs is used in the paper. The algorithm searches the neighborhood of current solution and performs occasional jumps to other regions. The algorithm starts from a feasible solution and infeasible solutions are sometime accepted for diversification. With optimality gap of about 0.1% the heuristic performs well on benchmark instances.

The adaptive large neighborhood search (ALNS) is developed by Coelho et al. [147] to solve the IRP as a special case of a broader problem including transshipments. After creating the vehicle routes by ALNS operators, the algorithm determines delivery quantities by using an exact min-cost network flow algorithm. In comparison with Archetti et al. [146], this matheuristic performs a little bit worse, in a case that no transshipments are considered. For the multi-vehicle version of IRP, Coelho et al. [148] propose an extension of the previous algorithm. By approximating the costs of inserting or removing customers from existing solutions through the exact solution of a MILP, better solutions are obtained.

Hewitt et al. [149] propose a fast way to obtain primal solutions using a branch-and-price method. They deal with a maritime IRP with a many-to-many structure and single product. They use heterogeneous fleet of vessels and a finite horizon. They state their method works extremely faster than MILP model and they provide comparable results.

<u>Extensions of the Basic Versions</u>

i. The Production-Routing Problem

Production-Routing Problem (PRP) is an extension of IRP including one more element of the supply chain, meaning production. The PRP integrates inventory and Lot-Sizing Problem over a given planning horizon with the Vehicle Routing Problem to perform the deliveries. Therefore, The PRP combines production and distribution decisions.

The PRP is introduced by Chandra [150] and then Chandra and Fisher [151]. Later, Chandra and Fisher [151], Herer and Roundy [152], Fumero and Vercellis [153], Bertazzi et al. [154], Bard and Nananukul [155], [156] study the PRP. Recently, Archetti et al. [157] and of Adulyasak et al. [158] also work on this topic.

Moreover, other constraints such as inventory and production set-up cost has been added to the model in Blumenfeld et al. [119].

ii. The IRP with Multiple Products

In the case study of Speranza and Ukovich [159], [160], they work on a multi-product flow for a single customer with deterministic frequencies. Following these studies, Bertazzi et al. [161] add multiple customers into model. Popovic et al. [162] study the multi-item IRP. In their model they deal with different types of fuel which have to be delivered to customers. Since the proposed MILP can only handle the smallest instance from a practical application, they solve the problem with variable neighborhood search (VNS) heuristic.

Moin et al. [163] study a multi-product version with multiple suppliers and one customer. By linear mathematical formulation they find upper and lower bounds for the problem. Mjirda et al. [164] improve their results by using a variable neighborhood search (VNS) heuristic. And then, Ramkumar et al. [165] analyze many-to-many case. They propose a MILP formulation for a multi-item multi-depot IRP. But, they cannot solve to optimality even the small instances due to time limit.

Multi-product formulation for a deterministic maritime problem is proposed by Ronen [166] but, it can only solve small instances. Coelho and Laporte [167] propose an exact MILP to solve muti-vehicle multi-product version of the problem. Their study also considers shared inventory capacity and shared vehicle capacity for products.

iii. The IRP with Direct Deliveries and Transshipment

Dealing with direct deliveries, Kleywegt et al. [168] and by Bertazzi [169] simplify the problem. Direct deliveries remove the routing dimension from it and more effective when economic order quantities for the customers are close to the vehicle capacities [133], [170].

Li et al. (2010) develop an analytic method for performance evaluation of this delivery strategy, whose effectiveness can be represented as a function of system parameters. In this context, some policies are proposed. Herer and Roundy [152] propose power-of-two policies, and Zhao et al. [171] propose a fixed partition policy combined with a tabu search heuristic, and for multi-product version Viswanathan and Mathur [172] propose a stationary nested joint replenishment policy.

Roundy [173] studies the case with multiple customers receiving direct deliveries at discrete times, and defines frequency based policies proven to be within 2% of the optimum in the worst case. In this model, inventory holding costs are linear, but there are fixed ordering and delivery costs.

Coelho et al. [147] introduce transshipments in the IRP framework. To decrease distribution cost, planned transshipment decisions should be added to the model.

As a mean of reducing stock outs when demand is more than inventory, Coelho et al. [174] propose transshipment within a DSIRP framework. Their results show that emergency transshipment is a valuable option to mitigate average stock outs while reducing distribution costs.

iv. The Consistent IRP

In some cases such as when very small deliveries take place on consecutive days, followed by a very large delivery, after which the customer is not visited for long period, cost-optimal solution may result in inconveniences for supplier and customers.

Christofides and Beasley [64], Beasley [175], Barlett and Ghoshal [176] or Zhong et al. [177] include workforce management within the periodic VRP for assigning territories to drivers. Smilowitz et al. [178] analyze potential trade-offs between workforce management and travel distance goals in a multi-objective PVRP.

By Coelho et al. [148], quality of service features are incorporated in IRP solutions. This is achieved by ensuring consistent solutions from three different aspects: quantities delivered, frequency of the deliveries and workforce management. Experiments on benchmark instances show that, ensuring consistent solutions over time increases the cost of the solution between 1% and 8% on average.

v. Stochastic Inventory-Routing

When the suppliers know customer demand just in a probabilistic sense, the problem is Stochastic Inventory-Routing Problem (SIRP). In the SIRP shortage can occur but to prevent that shortage there is a penalty cost and this penalty is usually modeled as a proportion of the unsatisfied demand. When there is no backlogging, unsatisfied demand is lost, but, the objective of SIRP remains the same.

In the SIRP the supplier, over a planning horizon, must decide on a distribution policy that maximizes its expected discounted value. Bard et al. [179], Federgruen and Zipkin [117] work with gas and oil industry as a case which mostly deals with SIRP.

For SIRP with the finite horizon, there are lots of heuristic algorithms. For a random demand environment and to accommodate inventory and shortage costs, Federgruen and Zipkin [117] develop the VRP heuristic of Fisher and Jaikumar [118]. And to consider multiple products, Federgruen et al. [180] extend their work. By considering the customers degree of urgency, Golden et al. [181] determine which customers to visit.

Using the rolling horizon framework of Bard et al. [179], Jaillet et al. [182] solve the problem for a short-term. The specific problem of them includes direct deliveries for emergency deliveries when customers run out of stock and satellite facilities where trucks can be replenished during their routes.

For an unknown demand varying within 10% of the mean value, Geiger and Sevaux [97] propose several polices base on delivery frequencies for the customers. They apply the record-to-record travel heuristic of Li et al. [183] to solve the problem for much more periods. To demonstrate the solution better, they use a pareto front approximation of the policies when moving from a total routing optimized solution to an inventory-optimized one.

Liu and Lee [184] solve the classical road-based IRP with time windows. While a combination of variable neighborhood search and tabu search is used in their algorithm, the algorithm is not indicated as so effective. Minkoff [185] suggest a heuristic approach based on a Markov decision model to a problem similar to the IRP, called the Delivery Dispatching Problem. First he makes the objective function simple with making it a sum of smaller and simpler objective functions, then, solve it heuristically.

Considering transportation and stock out costs and no inventory holding cost, Campbell et al. [144] introduce a dynamic programming model for the SIRP. Also, in their problem, supplier knows the inventory level at each of customers, the amount to deliver to each and how to combine them into routes, in the beginning of the periods.

Also, Berman and Larson [186] use dynamic programming to solve the case where the demand probability distributions are known. In their method, they adjust the amount of goods delivered to each customer, in order to minimize the expected sum of penalties.

The approach is followed by Kleywegt et al.[168], [187] who, as in Campbell et al. [144], use a Markov decision process to formulate the SIRP. Here, a set of customers must be served from a warehouse by means of a fleet of homogeneous capacitated vehicles. Each customer has an inventory capacity, and the problem is modeled in discrete time. Inventory at each customer at any given time is known to the supplier. Customer demands are stochastic and independent from each other, and the supplier knows the joint probability distribution of their demands, which does not change over time. The supplier must decide which customers to visit, how much to deliver to them, how to combine customers into routes, and which routes to assign to each vehicle. The set of admissible decisions is constrained by vehicle and customer capacities, driver working hours, possible time windows at the customers, and by any other constraint imposed by the system or the application. Although demands are stochastic, the cost of each decision is known to the supplier. Thus, Kleywegt et al. [168], [187] consider traveling costs, shortages which are proportional to the amount of unsatisfied and lost demand and holding costs. The problem is formulated so as to maximize the expected discounted value over an infinite horizon as a discrete time Markov decision process.

Kleywegt et al. [168] work on the cases with direct deliveries. However, Kleywegt et al. [187] allow deliveries to up to three customers per route. In the direct deliveries study of Kleywegt et al. [168] optimal solutions are obtained on instances with up to 60 customers and up to 16 vehicles, whereas in Kleywegt et al. [187] instances with up to 15 customers and five vehicles are solved. Adelman [188] prefers to limit maximal route duration and vehicle capacity instead of number of customers visited per route. He derives a linear program from a value function, and its optimal dual prices are used to calculate the optimal policy of the semi-Markov decision process.

Qu et al. [189] develop a periodic policy for a multi-item IRP as exceptions to the dynamic programming approach. Huang and Lin [190] solve it with an ant colony optimization algorithm. Hvattum and Løkketangen [191] and Hvattum et al. [192] solve the IRP with the stochastic information over a short horizon. They solve the problem using a GRASP by increasing the volume delivered to customers.

Another solution of SIRP is through the use of robust optimization. The structure of the solution is suitable to tackle with ambiguity in the circumstances information does not exist on the parameter probability distributions. Mini max solution (optimizing the problem and providing feasibility for all possible comprehensions of the bounded uncertain variables) support the solution. Aghezzaf [193] deals with the state of normally distributed customer demands and travel times with constant averages and bounded standard deviations. Robust optimization is used during his research to establish the distribution plan in the course of a non-linear mixed-integer programming formulation which is feasible for all possible realizations of the random variables. He uses Monte Carlo simulation for developing the plan's significant parameters (replenishment cycle times and safety stock levels). Solyalı et al. [194] works on such an accurate approach derived from robust optimization. The instances with up to seven periods and 30 customers within a reasonable computing time are settled by using the formulation.

2.4 Collection Problems

According to Taniguchi et al.[68], the processes for planning, optimizing and controlling logistics and transport activities is divided into forward and reverse logistics. In forward logistic the system is for the flow of goods from the producers to the consumers and in reverse logistic the flow from the consumers to facilities. The vast amount of waste in cities indicates the importance of this logistic activity which yields to large amounts of publication in the field of collection waste, mostly trying to reduce the total cost of this process.

The first paper on PVRP published by Beltrami and Bodin [69] is also one of the first papers studying on waste collection.

Golden et al. [70] classify waste management problems into residential collection, commercial collection, and roll-on-roll-off problems. In roll-on-roll-off problem, customers gather garbage in waste containers and after that they request for waste treatment services which by moving containers from specific customer locations to disposal places, they satisfy the service. And, in commercial collection problems, the containers from commercial places are being collected.

More specifically, the roll-on-roll-off VRP is discussed in Bodin et al. [71], where tractors move large trailers between locations and a disposal facility. The tractors can only move one trailer at a time. They propose a mathematical programming formulation, two lower bounds and four heuristic algorithms.

Kulcar [72] models a case study in Brussel for solid waste collection. To find convincing results, they apply their model in several types of transportation modes such as vehicle, rail and canal.

Within the case study of Chicago, Eisenstein and Iyer [73] publish their paper about scheduling of trucks to accumulate the garbage. Their new dynamic approach considers the different amounts of waste in different city blocks. In their model, using Markov decision process, they find more flexible results comparing to the previous system of gathering the garbage.

In the field of reverse logistic, Jayaraman et al. [74] model a MILP to design a network under a pull system. In their model the customer demands for recovered products.

While, in some parts of their study, they investigate the managerial use of the model for logistics decision-making with the aim of minimizing the total costs, their model for the location of distribution facilities, gives the exact results for transshipment, stocking and production. To consider the processing costs of returned products and inventory costs, Krikke et al. [75] model a MILP for the two stage reverse logistic network. The model involves the installment of remanufacturing processes in it. And, as their case study, they examine the data of a copier firm in Venlo- Océ.

Considering the Hanoi, Vietnam's case study, Tung and Pinnoi [76] research on a vehicle routing and scheduling problem to solve a waste collection problem. The main difference between their problem and others is that their problem had several steps; first by handcart the waste is picked up and delivered to gather sites, then there is the transshipment of waste from handcart to tipper. Also, there are some other problem specific differences, such as time table for the gathering sites and the time restrictions. To solve this problem, they use a construction heuristics to find the initial solution of the problem, and then they try to improve it by a specific improvement heuristic.

With respect to the vehicle's capacity, Mourao and Almeida [77] investigate a waste collection problem by solving a capacitated arc routing problem. In their study, after accumulating the waste and after reaching the vehicle's capacity, the vehicle delivers waste to specific facility and again gathers the remaining.

For the flexible visit frequency, which is not fixed in PVRP, Baptista et al. [78] research on collecting the recycling papers in the Almada municipality in Portugal. They solve this new problem using a heuristic based on method by Christofides and Beasley [48].

In a case that the PVRP has some intermediate facilities (PVRP-IF), Angelelli and Speranza [53] suggest a TS method and in Angelelli and Speranza [79] they extend their algorithm for measuring the operating cost of different waste-collection systems. They study PVRP with intermediate facilities (PVRP-IF). In their model, the capacity of a vehicle is renewed by visiting an intermediate facility.

Teixeira et al. [12] study PVRP with some other constraints. They implement a three-phase heuristic to deal with incorporate of the collection of different types of waste and a long planning period. In their study there are three types of waste which were plastic, glass and paper. In their assumption, they use geographical zones to distinguish the parts of the city, and after that they decide on types of waste collect that should be used in that zone and in their final step of heuristic the sites which the collect should be placed is decided and they solve the routing problem depending on previous decisions.

For the generalization of the roll-on-roll-off VRP, Archetti and Speranza [80] study a real life problem so-called 1-skip collection problem. In that problem, a fleet of vehicles must transport skips of waste, one at a time, from its location to one of different disposal sites, depending on the kind of waste contained in the skip.

On the other hand, Lacomme et al. [81] model a periodic capacitated arc routing problem on a mixed graph. In their model the demand of an arc depends on the period or on the date of the previous visit.

Prins and Bouchenoua [82] model the node, edge and arc routing problem (NEARP), that generalizes the VRP and the CARP. The NEARP deals with mixed graphs. The model requires nodes, arcs and edges. They propose a memetic algorithm and show its competitiveness on standard benchmark instances of the VRP and the CARP.

One way to deal with uncertainty is to do a single or multi-parameter sensitivity analysis. This approach is extended by using scenarios for the input parameters and obtaining the individual solution that performs best over the set of scenarios. In this regard, Listes and Dekker [83] propose a stochastic mixed integer programming model in a sand recycling network. The model presented for locating reverse logistic facilities differ in structure hardly from the traditional location models with the aim of maximizing the total profit. The results help to decide better for a reverse logistic under uncertainty.

Üster et al. [84] consider a multi-product closed-loop supply chain network design problem where they locate collection centers and remanufacturing facilities while coordinating the forward and reverse flows in the network so as to minimize the processing, transportation, and fixed location costs. The problem of interest is motivated by the practice of an original equipment manufacturer in the automotive industry that provides service parts for vehicle maintenance and repair. They design a semi-integrated network in which the direct logistics network exists and only collection and recovery centers must be located. The model optimizes the direct and reverse flows simultaneously. An exact method is developed based on the Benders decomposition technique.

Lu and Bostel [85] present a two-level location problem with three types of facility to be located in a specific reverse logistics system, named a Remanufacturing Network (RMN). They propose a 0-1 mixed integer-programming model that considers the forward and reverse flows and their interactions, simultaneously. To solve the model, they develop an algorithm based on the Lagrangian heuristics.

Wojanowski et al. [86] study the interplay between the industrial firms and government concerning the collection of used products from households. They present a continuous modeling framework for designing a drop-off facility network and determining the sales price to maximize the firm's profit under a specific deposit-refund. They show that a minimum deposit–refund requirement cannot achieve high collection rates for products with low return value and point out two complementary policy tools that can be used by the government.

Crevier et al. [87] address an extension of the multi-depot vehicle routing problem in which vehicles may be replenished at intermediate depots along their route. The study proposes a heuristic combining the adaptive memory principle, a tabu search method for the solution of sub-problems, and integer programming. Getting the initial results by this method, Tarantilis et al. [88] propose an algorithm to improve the results. The new algorithm is a three-step algorithmic framework for solving the vehicle-routing problem with intermediate replenishment facilities. The algorithm is consist of variable neighborhood search, tabu search which is used as a local search in VNS and the guided local search (GLS) which is used as post-optimization.

Du and Evans [89] analyze the reverse logistic networks that deal with the returns requiring repair service. A problem involving a manufacturer outsourcing to a third-party logistics provider for its post-sale services is defined. They define an advanced bi-objective MILP model. The objectives of the model include the minimization of the tardiness and the total costs. In order to solve the model, a hybrid-scatter search method is developed.

Aras et al. [90] demonstrate the problem of locating collection centers of a company with the aim of collecting used products from product holders. They develop a nonlinear model for determining the locations of collection centers in a simple reverse logistics network. They assume that a pick-up strategy is in place according to which vehicles with limited capacity are dispatched from the collection centers to the locations of product holders to transport the returns. The important point regarding their article is the capability of the presented model in determining the optimal buying price of used products with the objective of maximizing the total profit. Based on tabu search, they develop a heuristic approach to solve the model.

Pishvaee et al. [91] propose a mixed integer linear programming model to minimize the transportation and fixed opening costs in a multistage reverse logistics network. They also apply a simulated annealing (SA) algorithm with special neighborhood search mechanisms to their problem.

Hemmelmayr et al. [92] consider a real world waste collection problem in which glass, metal, plastics, or paper is brought to certain waste collection points by the citizens of a certain region. The collected materials are delivered to intermediate facilities (IF). The problem considers a planning horizon of several days. They develop a set of benchmark instances and propose a method that is hybrid of variable neighborhood search and dynamic programming. They manage to outperform previous metaheuristics.

Buhrkal et al. [93] demonstrate how to collect waste in an efficient way. They study the Waste Collection Vehicle Routing Problem with Time Window which is concerned with finding cost optimal routes for garbage trucks such that all garbage bins are emptied and the waste is driven to disposal sites while respecting customer time windows and ensuring that drivers are given the breaks that the law requires. They propose an adaptive large neighborhood search algorithm for solving the problem.

Chapter 3

A SELECTIVE AND PERIODIC INVENTORY ROUTING PROBLEM FOR WASTE VEGETABLE OIL COLLECTION

In this chapter, we explain the selective and periodic inventory routing problem for waste vegetable oil collection and present the exact model proposed by Aksen et al. [1].

The problem is defined on a complete directed graph with a set of n source nodes and a depot. The real road shortest path distances d_{ij} are defined for each arc (i, j) in the graph. The problem has a cyclic planning horizon over a period of seven days. The source nodes represent waste oil accumulation points and each source node i accumulates waste oil deterministically in a rate of a_{it} in each period t. In each period, several source nodes can be visited with a fleet of vehicles having a fixed capacity of Q, leaving the depot in order to collect the waste oil accumulated at the source nodes and return to depot. The depot represents the biodiesel production facility. When a source node is visited at a period, the total oil accumulated till then has to be collected, meaning partial collection is not an option.

The production facility has to have as much as the required oil r_t for the period t in order to produce enough biodiesel to meet its demands. The required amount of oil can be obtained by visiting the source nodes for waste oil, purchasing virgin oil, using on hand oil inventory obtained in the previous periods, or any combination of them.

A traveling cost c per unit distance traveled, a purchasing cost p per liter of virgin vegetable oil, a holding cost h per liter of waste oil per period, and a vehicle operating cost v per vehicle per period is used to calculate overall cost of the facility as the objective function.

The Selective and Periodic Inventory Routing Problem (SPIRP) searches to find a periodic collection schedule that repeats itself in every cycle. This schedule identifies the set of source nodes to be visited and the associated vehicle routes in each period. The objective is to minimize the sum of total travel cost, vehicle operating cost, inventory holding cost and purchasing cost while satisfying the production requirements and vehicle capacity constraints

The SPIRP is defined as an *NP*-hard problem since it generalizes several well-known *NP*-hard problems related to routing and lot-sizing. When the planning horizon is a single period, the problem reduces to a variant of VRP in which customer visits are selective and the facility meets its requirement through collection and/or purchasing. In the case of a multi period planning horizon with only a single source node, the problem reduces to a variant of the capacitated lot-sizing problem since the main decision is on which days to visit the customer, while considering the trade-offs among the transportation cost (which is a step function due to vehicle costs), the inventory holding cost, and the purchasing cost (which is analogous to the shortage cost). [1]

Aksen et al. [1] define an exact solution method for SPIRP. Their method has two main components; a visiting schedule that reveals which nodes are visited, and a set of vehicle routes for each period of the planning cycle. The MILP formulation they propose determines the visiting schedule using binary variables which determines the source nodes to be visited in each period. Moreover, integer variables are used to record the collected amounts and maintain the amount of accumulated waste oil at the source nodes according to the visiting schedule. The requirements at the production facility are managed by inventory balance constraints. For the vehicle routing decisions, the MILP model uses a single commodity flow formulation to ensure connectivity and sub tour elimination. Their method defines continuous variables to represent the flow of the commodities along the arcs traveled by the vehicles and incorporate the binary node selection variables into the flow balance constraints.

3.1 MILP model

The index sets, parameters, and decision variables of the model are defined as below [1].

Index Sets

$I = 0, 1, \dots, n$: the set of n source nodes and the depot 0,

$IC = 1, \dots, n$: the set of n source nodes only (a subset of I),

$T = 1, \dots, \tau$: the set of τ periods in the cyclic planning horizon.

Parameters

c : traveling cost per unit distance.

d_{ij} : distance from node i to node j, $(i, j \in I, d_{ij} \neq d_{ji})$.

a_{it} : waste vegetable oil accumulation amount in period t at node i, $(i \in IC, t \in T)$.

r_t : waste oil requirement of the company per period, $t \in T$.

h : inventory holding cost per period for storing one liter oil at the depot.

v : operating cost per vehicle.

c : virgin vegetable oil purchasing price per liter.

Q : vehicle capacity in liters.

A_i : total weekly accumulation of waste oil at node i, $(i \in IC)$. This number serves as the Big-M number in the model. It is calculated by the formula $A_i = \sum_{t \in T} a_{it}$.

Decision variables

X_{ijt} : binary variable indicating if arc (i, j) is traversed by a vehicle in period t, $(i, j \in I, t \in T)$.

Y_{it} : binary variable indicating if node i has been visited in period t, $(i \in IC, t \in T)$.

Z_i : binary variable indicating if node i has been visited at least once during a cycle $(i \in IC)$. It becomes 0, if node i is not visited at all.

F_{ijt} : the amount of waste oil flow from node i to node j in period t, $(i, j \in I, t \in T)$.

W_{it} : the amount of waste oil collected from node i in period t, $(i \in IC, t \in T)$.

I_{it} : ending inventory of waste oil by the end of period t at node i, $(i \in I, t \in T)$.

I_{i0} : initial inventory of waste oil at the beginning of the cycle at node i, $(i \in I)$.

S_t : the amount of waste oil purchased by the collecting company in period t, $(t \in T)$.

MILP formulation

$$\min TC = c\sum_{i\in \mathrm{I}}\sum_{j\in I,(j\neq i)}\sum_{t\in T} d_{ij}X_{ijt} + v\sum_{i\in \mathrm{IC}}\sum_{t\in T} X_{0it} + h\sum_{t\in T} I_{0t} + p\sum_{t\in T} S_t \tag{3.1}$$

$$\sum_{j\in I,(j\neq i)} F_{ijt} - \sum_{j\in I,(i\neq j)} F_{jit} = W_{it}, \forall i \in \mathrm{IC}, \forall t \in T \tag{3.2}$$

$$F_{ijt} \leq (Q - a_{jt})X_{ijt,} \forall i \in \mathrm{I}, \forall j \in \mathrm{I}, \forall t \in T, i \neq j \tag{3.3}$$

$$F_{ijt} \leq Q - W_{jt}, \forall i \in \mathrm{I}, \forall j \in \mathrm{IC}, \forall t \in T, i \neq j \tag{3.4}$$

$$F_{ijt} \geq W_{it} - A_i(1 - X_{ijt,}), \forall i \in \mathrm{IC}, \forall j \in \mathrm{I}, \forall t \in T, i \neq j \tag{3.5}$$

$$\sum_{j\in I,(j\neq i)} X_{jit} = Y_{it}, \forall i \in \mathrm{IC}, \forall t \in T \tag{3.6}$$

$$\sum_{j\in I,(j\neq i)} X_{ijt} = Y_{it}, \forall i \in \mathrm{IC}, \forall t \in T \tag{3.7}$$

$$\sum_{i\in \mathrm{IC}} X_{i0t} = \sum_{i\in \mathrm{IC}} X_{0it}, \forall t \in T \tag{3.8}$$

$$Q\sum_{i\in \mathrm{IC}} X_{0it} \geq \sum_{i\in \mathrm{IC}} W_{it}, \forall t \in T \tag{3.9}$$

$$W_{it} \leq A_i Y_{it}, \forall i \in \mathrm{IC}, \forall t \in T \tag{3.10}$$

$$I_{it} \leq A_i(1 - Y_{it,}), \forall i \in \mathrm{IC}, \forall t \in T \tag{3.11}$$

$$I_{it} = I_{it-1} + a_{it}Z_i - W_{it}, \forall i \in \mathrm{IC}, \forall t \in T \tag{3.12}$$

$$I_{i0} = I_{iT}, \forall i \in \mathrm{I} \tag{3.13}$$

$$I_{0t} = I_{0t-1} + \sum_{i\in \mathrm{IC}} W_{it} + S_t - r_t, \forall t \in \mathrm{T} \tag{3.14}$$

$$Z_i \leq \sum_{t\in T} Y_{it} = , \forall i \in \mathrm{IC} \tag{3.15}$$

$$Z_i \geq Y_{it}, \forall i \in \mathrm{IC}, \forall t \in \mathrm{T} \tag{3.16}$$

$$X_{ijt} + X_{jit} \leq Y_{it}, \forall i \in \mathrm{IC}, \forall j \in \mathrm{IC}, \forall t \in T, i \neq j \tag{3.17}$$

$$X_{i0t} \leq Y_{it}, \forall i \in \mathrm{IC}, \forall t \in T \tag{3.18}$$

$$X_{0it} \leq Y_{it}, \forall i \in \mathrm{IC}, \forall t \in T \tag{3.19}$$

$$X_{ijt} \in \{0, 1\}, \forall i \in \mathrm{I}, \forall j \in \mathrm{I}, \forall t \in T, i \neq j \tag{3.20}$$

$$Y_{it} \in \{0, 1\}, \forall i \in \mathrm{IC}, \forall t \in T \tag{3.21}$$

$$Z_i \in \{0, 1\}, \forall i \in \mathrm{IC} \tag{3.22}$$

$$F_{ijt} \geq 0, \forall i \in \mathrm{I}, \forall j \in \mathrm{I}, \forall t \in T, i \neq j \tag{3.23}$$

$$W_{it} \geq 0, \forall i \in \mathrm{IC}, \forall t \in T \tag{3.24}$$

$$I_{it} \geq 0, \forall i \in \mathrm{I}, \forall t \in T \tag{3.25}$$

$$I_{i0} \geq 0, \forall i \in \mathrm{I} \tag{3.26}$$

$$S_t \geq 0, \forall t \in \mathrm{T} \tag{3.27}$$

The objective function of the model, the total cost is the sum of four different cost functions; transportation costs, vehicle operating costs, inventory holding costs, and purchasing costs incurred by the collection company in the planning horizon.

Constraints (3.2) are used to balance the flow at each source node i. Constraints (3.3) and (3.4) control the upper bounds on the flow variables F_{ijt} by considering the vehicle capacity and the amount of waste oil collected from node j when a vehicle travels from node i to node j in period t. Constraints (3.5) control the lower bounds on flow variables. They make sure that if a vehicle travels from i to j in period t, all accumulated oil at node i is collected. Constraints (3.6) and (3.7) are incoming and outgoing degree balance constraints for each source node i, ensuring that the incoming/outgoing degree of node i is equal to 1, if node i is visited in period t; and equal to 0, otherwise. (3.6) and (3.7) relate the binary variables Y_{it} to X_{ijt}. Constraints (3.8) are used as the degree balance constraint for the depot, imposing the incoming and outgoing degrees to be the equal.

Constraints (3.9) are used to ensure that the number of vehicles dispatched in a period is sufficient to carry the collected amount with respect to the total vehicle capacity. Constraints (3.10) make sure that the collection amount at node i in period t is 0 if it is not visited in that period. Constraints (3.11)-(3.16) are used to calculate the inventory at the source nodes and the depot. Constraints (3.11) are used to prevent partial collection of waste oil at a source node i, by making sure that the inventory at node i is zero at the end of period t, if it is visited in that period. Constraints (3.12) control the ending inventory at a source node i in period t and relates the integer variables; the daily accumulated waste oil at node i in period t, a_{it}, and the amount of waste collected from node i in period t, W_{it}. If node i is not visited in period t, then W_{it} will be 0 and the inventory will be increased by a_{it}. However, when node i is not visited at all, meaning Z_i = 0, its inventory amount remains constant. Constraints (3.13) make sure that the beginning and ending inventories of the planning horizon is equal for each source node i. (3.14) are the inventory balance constraints for the depot that considering the total collected amount, the purchased amount, and the required amount of the period t. Constraints (3.15) and (3.16) relate the binary decision variables Z_i to Y_{it} so that if node i is visited in any period, then Z_i becomes 1; and 0, otherwise. To tighten the model, they include the sub tour elimination constraints (3.17) to break sub tours of size two, and constraints (3.18) and (3.19) to avoid visits to a node which is not in the schedule.

3.2 Partial Relaxation

Aksen et al. [1] also introduce a partial relaxation of the MILP model to generate stronger lower bounds for SPIRP. They first get rid of the binary constraints (3.20) and add the following additional constraints. The integer variable V_t represents the number of vehicles dispatched in period t.

$$X_{ijt} \leq 0, \forall i \in \mathrm{I}, \forall j \in \mathrm{I}, \forall t \in T, i \neq j \tag{3.28}$$

$$V_t = \sum_{j \in IC} X_{0jt} \ , \forall t \in T \tag{3.29}$$

$$V_t \epsilon \ Z^+, \forall t \in T \tag{3.30}$$

Therefore, they apply linear relaxation on the binary sequencing variables X_{ijt}, but using equations (3.29) and (3.30), they strengthen the model by imposing integrality on V_t variables which are actually the sum of X_{0jt} variables over j.

Furthermore, instead of using equations (3.17) - (3.19) as tightening constraints, the following equations (3.31) and (3.32) are used.

$$X_{ijt} \leq Y_{it}, \forall i \in IC, \forall j \in I, \forall t \in T, i \neq j \tag{3.31}$$

$$X_{jit} \leq Y_{it}, \forall i \in IC, \forall j \in I, \forall t \in T, i \neq j \tag{3.32}$$

They state this partial linear relaxation (PLR) model provides quite strong lower bounds when solved within a time limit of one hour for the problem instances of 25.

3.3 Relaxation without Routing

We propose a new relaxation of the MILP model to provide easy-to-compute lower bounds for especially larger instances. We eliminate F_{ijt} flow variables, X_{ijt} binary variables; and related constraints (3.2)-(3.8), (3.17)-(3.19), (3.20) and (3.23), meaning eliminating the routing part of the problem. The number of vehicles dispatched is controlled by a new integer variable, namely V_t for each period by changing constraint (3.9) with (3.33) and (3.34).

$$Q \ V_t \ \geq \sum_{i \in \mathrm{IC}} W_{it} \, , \forall t \in T \tag{3.33}$$

$$V_t \geq 0, \forall t \in \mathrm{T} \tag{3.34}$$

We also change the objective function to replace the routing costs, namely transportation and vehicle operating cost, with new cost functions giving lower bound values for previous functions. We define minimum distance parameters $Min(d_{0i})$ and $Min(d_{i0})$ representing the distances between the depot and the closest customers in each direction. The vehicle operation cost and transportation cost parts of the objective function is modified as follow:

$$\min TC = c\ (Min(d_{0i}) + Min(d_{i0})) \sum_{t \in T} V_t + v \sum_{t \in T} V_t + h \sum_{t \in T} I_{0t} + p \sum_{t \in T} S_t \qquad (3.35)$$

The relaxation without routing (RR) model is computed easily in seconds; however the lower bounds are not very strong. Nevertheless, RR provides better bounds than PLR for some instances. We will discuss their performances further in chapter 5.

Chapter 4

AN ADAPTIVE LARGE NEIGHBORHOOD SEARCH ALGORITHM FOR SPIRP

The recent heuristic studies in the literature based on Large Neighborhood Search (LNS) presented a noticeable success in the fields of transportation and scheduling. The success of the method comes mostly from its ability to search a more complicated neighborhood with respect to previous heuristic methods used in these fields. Searching in a larger neighborhood increases the chance of finding better solutions, which also helps to find better objective values at the end. [195]

Especially when the problem has tighter constraints, a small neighborhood can fail to search throughout the solution space by getting stuck in a smaller search space. However, the large neighborhood can search in a larger space, which is actually the idea behind LNS. [195]

LNS is in the class of Very Large Scale Neighborhood search (VLSN) heuristic algorithms according to Ahuja et al. [197]. This class of heuristic methods uses the power of large neighborhoods to find high quality local optima, which can lead to better results by searching a larger neighborhood. The disadvantage of VLSN algorithms is as the neighborhood gets larger, the time the algorithm requires gets longer. To deal with the time problem, the neighborhoods used in VLNS algorithm are limited to a part of the original search space.

Large Neighborhood Search was first introduced by Shaw [196] in 1998. The article uses LNS to solve VRP, one of the most studied routing problems. The LNS heuristic requires a previous step to create an initial solution. Then, the algorithm applies destroy and repair methods to improve the objective value. To limit the search space of neighborhoods, Shaw [196] starts with a smaller neighborhood and gradually increases the degree of the neighborhood, which creates a larger neighborhood. On the other hand, Pisinger et al. [198] applies a random selection on the degree of the neighborhood in each iteration.

Pisinger et al. [198] modify LNS by defining several destroy and repair methods to be used throughout the heuristic, whereas LNS uses only one destroy method and one repair method. They define this new heuristic as Adaptive Large Neighborhood Search (ALNS). When compared to LNS, ALNS can make bigger changes to the current solution by exploring a larger search space with several destroy and repair methods. Moreover, ALNS dynamically controls the probability of using a neighborhood according to its performance throughout the search [195].

Pisinger et al. [198] state that the LNS methods have been very successful within the areas of routing and scheduling problems. Since our focus of study is on a routing and scheduling problem, namely SPIRP, we apply an ALNS algorithm on SPIRP. Our algorithm adopts the algorithm proposed by Coelho et al. [199], an ALNS heuristic on Inventory Routing Problem with Transshipments (IRPT) to SPIRP. This ALNS algorithm differs from the previous ones in terms of the applications of destroy and repair moves. Coelho et al. [199] do not follow the rule of applying a repair move and a destroy move each iteration. For some iterations, they just apply destroy or just repair moves. In this study, we use the moves they described; however after each move we apply a repair step to adopt several variables of different cost functions in our objective. A change in the variables of a cost function creates a need to adopt other variables. This need of modifying different groups of cost function comes from the characteristics of our problem such as its selective structure, the flexibility on the number of vehicles used, not allowing the transshipment between source nodes and using no inventory policies at the source nodes. The method Coelho et al. [199] describes can provide high quality solutions within reasonable computing times on large set of instances comparing to the exact model they presented.

The detailed structure of the ALNS algorithm is explained next.

4.1 Main Structure of the Algorithm

The ALNS algorithm is described in four main components [199]:

Large Neighborhood:

The neighborhoods used are designed to make a number of changes to the current solution. In our algorithm, moves can modify the source node selection, visits to a source node in the periodic schedule, number of vehicles used in a period or in several periods, amount of inventory, virgin oil purchase amount and the sequence of source nodes visited in a route. These changes may require recalculating all the optimization function elements; inventory amounts, transportation cost, purchasing amounts and vehicle operating cost after each move.

Adaptive Search Engine:

In order to decide which neighborhood to use in each iteration, a roulette-wheel mechanism is used. Each neighborhood has a weight which represents its share at the wheel. The weights of the neighborhoods are determined according to past performances of the neighborhood. Let w_i be the weight assigned to neighborhood i depending on its past performance. Then the neighborhood is chosen with probability $w_i / \sum_{k=1}^{\#\,neig.} w_k$ (# neigh = number of neighborhoods defined for ALNS).

Adaptive Weight Adjustment:

At the beginning, each neighborhood has the same probability of being chosen. During the run, the weights are updated several times. The run is divided into segments containing the same number of iterations, namely φ. At the end of each segment, the weights are updated depending on their performances during the current segment. The performances are recorded by keeping scores of neighborhoods, represented here by π_i for neighborhood i. At the beginning of each segment, scores are set to 0. After each iteration, the score of the neighborhood used at that iteration is updated as below, where $\sigma_1, \sigma_2, \sigma_3$ are integer numbers satisfying $\sigma_1 > \sigma_2 > \sigma_3$; s' is the new solution, s_{best} is the best solution and s be the current solution; and $f(s)$ is the objective function to be minimized.

$\boldsymbol{if}\ f(s') < f(s_{best})\ \boldsymbol{then}$

$\quad \pi_i = \pi_i + \sigma_1$

$\boldsymbol{else\ if}\ f(s') < f(s)\ \boldsymbol{then}$

$\quad \pi_i = \pi_i + \sigma_2$

$\boldsymbol{else\ if}$ s' is accepted by the simulated annealing criterion $\boldsymbol{then}$

$\quad \pi_i = \pi_i + \sigma_3$

$\boldsymbol{else}$

$\quad \pi_i = \pi_i$

The main idea here is that the better the new solution is, the larger is the increment on the score of the chosen neighborhood. At the end of each segment, the weights are updated as follows:

$$w_{i,j+1} = \begin{cases} w_{ij}, & if\ o_{ij} = 0 \\ (1 - \eta)w_{ij} + \eta\, \pi_i / o_{ij}, & if\ o_{ij} \neq 0 \end{cases}$$

Here, o_{ij} is the number of times neighborhood i is used in the current segment j, w_{ij} is the weight of neighborhood i in the current segment j and $\eta \in [0,\ 1]$ is the reaction factor controlling how much the last segment affects the current weights.

Acceptance and stopping criteria:

The acceptance criterion used is as in simulated annealing. Let p be a random variable with uniform distribution taking values between 0 and 1, and τ be the temperature parameter. The temperature is started at τ_{start} and it is decreased by a cooling factor ϕ at each iteration, where $0 < \phi < 1$.

$\boldsymbol{if}\ f(s') < f(s)$

$\quad s \leftarrow s';$

$\boldsymbol{else\ if}\ p < e^{-(f(s') - f(s))/\tau}$

$\quad s \leftarrow s';$

$\boldsymbol{else}$

$\quad s \leftarrow s;$

Moreover, a time limit of one hour is used as stopping criterion in order to avoid long CPU times.

4.2 Applying ALNS

Since SPIRP includes several management decisions in one problem, the problem itself is highly complicated. Therefore, the configuration of the algorithm is a complex one as well. The data structure we create for this problem includes following elements:

- There is a distance matrix with size $(n + 1) * (n + 1)$ that keeps real life shortest path distances between source nodes and the depot. (n is the number of source nodes as defined in Chapter 3).
- Every source node has an accumulation array in the length of the planning horizon, say k. Each element in the array corresponds to the accumulation rate of that source node for a specific period.
- Every source node has a binary schedule array of size k which has the same length with the accumulation array. Elements with the value of 1 means that the source node is visited in those periods. If an element is 0, then the source node is not visited in that period.
- For each source node, a collection array keeps the accumulated amounts with respect to schedule and accumulation arrays. These are the values corresponding to the collected amounts from source nodes.
- Each period has a list of source nodes that are visited in that period.
- Each period has a list of routes such that each element corresponding to a vehicle containing the list of source nodes visited by that vehicle.
- For the depot, an inventory array of length k keeps the amount of inventory at the depot at the beginning of each period.

The algorithm starts with known distance matrix and accumulation array. The initial solution provides the schedule array for each source node. Using these data, the collection array for each source node, inventory array for the depot and total purchase amount are calculated. Source nodes list for each period is created. Then, Clark and Wright parallel savings algorithm [17] is applied for each period to create the route list. As the part of the initial solution, several improvement heuristics are applied that is explained in the next section.

Every iteration, a move is chosen according to the roulette-wheel mechanism as explained in Section (4.1). After each move is applied, relatively more time is spent to restructure the solution as follows:

- The change starts with updating the schedule array of related source nodes and the source nodes list of necessary periods.
- The collection array is reformed.
- If needed, the inventory array and the purchase amount are also updated.
- Moreover, at each iteration, if a need for an extra vehicle or an opportunity to decrease the number of vehicles in a period arises, the parallel savings algorithm is solved from scratch and improvement heuristics are applied as in the initial solution.

The weights of all moves are started with equal values, i.e. $1/11$. The segment length φ in our algorithm is 200. The score updates $\sigma_1, \sigma_2, \sigma_3$ take values 10, 5, and 2, respectively. The reaction factor η is 0.7. The starting temperature τ_{start} is 100,000 and the cooling factor ϕ is 0.99977 which corresponds to 50,000 iterations. For larger instances with over 50 source nodes, the cooling factor becomes 0.9996, corresponding to 100,000 iterations. For every parameter, several sensitivity analyses are performed and these values are found to be more effective.

4.3 Initial Solution

As explained before, the algorithms make gradual improvements over an initial solution. We observe that a relatively better initial solution gives better results with respect to a random initial solution. In order to find an initial solution, we use the solution of the relaxation without routing (RR) model. The model gives us a visiting schedule for each source node during the planning horizon. The schedule of the solution is used for the scheduling part of the problem and for the routing part, the parallel savings algorithm is used to determine the routes for each period in the planning horizon. On top of that, to find a better initial solution, below improvement heuristics are applied. These heuristics are proven to perform well for VRP in the literature [21]:

Intra-route 2-Opt

Two edges are removed from the tour and the two remaining segments are reconnected

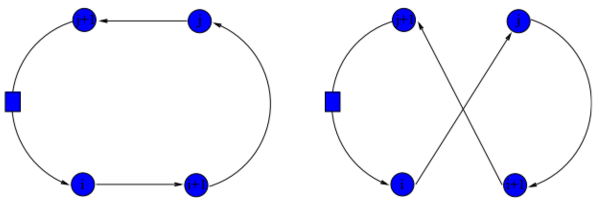

Figure 4. 1: Intra-route 2-Opt

Intra-route 3-Opt

Three edges are removed from the tour and the three remaining segments are reconnected in all possible ways

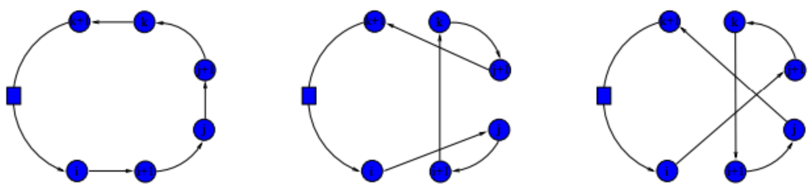

Figure 4. 2: Intra-route 3-Opt

Inter-route 2-Opt

Two edges from different routes are replaced by two new edges

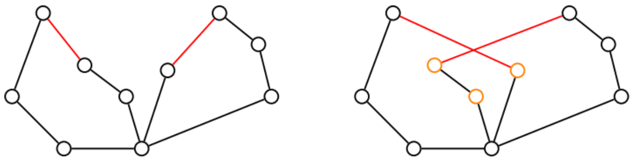

Figure 4. 3: Inter-route 2-Opt

Inter-route customer move

A customer is moved from one route to another.

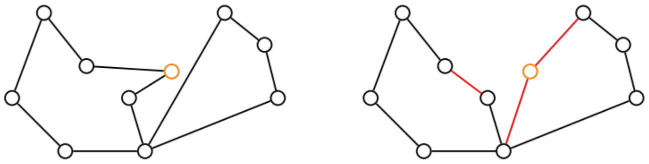

Figure 4. 4: Inter-route customer move

2 routes customer exchange

Two strings of at most k vertices are exchanged between two routes.

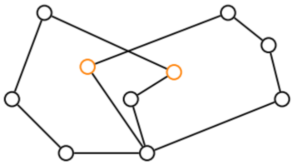

Figure 4. 5: 2 routes customer exchange

3 routes cycle customer exchange

Three routes are considered and three customers from each route are shifted to the next route of the cyclic permutation.

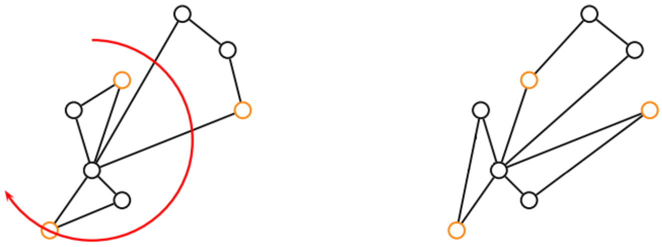

Figure 4. 6: 3 routes cycle customer exchange

4.4 Moves of the Algorithm

Because of the option to purchase in SPIRP, the problem does not have an infeasibility problem. Therefore, we are free to apply any kind of move in our algorithm. However, because of the complexity of the problem, even a small move can create considerable changes. Periodic structure of the problem requires changes in more than a single period for most moves. Therefore, even with the simple moves, the time to restructure the solution is significant. For relatively more complicated moves, the case is even more time consuming. For instance; new routes are constructed from scratch for several periods.

The moves we use in this study are described in [199] for IRPT by Coelho et al. We adopt for SPIRP. After each move, we apply a repair step to reform the data structure of the new solution and recalculate the optimization functions with their related variables. Several moves are applied ρ times at a single iteration, where ρ is an integer randomly chosen between 1 and 3 with semi-triangular distribution with negative slope. The probability of 1 to be chosen is $5/9$, 2 is $3/9$ and 3 is $1/9$. Since the probability of ρ being 1 is relatively high, in most iterations, the moves are applied only once, as in the classical approach in metaheuristics. However, with relatively smaller probabilities, ρ can be 2 or 3 which transforms a move into a larger move that can search a larger solution space.

Each move creates changes on the schedule of either one source node or several of them. The source nodes that moves are applied on are chosen among a subset of source nodes randomly or according to a predefined rule. Considering the characteristics of the moves, we define subsets of the source nodes as such:

- Subset 1: Source nodes that are not in the current solution, meaning the source nodes that are not visited at all in the current solution.
- Subset 2: Source nodes that are visited once in the current solution.
- Subset 3: Source nodes that are visited more than once, but less than the planning period k in the current solution.
- Subset 4: Source nodes that are visited in all periods in the planning horizon.

Even though a move modifies only the source list of a single period, updating the routing list, inventory amounts and purchase amounts of several periods might be required. Before defining the moves, we explain the types of updating procedure the data structure goes through after a move:

4.4.1 Repair Schemes

Repair 1

It is used in the case a move removes a source node that is in Subset 3 or 4 from a period. The source node still stays in the solution at another period. Hence, the collected amount in the removed period is transferred to the next visiting period. Thus, to update the next visiting period also becomes necessary. For the removed period, the source node list, and routing list are updated. We check for the possibility of reducing the number of vehicles for this period since the total collected amount is decreased with the removal of a source node. To do this, we check whether the total collected amount for this period is lower than the total capacity of remaining vehicles if one of them is removed. If this inequality holds, removing one vehicle might be possible. Then, we solve the parallel savings algorithm from scratch and apply improvement heuristics for this period.

For the next period, we first need to check whether the vehicle has enough capacity to carry the additional collection amount for the source node. If there is enough slack capacity, we only update the total collected amount and slack capacity of the vehicle for the next visiting period. In case the space is not enough, we check whether there is a vehicle with enough slack capacity to carry that source node in that period. If there is, we insert the node with the cheapest insertion rule. If there is not enough space, we create a new vehicle that only carries that node. In this case we check for the possibility of reducing the number of vehicles for this period as we describe for removed period. If removing one vehicle is probable, we solve the parallel savings algorithm from scratch and apply improvement heuristics to this period as well.

Total collected amount for the planning horizon stays the same but the collected amounts for two periods are changed. The total purchased amount does not change and so purchasing cost stays same. However, with different collection amounts for the periods, the inventory amounts for each period and so the inventory cost change. Furthermore, since some routes are changed, transportation cost changes. Lastly, vehicle operation cost might change if number of vehicles used differs.

Repair 2

It is in the case a move inserts a source node that is in Subset 2 or 3 into a period. The source node was already in the solution for at least one other period before insertion. Hence, the new collected amount in the inserted period is actually transferred from the next visiting period. Thus, to update the next visiting period also becomes necessary. For the inserted period, we first check whether there is a vehicle with enough slack capacity to carry the inserted source node. If there is, we insert the node with the cheapest insertion rule into that vehicle. If there is not enough space in any vehicle, we create a new vehicle that only carries that node. In this case we check for the possibility of reducing the number of vehicles for this period as described in Update 1. If removing one vehicle is probable, we solve the parallel savings algorithm from scratch and apply improvement heuristics to this period as well.

For the next period, the collected amount of the chosen node is decreased as much as the transferred amount to the new visiting period of the source node. We decrease the total collected amount of the period and increase the slack capacity of the vehicle that the node is in. We check for the possibility of reducing the number of vehicles as described in Update 1, since the total collected amount is decreased. To do this, we solve the parallel savings algorithm from scratch and apply improvement heuristics for this period as well.

Total collected amount for the planning horizon stays the same but the collected amounts for two periods are changed. The total purchase amount does not change and so purchasing cost stays the same. However, with different collection amounts for the periods, the inventory amounts for each period and thus the inventory cost change. Furthermore, since some routes are changed, transportation cost changes. Lastly, vehicle operation cost might change if the number of vehicles used differs.

Repair 3

It is in the case a move deletes a source node that is in Subset 2, 3 or 4 from all periods. For the periods the node is removed, the source node list, and routing list are updated. Then, we check for the possibility of reducing the number of vehicles for each period as described in Update 1. Then, we solve the parallel savings algorithm from scratch and apply improvement heuristics for the periods removing a vehicle is probable.

Since the node is removed from visiting schedule, the facility might fall short to meet the requirements if there is not enough waste oil is collected. In this case, the company purchases virgin oil as required. Hence, the purchasing cost might increase. Inventory amounts decrease and thus the inventory cost decreases. Furthermore, since the node is removed from some routes, transportation cost decreases. Lastly, vehicle operation cost might change if number of vehicles used differs.

Repair 4

It is in the case a move inserts a source node that is in Subset 1 into a random period. For the random period, we first check whether there is a vehicle with enough slack capacity to carry the inserted source node. If there is, we insert the node with the cheapest insertion rule into that vehicle. If there is not enough space in any vehicle, we create a new vehicle that only carries that node. In this case we check for the possibility of reducing the number of vehicles for this period as described in Update 1. If removing one vehicle is probable, we solve the parallel savings algorithm from scratch and apply improvement heuristics to this period as well.

If the company has been purchasing virgin oil before the insertion, the inserted amount is subtracted from the purchased amount without allowing negative purchase. Hence, the purchasing cost might decrease. Inventory amount increases and thus the inventory cost increases. Furthermore, since the node is inserted into a route, transportation cost increases. Lastly, vehicle operation cost might change if number of vehicles used differs.

Repair 5

It is in the case a move exchanges all the source nodes of a period with the source nodes of another period. For two periods, all the routes are destroyed. Because several source nodes' collection values change, it is highly possible all of the periods are affected by this exchange. We first update the source node lists of the exchanged periods and the schedule arrays of the exchanged source nodes. Then, as explained in Section (4.2), we restructure these two periods from scratch with their new nodes. Furthermore, we solve the parallel savings algorithm and apply improvement heuristics for other periods as well.

Total collected amount for the planning horizon stays the same but the collected amounts for almost all periods are changed. The total purchase amount does not change and so purchasing cost stays same. However, with different collection amounts for the periods, the inventory amounts for each period and so the inventory cost change. Furthermore, since all the routes in the solution updated, transportation cost changes. Lastly, vehicle operation cost might also change.

4.4.2 Definition of the Moves

The 11 movements used in this algorithm are as below:

1. **Randomly remove ρ visits:**

   Randomly select one period and remove one random source node that is in Subset 3 or 4 and in that period. Then, apply Repair 1. This move is repeated ρ times.

2. **Randomly insert ρ visits:**

   Randomly select one source node in Subset 2 or 3 and one random period that the source node is not visited. Insert the source node is into the chosen period. Then, apply Repair 2. This move is repeated ρ times.

3. **Remove the worst source node:**

   Remove all the visits of one source node in Subset 2, 3 or 4 such that the objective value will save the most when it is removed from the visiting schedule. All the source nodes are removed one by one and the most efficient removal is chosen. Then, apply Repair 3.

4. **Insert the best source node:**

   Insert one source node in Subset 1 to a random period in such that the objective value will save the most, when it is inserted. Then, apply Repair 4.

5. **Shaw removal:**

   Randomly select one period and a source node in Subset 3 or 4 and in that period; compute the distance from the selected source node to the closest source node in the same period, namely $dist_{min}$. Then, remove all source nodes in Subset 3 or 4 within the range of 2*****$dist_{min}$ from the selected node in the same period. For each source node removed, apply Repair 1.

6. **Shaw insertion:**

   Randomly select one period and one source node in Subset 2 or 3 and that is not visited in that period. Then compute the distance to the closest source node, namely $dist_{min}$ and insert all source nodes in Subset 2 or 3 within the range of 2*$dist_{min}$ of the selected node into that period. For each source node inserted, apply Repair 2.

7. **Remove ρ source nodes:**

   Randomly select one source node in Subset 2, 3 or 4 and remove it from all periods. Then, apply Repair 3. This move is repeated ρ times.

8. **Insert ρ source nodes:**

   Randomly select one source node in Subset 1 and insert it into a random period. Then, apply Repair 4. This move is repeated ρ times.

9. **Empty one period:**

   Randomly select one period and remove all the source nodes in that period. For the removed source nodes in Subset 2, apply Repair 3 and for the removed source nodes in Subset 3 or 4, apply Repair 1.

10. **Swap routes:**

    Randomly select two periods and swap all the source nodes in these periods. Then, apply Repair 5.

11. **Randomly move ρ visits:**

    Randomly select one period and a random source node in Subset 2 or 3 and in that period, remove the source node from that period and insert it into another random period. For this move, first insert the source node into the second period and apply Repair 2. Then, remove the source node from the first period and apply Repair 1. This move is repeated ρ times.

In Chapter 5, we present the accumulation of real life data, and the computational results.

Pseudo code of ALNS algorithm for SPIRP – part 1

All weights are set to 1 and all scores are set to 0.
$s \leftarrow$ initial solution.
$s_{best} \leftarrow s$.
$\tau \leftarrow \tau_{start}$.
$\boldsymbol{while}$ *iterations* < 50,000 or *time* < 3,600 sec
    $s' \leftarrow s$.
    Select a movement using the roulette-wheel mechanism based on the weights of the current segment.
    Apply the movement to s' and update the number of times it is used.
    Fix routing decisions, solve the remaining problem taking into account inventory holding costs, purchasing cost, vehicle operating cost and transshipment cost.
    $\boldsymbol{if}\ f(s') < f(s)\ \boldsymbol{then}$
        $s \leftarrow s'$;
        $\boldsymbol{if}\ f(s) < f(s_{best})\ \boldsymbol{then}$
            $s_{best} \leftarrow s$;
            increase the score for the neighborhood used by σ_1;
        $\boldsymbol{else}$
            increase the score for the heuristic used by σ_2;
        $\boldsymbol{end\ if}$
    $\boldsymbol{else}$
        $\boldsymbol{if}\ s'$ is accepted by the simulated annealing criterion $\boldsymbol{then}$
            $s \leftarrow s'$;
            increase the score for the heuristic used by σ_3.
        $\boldsymbol{end\ if}$
    $\boldsymbol{end\ if}$
    $\boldsymbol{if}$ the end of the segment, 200 iterations, is reached $\boldsymbol{then}$
        update the weights of all heuristics and reset their scores.

Pseudo code of ALNS algorithm for SPIRP – part 2

$\quad\quad \boldsymbol{end\ if}$

$\quad\quad \tau \leftarrow \varphi\, \tau;$

$\boldsymbol{end\ while}$

Every 200 iterations perform intra-route 2-opt, intra-route 3opt, inter-route 2-opt, 3 route cycle customer exchange, customer move, 2-route customer exchange to improve the routes.

$\boldsymbol{return}\ s_{best}$

Chapter 5

COMPUTATIONAL RESULTS

In this chapter, we solve the selective and periodic inventory routing problem (SPIRP) in a case study about a waste collection logistics problem of a biodiesel production facility in Istanbul, Turkey by designing an adaptive large neighborhood search heuristic (ALNS). We solve problems with 20 to 100 source nodes for a 7-day cyclic planning horizon. To evaluate the performance of the algorithm, we generate lower bounds with the partial linear relaxation (PLR) model proposed by Aksen et. al. [1] and relaxation without routing (RR) model as described in Section (3.3). Moreover, for the instances with 20 to 40 source nodes, we solved the SPIRP model proposed by Aksen et. al. [1] to compare ALNS with SPIRP for small-sized instances. Due to the fact that SPIRP is shown in [1] to perform poorly for instances with 40 nodes, we do not run the model for larger instances. In the following sections, we describe the data, present the results and give some analysis.

5.1. Acquisition of the problem data

For waste vegetable oil collection, we include 20, 25, 30, 35, 40, 50, 60, 80, and 100 restaurants to the company's collection program. The restaurants and the recycling facility operated by the company constitute a complete collection network. The restaurants are the source nodes and the recycling facility is the depot. The asymmetric shortest path distances between each origin and destination pair on this complete network have been obtained from Google Maps. These distances are multiplied by the unit traveling cost correspond to the asymmetric arc costs of the complete network under consideration. All restaurants are located on the Asian side of Istanbul, while the depot is situated in Gebze, about 50 km east of Istanbul on the northern shore of the Sea of Marmara. The distribution of the restaurants used as source nodes over the Asian side of Istanbul in this study has a similar structure with the real life distribution of the restaurants in the Asian side of Istanbul. Figure (4.1) shows the source nodes' geographical distribution on the eastern side of the Bosphorus. The reason most of the source nodes appear to be close to each other is that the locations of the hospitality businesses and dining facilities are not dispersed homogenously over the Asian side of Istanbul in reality.

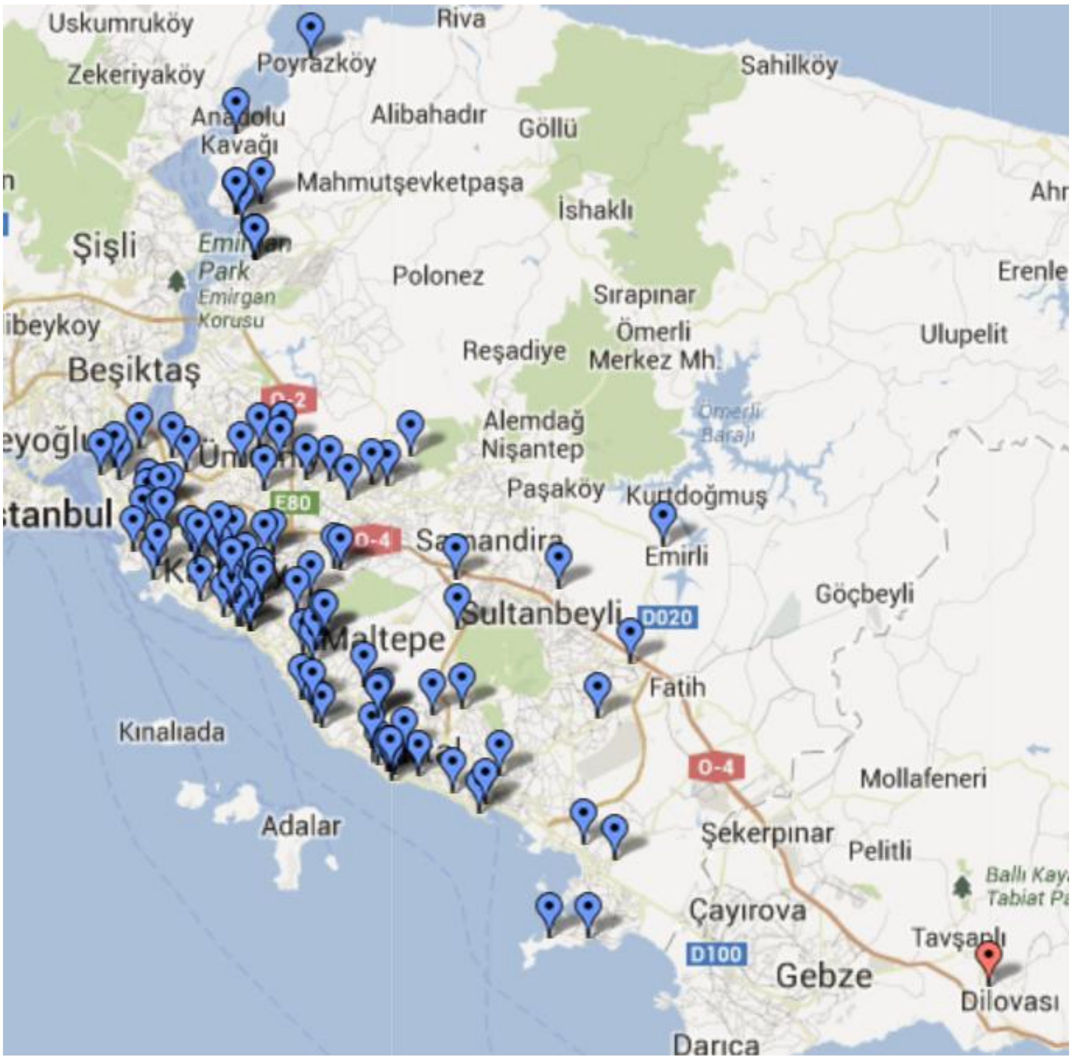

Figure 5. 1: The geographical locations of the restaurants on the Asian side of Istanbul

Besides the distances, there are several other input parameters such as the costs of inventory holding, transportation, purchasing, and vehicle operating; the vehicle capacity, the daily waste oil accumulation rates at each restaurant, and the daily waste oil requirement of the company. The values of these parameters are obtained, where possible, from various information sources on the web and through private communication with the company. For the daily accumulation rates, a questionnaire is used to be able to generate relatively realistic values. The instances are created to assess the performance of our algorithm with different real life scenarios.

The company policy is to adopt a uniform vehicle type for its collection operations. We used the light commercial vehicle: Fiat Fiorino Cargo. Its fuel and operating costs (parameters c and v in the SPIRP exact model) are calculated in Table (5.1). The data on driver wages, vehicle leasing costs, and Euro 4 diesel prices were inquired in August 2013, and may show fluctuations throughout the year.

Since virgin oil can also be used as raw material in biodiesel production [104], we assume that purchasing price (p) is at most the wholesale price of virgin vegetable oil, which is around 3.5 TL/lt. We also run the algorithm with another p value, namely 2.5 TL/lt. The cost of storing one liter of waste oil in the depot of the company, namely$h$, has been calculated as the daily interest rate times the highest purchasing price. This yields 0.02 TL/day for h.

Table 5. 1: Fuel and daily operating costs of the light commercial vehicle

Vehicle Model	**Fiat Fiorino Cargo**
Payload (except the driver)	550 kg
Fuel Consumption (urban)	6 lt / 100 km
Leasing Cost	50 TL/day
Vehicle Operating Cost	90 TL/day
Traveling Cost	0.24 TL/km
Wages of the Drivers	40 TL/day
Price of Euro 4 Diesel	3.91 TL/lt

The accumulation values a_{it} have been generated according to a questionnaire on waste oil collection in Turkey. The questionnaire shows that the large-sized restaurants accumulate approximately 50 lt/day waste oil per day, the medium-sized restaurants accumulate around 30 lt/day and the small-sized restaurants accumulate around 15 lt/day. These values are taken into account to generate related a_{it} values. For each day $t \in$

$\{1, \ldots, 7\}$, a_{it} values are derived from a normal distribution with means 15, 30 and 50 according to the restaurant type of the instance with variances 5, 15 and 25 respectively.

We test the proposed ALNS algorithm with three levels of waste oil requirements to assess the performance of the algorithm under different scenarios; low, medium, and high. The requirements are calculated according to total accumulation in a week at the restaurants chosen as the source nodes. The low accumulations are calculated as collecting around ½ of the total accumulation, the medium accumulations are calculated as collecting around ¾ of the total accumulation and finally the high accumulations are calculated as collecting approximately all of the total accumulation in the restaurants. To calculate the requirements for each day, these total requirement values are divided by the cycle length, 7. Since the waste oil requirements of the company are determined according to long-term production plans, we assume that the daily requirements do not vary across the 7-day production cycle.

Overall 54 SPIRP case instances for a waste collection problem are generated. The instances differ in size of the source nodes, daily requirement levels, and vegetable oil purchasing prices per liter. The problem instance names are indicative of these specifications. For example, (20n-270r-2.5p) means that the number of source nodes in the collection program is 20, the daily waste oil requirement is 270 lt/day, and the unit purchasing price of virgin oil is 2.50 TL/lt.

5.2. Computing platform, Cplex options and JAVA programming

All experiments and scenario analyses were conducted on a server equipped with Intel Xeon E5-2643 3.30 GHz Quad-Core processor and 32 GB RAM. The operating system of this PC is 64-bit Windows 7 Professional Service Pack 1.

The ALNS algorithms are coded in JAVA and compiled with version 10.15.2.0. For every instance, 10 parallel ALNS runs are taken by converting Java files to jar files and starting the same 10 jar files through comment window. In this way, the computing load is distributed on all the cores. Among 10 results, the best result is recorded as the objective value of the algorithm and the time to run 10 parallel algorithms is recorded as the CPU time of the ALNS algorithm for each instance.

The 64-bit version of the mathematical modeling and optimization suite GAMS 23.7 was used to create the SPIRP and PLR models proposed by Aksen et al. [1]; and the RR model. Cplex 12.3 was employed with the following options turned on: nodelim 50000000; threads 0; parallelmode 1; workmem 30000; nodefileind 2 (GAMS/Cplex 12 Solver Manual). This way, the computing load of Cplex is distributed on to as many as four cores of the Xeon Quad-Core processor.

For the small instances, the SPIRP model is solved to compare the performance of the ALNS algorithm. The time limit for the SPIRP model was set to 2 hours, 2.5 hours, 3 hours, 3.5 hours and 4 hours for the instances with respectively 20, 25, 30, 35 and 40 source nodes. The model is not used to solve large instances since it starts to perform poorly as the instance size gets larger.

The lower bounds are calculated using both the PLR and the RR models to evaluate the performance of the ALNS algorithm. The PLR model was run for 1 hours, 1.25 hours, 1.5 hours, 1.75 hours, 2 hours, 2.5 hours, 3 hours, 4 hours, and 5 hours for instances with respectively 20, 25, 30, 35, 40, 50, 60, 80 and 100 source nodes. The RR model is solved instantly in all of the instances. Moreover, since the MILP model run for the small instances also give lower bounds, its lower bound values are also used as performance evaluators for the small instances.

5.3 Test results and optimality gaps

The ALNS algorithm is applied to 54 instances described in Section (5.1). To evaluate the quality of the ALNS solutions, three different lower bound methods are used as explained in Section (5.2). For each small instance, a lower bound is obtained by solving the SPIRP exact model in a limited time, the PRL model in a limited time and the RR model. For the large instances, only the PLR model and the RR models are solved since solving the exact model itself for these instances is shown to be time consuming and not useful. To evaluate the performance of the algorithm, we report the following:

- TC_{ALNS} : The best objective value of 10 ALNS runs in parallel.
- TC_{SPIRP} : The best feasible objective value, the final upper bound, obtained by running the exact MILP model for a preset time.
- CPU_{ALNS} : The CPU time of 10 parallel ALNS runs.

- CPU_{SPIRP} : The CPU time of the SPIRP model, a preset time limit.
- LB_{SPIRP} : The lower bound value of the SPIRP model when run for CPU_{SPIRP} .
- LB_{PLR} : The lower bound value of the PLR model when run for CPU_{SPIRP} /2.
- LB_{RR} : The optimal objective value of the RR model.
- $BestLB$: $Max\{ LB_{SPIRP} ,\ LB_{PLR} ,\ LB_{RR} \}$. (LB_{SPIRP} is available only for small instances)
- Gap_{SPIRP} : The final gap between the best feasible objective value obtained from the MILP model, namely TC_{SPIRP} and the best lower bound obtained by taking the highest value among LB_{SPIRP} , LB_{PLR} , and LB_{RR} , i.e. $\frac{TC_{SPIRP} - BestLB}{BestLB}$.
- Gap_{ALNS} : The final gap between the best objective value of 10 ALNS runs, namely TC_{ALNS} and the best lower bound obtained by taking the highest value among LB_{SPIRP} , LB_{PLR} , and LB_{RR} , i.e. $\frac{TC_{ALNS} - BestLB}{BestLB}$.
- $Gap_{SPIRP-ALNS}$: The final gap between the best feasible objective value obtained from the MILP model, namely TC_{SPIRP} and the best objective value of 10 ALNS runs, namely TC_{ALNS} , i.e. $\frac{TC_{SPIRP} - TC_{ALNS}}{TC_{SPIRP}}$.

First, we compare the performances of the ALNS algorithm and the SPIRP model on the small instances. For the instances with 20 and 25 nodes the exact model provides better objective values. When the size of the nodes increases to 30 and more, the ALNS algorithm outperforms the MILP model. The methods are evaluated with respect to the best of the lower bound values obtained by solving the exact model and the PLR model for limited times, and by solving the RR model to optimality. Table (5.2) presents the experimental results for the small instances and Table (5.3) summarizes the comparative performances of the two solution methods with respect to the node sizes.

Table 5. 2: Computational results for the small size instances

	LB_{SPIRP}	LB_{PLR}	LB_{RR}	TC_{SPIRP}	TC_{ALNS}	Gap_{SPIRP} (%)	Gap_{ALNS} (%)	CPU_{ALNS} (s)
Instances								
20n-270r-2.5p	411.7	**471.1**	466.7	**473.4**	480.1	0.48	1.89	3.2
20n-410r-2.5p	617.0	**701.0**	692.0	**707.9**	711.8	0.99	1.55	2.8
20n-540r-2.5p	803.7	**806.6**	788.4	**818.6**	830.6	1.49	2.98	8.6
20n-270r-3.5p	411.7	**471.1**	466.7	**473.4**	479.9	0.48	1.85	3.9
20n-410r-3.5p	617.0	**701.0**	692.0	**707.9**	712.0	0.99	1.58	3.7
20n-540r-3.5p	803.7	**806.6**	788.4	**818.6**	830.6	1.49	2.97	9.8
25n-320r-2.5p	482.8	**583.5**	579.3	594.4	**593.2**	1.86	1.65	4.7
25n-480r-2.5p	718.4	**798.0**	788.0	**808.4**	813.0	1.30	1.88	7.1
25n-640r-2.5p	966.1	**1,051.1**	1,030.0	1,078.3	**1,074.1**	2.59	2.19	25.4
25n-320r-3.5p	483.1	**583.3**	579.5	**591.0**	594.6	1.32	1.93	5.1
25n-480r-3.5p	718.2	**798.1**	788.0	**808.9**	820.2	1.35	2.76	6.7
25n-640r-3.5p	965.0	**1,056.4**	1,029.5	**1,078.1**	1,083.4	2.06	2.55	35.3
30n-420r-2.5p	617.0	**684.4**	680.8	**706.9**	709.4	3.28	3.65	12.1
30n-630r-2.5p	923.5	928.2	**940.6**	1,052.1	**1,008.4**	11.85	7.21	15.2
30n-840r-2.5p	1,240.3	**1,272.3**	1,237.9	1,421.3	**1,420.4**	11.71	11.64	224.4
30n-420r-3.5p	616.7	**684.3**	680.8	**702.5**	713.5	2.66	4.27	14.0
30n-630r-3.5p	923.5	930.1	**950.6**	**1,046.9**	1,066.3	10.13	12.17	14.1
30n-840r-3.5p	1,240.7	**1,272.5**	1,237.9	1,432.2	**1,421.5**	12.55	11.71	170.9
35n-480r-2.5p	700.4	**769.8**	764.7	**803.0**	809.0	4.31	5.09	6.3
35n-710r-2.5p	1,035.3	1,044.8	**1,064.6**	1,281.9	**1,178.3**	20.41	10.68	13.7
35n-950r-2.5p	1,402.0	1,432.5	**1,434.9**	1,630.5	**1,554.6**	13.63	8.34	96.1
35n-480r-3.5p	700.2	757.5	**764.7**	818.4	**810.8**	7.02	6.02	8.7
35n-710r-3.5p	1,035.3	1,068.4	**1,084.6**	1,184.4	**1,180.5**	9.20	8.85	10.9
35n-950r-3.5p	1,404.1	**1,450.5**	1,434.8	1,637.5	**1,581.2**	12.89	9.01	79.6
40n-550r-2.5p	**783.9**	782.2	772.8	937.3	**832.4**	19.58	6.19	7.5
40n-820r-2.5p	1,194.3	1,195.6	**1,230.8**	1,599.4	**1,334.5**	29.95	8.43	12.3
40n-1090r-2.5p	**1,584.4**	1,575.9	1,547.0	2,089.4	**1,673.4**	31.88	5.62	141.1
40n-550r-3.5p	**783.9**	782.2	772.8	937.3	**838.5**	19.58	6.97	9.5
40n-820r-3.5p	1,194.3	1,195.6	**1,230.8**	1,599.4	**1,373.5**	29.95	11.60	11.8
40n-1090r-3.5p	**1,584.4**	1,575.9	1,547.0	2,089.4	**1,671.5**	31.88	5.50	239.8

The best lower bound value of each instance is indicated with bold coloring in Table (5.2) among the first three columns. Moreover, the best objective value between the

SPIRP model and the ALNS algorithm is also colored bold for each instance. Especially for the small instances with up to 30 nodes, the PLR model performs better than other two lower bound methods. When the node size gets larger, the SPIRP and RR models also accomplish to give the best lower bounds for some instances. For the small instances with 20 and 25 nodes, the objective values obtained from the SPIRP model find near optimal solutions with less than 2% gap on average. Even though for these instances the ALNS algorithm cannot give better solutions than the SPIRP model, it performs nearly as good in really short times.

Table 5. 3: MILP versus ALNS performance on small instances (on average)

	TC_{SPIRP}	TC_{ALNS}	Gap_{SPIRP} (%)	Gap_{ALNS} (%)	$Gap_{SPIRP-ALNS}$ (%)	CPU_{ALNS} (s)	CPU_{SPIRP} (s)
N							
20	**666.7**	674.2	**1.07**	2.21	-1.13	5.3	7200
25	**826.5**	829.7	**1.82**	2.22	-0.39	14.1	9000
30	1060.3	**1056.6**	9.60	**9.21**	0.35	75.1	10800
35	1226.0	**1185.7**	11.98	**8.30**	3.28	35.9	12600
40	1542.0	**1287.3**	28.54	**7.30**	16.52	70.3	14400

Table (5.3) summarizes the performances of the two methods comparatively on average values over the small instances. As we state before, ALNS is not as effective as the SPIRP model to solve small instances. However, the ALNS outperforms the SPIRP model when number of node is 30 and higher. Table (5.3) shows the success of ALNS as the node size gets larger. $Gap_{SPIRP-ALNS}$ starting with negative values becomes positive for larger instances. For the instances with 40 nodes, ALNS improves the SPIRP model's best objective value by 16.5%. In addition, the average CPU times of ALNS is inconsiderable compared to the CPU times of the SPIRP model.

We also test our algorithm on large instances to solve more realistic problems in reasonable times as shown in Table (5.4). Because the SPIRP model does not perform well for large instances even with long CPU times, we only present here results of the ALNS algorithm for these instances. To assess the quality of ALNS, we use the PLR and the RR models to obtain lower bounds. We indicate the best lower bound for each instance with bold coloring in Table (5.4). We cannot say any lower bound model outperforms the other one. For different instances, different models give the best lower bounds. For the large instances, the algorithm performs with 10.7% gap on average. As the node size increases, the average gap shows an upward trend. This is also true for the CPU times.

Table 5. 4: Computational results for the large size instances

	$\mathbf{LB_{PLR}}$	$\mathbf{LB_{RR}}$	$\mathbf{TC_{ALNS}}$	$\mathbf{Gap_{ALNS}}$ **(%)**	$\mathbf{CPU_{ALNS}}$ **(s)**
Instances					
50n-650r-2.5p	944.0	**1,000.8**	1,069.3	6.84	10.2
50n-910r-2.5p	1,315.1	**1,331.4**	1,447.1	8.69	19.2
50n-1300r-2.5p	**1,886.7**	1,877.5	2,129.9	12.89	286.4
50n-650r-3.5p	944.0	**1,002.0**	1,070.2	6.80	12.4
50n-910r-3.5p	1,315.1	**1,334.5**	1,489.6	11.62	29.7
50n-1300r-3.5p	**1,886.7**	1,876.4	2,134.2	13.12	291.4
60n-800r-2.5p	1,163.1	**1,229.3**	1,320.8	7.44	17.4
60n-1190r-2.5p	1,716.8	**1,781.9**	1,936.5	8.67	48.6
60n-1590r-2.5p	2,316.4	**2,344.9**	2,544.9	8.53	432.4
60n-800r-3.5p	1,163.1	**1,229.3**	1,326.5	7.90	19.3
60n-1190r-3.5p	1,716.8	**1,781.9**	1,954.1	9.67	55.1
60n-1590r-3.5p	2,316.4	**2,344.9**	2,562.6	9.29	395.0
80n-1070r-2.5p	1,517.3	**1,529.8**	1,722.8	12.62	59.5
80n-1610r-2.5p	**2,297.2**	2,293.3	2,535.4	10.37	173.2
80n-2150r-2.5p	**3,193.2**	3,155.1	3,473.1	8.76	979.5
80n-1070r-3.5p	1,517.3	**1,529.8**	1,732.8	13.27	53.0
80n-1610r-3.5p	**2,297.2**	2,293.3	2,581.9	12.39	198.7
80n-2150r-3.5p	**3,193.2**	3,155.1	3,512.1	9.99	1,038.3
100n-1330r-2.5p	**1,832.3**	1,806.0	2,083.5	13.71	89.1
100n-2000r-2.5p	**2,830.0**	2,734.7	3,149.7	11.30	256.3
100n-2670r-2.5p	**3,905.9**	3,665.5	4,293.2	9.92	3,492.3
100n-1330r-3.5p	**1,832.3**	1,806.0	2,149.1	17.29	77.4
100n-2000r-3.5p	**2,830.0**	2,734.7	3,233.7	14.27	295.1
100n-2670r-3.5p	**3,905.9**	3,665.5	4,333.3	10.94	3,294.8

5.4 Analysis of the ALNS Algorithm

In this section, we present some summary information about the performance of our algorithm. As the size of the problem gets larger than 80, the CPU time of the ALNS grows rapidly as shown in Figure (5.2). While the algorithm ends on average in a couple of seconds for our smallest instances, it takes almost half an hour for the largest ones. However, when compared to the SPIRP model, the longest CPU time of our algorithm for the largest instance is even smaller than the shortest CPU time of the SPIRP model for the smallest instance. When compared timewise, our algorithm is a sure winner.

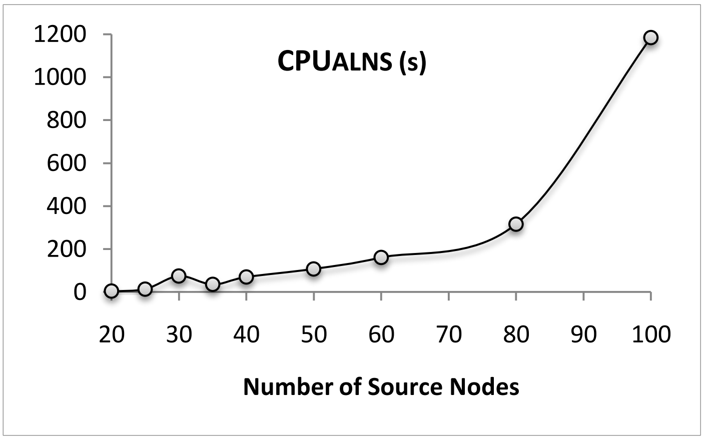

Figure 5. 2: The CPU time of ALNS with respect to the number of source nodes

We also observe that there is a similar upward trend in CPU times for all instances with different number of source nodes when the requirement levels of raw material are increased, which can be seen in Table (5.5). For the same size of source nodes, higher requirement level increases the solution time of the algorithm. With fewer requirements, the company is freer to choose among different source nodes to apply different moves. However, as the requirement levels increase, the selective part of the problem weakens and the moves designed for this characteristic of the algorithm become ineffective. The algorithm has difficulty to find moves that can be applied to the solution. In addition, the extensive randomness in the algorithm affects the CPU time negatively when the solution space gets smaller. If it was for the other metaheuristics, the CPU time of the algorithm might decrease with the decrease in the solution space since those methods are based on covering a local area completely. Another factor causing higher CPU times with high requirements is the reconstruction processes of the routes as in the update step of most moves. Solving the parallel savings and improvement heuristics to construct new routes take much more time since almost all the source nodes are needed to be visited in the solution to meet high requirements.

Table 5. 5: CPU times with respect to requirement levels and number of source nodes

CPU_{ALNS} (s)									
	Number of Source Nodes								
	20	25	30	35	40	50	60	80	100
Requirements									
LOW	7	10	26	15	17	23	37	113	166
MED	6	14	29	25	24	49	104	372	551
HIGH	18	61	395	176	381	578	827	2,018	6,787

We next analyze the performance of each move separately. To conduct our analysis, we chose one instance from each group of instances with the same number of source nodes, namely 9 test instances. One by one we removed all the neighborhoods from the algorithm and ran these new algorithms for each instance. We observe that removing each neighborhood deteriorates our algorithm, proving the need for each single move for the success of ALNS. However, while removing some moves creates great worsening at the quality of the solution, some others affect the solution less. Removing the moves "Remove ρ source nodes" and "Remove the worst source node" affect the solution relatively more compared to the other moves.

Furthermore, we evaluate the effectiveness of the moves in finding a better incumbent solution, a better current solution and a new solution accepted with simulated annealing as presented in Table (5.6). These values are the sum of 10 runs for all test instances. "Remove the worst source node" move is seen to be more effective in finding new incumbent solution and a better current solution with respect to the other moves. "Insert the best source node" and "Swap routes" moves also seems more successful updating the both incumbent and current solutions. As it is stated before, different characteristics of

ALNS moves is one of the strongest features of this algorithm. Even though some moves shows more success, it is not possible to carry solution into better solutions without other moves. We also record the percentages of the solution updates during the algorithm. With only 0.16% of iterations, the incumbent solution is updated. However almost half of the iterations, only the current solution is updated. Overall 66.5% of iterations, a new solution is accepted during the algorithm.

Table 5. 6: The performances of moves in finding new solutions

	Incumbent	**Current**	**Acceptance**
Moves			
Randomly remove ρ visits	13	86579	58176
Randomly insert ρ visits	8	17571	116775
Remove the worst source node	6356	1075346	370
Insert the best source node	344	192351	114469
Shaw removal	19	86809	54546
Shaw insertion	16	92844	107949
Remove ρ source nodes	74	69928	115086
Insert ρ source nodes	24	100943	109297
Empty one period	14	65958	60358
Swap routes	263	217583	113939
Randomly move ρ visits	27	54863	73957
% of Solution Updates in the Algorithm	**0.16**	**45.79**	**20.55**

Even though some moves perform better according to Table (5.6), we note that all the neighborhoods impact the performance of our algorithm either by intensifying or diversifying the search with their different characteristics.

Chapter 6

CONCLUSIONS

In this thesis, we studied a selective and periodic inventory routing problem (SPIRP) for a waste vegetable oil collecting biodiesel production facility. We developed an adaptive large neighborhood search (ALNS) algorithm for this reverse logistics problem. It requires to decide on which of the present source points to include in the collection program, which periodic routing schedule to repeat over an infinite horizon, how much virgin oil to purchase on each day and how many vehicles and in which routes to operate such that the total collection, inventory and purchasing costs are minimized while the production requirements and operational constraints are met.

The main features of an ALNS algorithm are large neighborhoods, changing weight of the moves with the past performances, roulette-wheel selection mechanism as well as an acceptance criterion for the neighborhood solutions inspired by Simulated Annealing. We implemented ALNS to SPIRP by applying a rich set of neighborhoods.

The MILP for this problem proposed recently was not successful to solve instances more than 30 source nodes with a reasonable gap in a reasonable time. Moreover, the model gives near optimal solutions for small size instances only with long CPU times.

In this study, we solve 54 problem instances of size 20 to 100. For small sized instances up to 40 nodes, we compare our results with the MILP model. In instances with less than 30 source nodes, ALNS cannot perform as well as MILP. However, for 30 and above, ALNS outperforms MILP significantly. For instances with 40 source nodes, ALNS improves the MILP solutions with 16.5% on the average. The CPU time of our algorithm is only several seconds for small instances, whereas the MILP model works for hours to give similar gaps with our algorithm.

For larger instances, we compare our algorithm with respect to lower bounds obtained from two models, namely PLR and RR. However, none of these two models could provide us tight enough bounds. PLR was proposed earlier as a partial relaxation model of the MILP. The new relaxation model RR outperforms PLR for some instances. For instances with 50 to 100 source nodes, we solve the problem with 10.7% gap in 484 seconds on average.

As future work the lower bounds can be improved. Moreover, new neighborhoods specific to this problem may be introduced. Also, we can study the SPIRP with heterogeneous fleet and stochastic accumulation rates.

BIBLIOGRAPHY

[1] D. Aksen, O. Kaya, F. Salman, and Y. Akça. Selective and periodic inventory routing problem for waste vegetable oil collection. *Optimization Letters*, 6(6): 1063-1080, 2012.

[2] K. Gaudette and R. Giuntini. Remanufacturing: The next great opportunity for boosting us productivity. *Business Horizons*, 46(6): 41-48, 2003.

[3] V. Jain. Special issue on sustainable supply chain management and reverse logistics. *International Journal of Production Research*, 50(5):1239-1242, 2012.

[4] M. G. Kulkarni and A. K. Dalai. Waste cooking oil—An Economical Source for Biodiesel: A Review. *Industrial & Engineering Chemistry Research*, 45(9): 2901-2913, 2006.

[5] J. F. Gonzalez, J. M. Encinar and A. Rodriguez-Reinares. Biodiesel from used frying oil. Variables affecting the yields and characteristics of the biodiesel. *Industrial and Engineering Chemistry Research*, 44(15):5491-5499, 2005.

[6] Z. J. Predojevic. The production of biodiesel from waste frying oils: A comparison of different purifcation steps. *Fuel*, 87(17-18): 3522-3528, 2008.

[7] B. Adams, L. G. Schumacher and J. V. Gerpen. Biodiesel fuels. *Elsevier, Encyclopedia of Energ*, 1:151-162, 2004.

[8] M. Pugazhvadivua and K. Jeyachandran. Investigations on the performance and exhaust emissions of a diesel engine using preheated wco as fuel. *Renewable Energy*, 30:2189-2202, 2005.

[9] Albiyobir. (in turkish: Alternatif enerji ve biyodizel Üreticileri birliği), 2010. Biyodizel nedir? (What is biodiesel?). Access at: http://www.albiyobir.org.tr/biyodizel.htm., Retrieved on October 27, 2010.

[10] L. VanWassenhove, P. Beullens and D. Van Oudheusde. Collection and vehicle routing issues in reverse logistics, R. Dekker, M. Fleischmann and K. Inderfurth (Eds.). Reverse logistics - quantitative models for closed loop supply chains: 95-134, 2004.

[11] J.M. Bloemhof-Ruwaard, M. Fleischmann, P. Beullens and L.V. Wassenhove. The impact of product recovery on logistics network design. *Production and Operations Management,* 10(2):156-173, 2001.

[12] A.P. Antunes, J.P. de Souza and J. Teixeira. Recyclable waste collection planning-a case study. *European Journal of Operational Research*, 158(3): 543-554, 2004.

[13] G. Zobolas, C.D. Tarantilis, G. Ioannou, P.P. Repoussis and D.C. Paraskevopoulos. A web-based decision support system for waste lube oils collection and recycling. *European Journal of Operational Research*, 195(3): 676-700, 2009.

[14] J. H. Ramser and G. B. Danzig. The truck dispatching problem. *Management Science*, 6(1): 80-91, 1959.

[15] F. Mafiolli. The vehicle routing problem: A book review. *Operations Research*, 1(2): 149- 153, 2003.

[16] D. Vigo and P. Toth. Branch-and-bound algorithms for the capacitated VRP. *Society for Industrial and Applied Mathematics*, pages 29-51, 2001.

[17] G. Clarke and J.V. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4): 568-581, 1964.

[18] S. Eilon, C. Watson-Gandy and N. Christofides. *Distribution Management*, Griffin, London, 1971.

[19] T.G. Crainic and G. Laporte, (Eds.). *Fleet Management and Logistics*, Kluwer Academic Publishers, 1998.

[20] B.L. Golden and A.A. Assad, (Eds.). *Vehicle Routing: Methods and Studies*, North Holland, Amsterdam, 1988.

[21] P. Toth and D. Vigo, (Eds.). *The Vehicle Routing Problem*, Monographs on Discrete Mathematics and Applications 9, SIAM. 2.20, 2002b.

[22] A.A. Assad, Modeling and Implementation Issues in Vehicle Routing, B.L. Golden, A.A. Assad (Eds.). Vehicle Routing: Methods and Studies, North Holland, Amsterdam, 7-46. 1988.

[23] L. Bodin and B. Golden. Classification in Vehicle Routing and Scheduling, *Networks*, 11(2): 97-108, 1981.

[24] L. Bodin, B. Golden, A.A. Assad and M. Ball. The State of the Art in the Routing and Scheduling of Vehicles and Crews, *Computers and Operation Research,* 10(2):63-212. 1983.

[25] N. Christofides, A. Mignozzi and P. Toth. *The* Vehicle Routing Problem, N. Christofides, A. Mignozzi, P. Toth and C. Sandi (Eds.), Combinatorial Optimization, Wiley, Chichester. 1979

[26] N. Christofides, Vehicle Routing, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (Eds.). The Traveling Salesman Problem, 431-448, Wiley, Chichester. 1985.

[27] J.F. Cordeau, G. Deasulniers, J. Desrosiers, M.M. Solomon and F. Soumis. VRP with Time Windows, P. Toth and D. Vigo (Eds.), The Vehicle Routing Problem, Monographs on Discrete Mathematics and Applications 9, 157-193, SIAM. 2002.

[28] J.F. Cordeau, M. Gendreau, A. Hertz, G. Laporte and J.S. Sormany. New Heuristics for the Vehicle Routing Problem, A. Langevine and D. Riopel, (Eds.), Logistics Systems: Design and Optimization, Wiley and Sons, 279- 298. 2005.

[29] M. Desrochers, J.K. Lenstra, and M.W.P. Savelsbergh. A Classification Scheme for Vehicle Routing and Scheduling Problems, *European Journal of Operational Research*, 46(3): 322-332, 1990.

[30] M.L. Fisher, Vehicle Routing, M.O. Ball, T.L. Magnanti, C.L. Momma, G.L. Nemhauser (Eds.). Network Routing, Handbooks in Operations Research and Management Science 8, 1-33, North Holland, Amsterdam. 1995.

[31] G. Laporte. The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms, *European Journal of Operational Research*, 59(3): 345-358, 1992.

[32] G. Laporte , M. Gendreau, J.-Y. Potvin, and F. Semet. Classical and Modern Heuristics for the Vehicle Routing Problem, *International Transactions in Operational Research*, 7(4-5): 285-300, 2000.

[33] J. K. Lenstra and A.H.G. Rinnoy Kan. Complexity of Vehicle Routing and Scheduling Problems. *Networks* 11(2), 221-227, 1981.

[34] Y. Marinakis and A. Migdalas. Heuristic Solutions of Vehicle Routing Problems in Supply Chain Management, Combinatorial and Global Optimization, P.M. Pardalos et al. (Eds.), Scientific World, 205-235, 2002.

[35] T. Magnanti. Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects. *Networks*, 11(2), 179-213, 1981.

[36] P. Toth and D. Vigo. An Overview of Vehicle Routing Problems, P. Toth and D. Vigo. (Eds.), The Vehicle Routing Problem, Monographs on Discrete Mathematics and Applications 9, 1-26, SIAM. 2002c.

[37] Y. Nobert and G. Laporte. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations Research*, 5(2): 77-85, 1983.

[38] D. Vigo, M. Fischetti and P. Toth. A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Operations Research*, 42(5):846-859, 1994.

[39] N.R. Achuthan, K.K. Lau and M.J. Kumar. Parallel implementation of branch and bound algorithm for solving vehicle routing problem on nows. *Third international symposium on parallel architecture, Algorithms and Networks*, 247-253, 1997.

[40] E. Benavent, A. Corbern, D. Naddef, G. Rinaldi, P. Augerat and J.M. Belenguer. Computational results with a branch and cut code for the capacitated vehicle routing problem. *Technical Research Report 949-M, Université Joseph Fourier*, 1995.

[41] G. Yu, J.F. Bard and G. Kontoravdis. A branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science*, 36(2):250-269, 2002.

[42] R.W. Eglese, J. Lysgaard and A.N. Letchford. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100:250-269, 2002.

[43] M. P. de Aragão, M. Reis, E. Uchoa, R. F. Werneck, R. Fukasawa and J. Lysgaard. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Proceedings of the X IPCO, Lecture Notes in Computer Science*, 3064:1-15, 2004.

[44] J. Lysgaard, M. P. de Aragão, M. Reis, E. Uchoa, R. F. Werneck and R. Fukasawa, H. Longo. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106:491-511, 2006.

[45] P. Francis, K. Smilowitz, and M. Tzur. The periodic vehicle routing problem and its extensions. B. L. Golden, S. Raghavan, and E. A. Wasil (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges, pages 239-261. Springer, New York, 2008

[46] L. D. Bodin and E. J. Beltrami. Networks and vehicle routing for municipal waste collection. *Networks*, 4(1):65-94, 1974.

[47] W. Igo and R. A. Russell. An assignment routing problem. *Networks*, 9(1):1-17, 1979.

[48] J. E. Beasley and N. Christofides. The period routing problem. *Networks*, 14(2):237-256, 1984.

[49] C. C. R. Tan and J. E. Beasley. A heuristic algorithm for the period vehicle routing problem. *Omega*, 12(5): 497-504. 1984.

[50] R. Baldacci, E. Bartolini, A. Mingozzi and A.Valletta. An Exact Algorithm for the Period Routing Problem. *Operations Research*, 59(1):228-241, 2011.

[51] J.-F. Cordeau, M. Gendreau and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 30(2):105-119. 1997.

[52] E. Hadjiconstantinou and R. Baldacci. A multi-depot period vehicle routing problem arising in the utilities sector. *The Journal of the Operational Research Society*, 49(12):1239–1248. 1998.

[53] E. Angelelli and M. G. Speranza. The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, 137(2): 233-247. 2002.

[54] J.-F. Cordeau, G. Laporte and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52(8):928–936, 2001.

[55] P. M. Francis, K. R. Smilowitz, and M. Tzur. The period vehicle routing problem with service choice. *Transportation Science*, 40(4):439-454, 2006.

[56] J. Alegre, M. Laguna and J. Pacheco. Optimizing the Periodic Pick-up of Raw Materials for a Manufacturer of Auto Parts. *European Journal of Operational Research*, 179(3):736-746, 2007

[57] R. A. Russell. An effective heuristic for the M-tour traveling salesman problem with some side conditions. *Operations Research*, 25(3):517-524, 1977.

[58] R. A. Russell and D. Gribbin. A multiphase approach to the period routing problem. *Networks* 21(7):747-765, 1991.

[59] M. Gaudioso and G. Paletta. A heuristic for the period vehicle routing problem. *Transportation Science* 26(2):86-92, 1992.

[60] I. M. Chao, B. L. Golden and E. A. Wasil. An improved heuristic for the period vehicle routing problem. *Networks*, 26(1):22-44, 1995.

[61] L.M.A. Drummond, L.S. Ochi and D.S. Vianna. An asynchronous parallel metaheuristic for the period vehicle routing problem. *Future Generation Computer Systems*, 17(4):379-386, 2001

[62] V. Hemmelmayr, K. Doerner, and R. Hartl. A variable neighborhood search for periodic routing problems. *European Journal of Operational Research*, 195(3): 791-802, 2009.

[63] T. Vidal, T. Crainic, M. Gendreau, N. Lahrichi and W. Rei. A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems. *Operations Research,* 60(3): 611-624, 2012.

[64] J. Pacheco, A. Alvarez, I. García and F. Angel-Bello. Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates. *Journal of the Operational Research Society,* 63(5): 569-581, 2011.

[65] V. Cacchiani, V. C. Hemmelmayr and F. Tricoire. A Set Covering based Heuristic Algorithm for the Periodic Vehicle Routing Problem. *Discrete Applied Mathematics*, 2013

[66] K. R. Smilowitz and P. M. Francis, Modeling techniques for periodic vehicle routing problems. *Transportation Research, Part B*, 40(10):872-884, 2006.

[67] F. Vanderbeck and M. Mourgaya. Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, 183(3):1028-1041, 2007.

[68] E. Taniguchi, RG. Thompson and T. Yamada. Modelling city logistics. E. Taniguchi, RG.Thompson, (Eds.), City Logistics I, Institute of Systems Science Research , Kyoto, 3-37, 1999.

[69] E. Beltrami and L. Bodin. Networks and vehicle routing for municipal waste collection. *Networks*, 4(1):65-94, 1974.

[70] B. Golden, A. Assad and E. Wasil. Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy and newspaper industry. P. Toth, D. Vigo, (eds.) The Vehicle Routing Problem, 245–286. SIAM, Philadelphia. 2001.

[71] L. Bodin, A. Mingozzi, R. Baldacci and M. Ball. The rollon-rolloff vehicle routing problem. *Transportation Science*, 34(3): 271-288, 2000.

[72] T. Kulcar. Optimizing solid waste collection in Brussels. *European Journal of Operational Research*, 90(1): 71–77, 1996.

[73] D. Eisenstein and A. Iyer. Garbage collection in Chicago: a dynamic scheduling model. *Management Science*, 43(7): 922–933, 1997.

[74] V. Jayaraman, VDR. Guige Jr and R. Srivastava. A closed-loop logistics model for remanufacturing. *Journal of Operational Research Society*, 50(5):497-508, 1999.

[75] H.R. Krikke, A. Van Harten and P.C. Schuur. Reverse logistic network redesign for copiers. *OR Spektrum*, 21:381-409, 1999.

[76] D. Tung and A. Pinnoi. Vehicle routing-scheduling for waste collection in Hanoi. *European Journal of Operational Research*, 125(3): 449-468, 2000

[77] M. Mourão and M. Almeida. Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. *European Journal of Operational Research*, 121(2):420-434, 2000.

[78] S. Baptista, R. Oliveira and E. Zuquete. A period vehicle routing case study. *European Journal of Operational Research*, 139(2): 220-229, 2002.

[79] E. Angelelli and M. Speranza. The application of a vehicle routing model to a waste-collection problem: two case studies. *Journal of Operational Research Society*, 53(9): 944-952, 2002.

[80] C. Archetti and M. Speranza. Vehicle routing in the 1-skip collection problem. *Journal of Operational Research Society,* 55(7):717-727, 2004.

[81] P. Lacomme, C. Prins and W. Ramdane-Cherif. Evolutionary algorithms for periodic arc routing problems. *European Journal of Operational Research*, 165(2), 535-553, 2005.

[82] C. Prins and S. Bouchenoua. A memetic algorithm solving the VRP, the CARP and general routing problems with nodes, edges and arcs. *Recent Advances in Memetic Algorithms*, 166: 65-85, 2005.

[83] O. Listes and R. Dekker. A stochastic approach to a case study for product recovery network design. *European Journal of Operational Research*, 160(1):268–287, 2007.

[84] H. Üster, G. Easwaran, E. Akçali and S. Çetinkaya. Benders decomposition with alternative multiple cuts for a multi-product closed-loop supply chain network design model. *Naval Research Logistics,* 54(8):890–907, 2007.

[85] Z. Lu and N. Bostel. A facility location model for logistics systems including reverse flows: the case of remanufacturing activities. *Computers & Operations Research*, 34(2):299-323, 2007.

[86] R. Wojanowski, V. Verter and T. Boyaci. Retail–collection network design under deposit–refund. *Computers & Operations Research*, 34(2):324–345, 2007.

[87] B. Crevier, J. Cordeau and G. Laporte. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2): 756-773, 2007.

[88] C. Tarantilis, E. Zachariadis and C. Kiranoudis, A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. *INFORMS Journal on Computing*, 20(1):154, 2008.

[89] F. Du, G.W. Evans. A bi-objective reverse logistics network analysis for post-sale service. *Computers & Operations Research*, 35(8):2617–2634, 2008

[90] N. Aras, D. Aksen and A.G. Tanugur. Locating collection centers for incentive-dependent returns under a pickup policy with capacitated vehicles. *European Journal of Operational Research*, 191(3):1223–1240, 2008.

[91] M.S. Pishvaee, K. Kianfar and B. Karimi. Reverse logistics network design using simulated annealing, *The International Journal of Advanced Manufacturing Technology*, 47:269-28, 2010.

[92] V. Hemmelmayr, K. Doerner, R. Hartl and S. Rath. A heuristic solution method for node routing based solid waste collection problems, *Journal of Heuristics,* 19(2):129-156, 2013.

[93] K. Buhrkal, A. Larsen and S. Ropke. The waste collection vehicle routing problem with time windows in a city logistics context, *Procedia- Social and Behavioral Sciences*, 39: 241-254, 2012.

[94] A. Angulo, H. Nachtmann, and M. A. Waller. Supply chain information sharing in a vendor managed inventory partnership. *Journal of Business Logistics*, 25(1):101-120, 2004.

[95] H. L. Lee and W. Seungjin. The whose, where and how of inventory control design. *Supply Chain Management Review*, 12(8):22-29, 2008

[96] D. Simchi-Levi, X. Chen, and J. Bramel. *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics and Supply Chain Management*. Springer-Verlag, New York, 2005.

[97] M. J. Geiger and M. Sevaux. Practical inventory routing: A problem definition and an optimization method. C. Artner, K. F. Doerner, R. F. Hartl, and F. Tricoire (Eds.), Proceedings of the EU/MEeting: Workshop on Client-Centered Logistics and International Aid, 32-35, Vienna, 2011a.

[98] A. Campbell and M. Savelsbergh. "Inventory Routing in Practice", The Vehicle Routing Problem, P. Toth and D. Vigo (Eds.), The Vehicle Routing Problem, Monographs on Discrete Mathematics and Applications 9, SIAM. 2002.

[99] A. Kleywegt, M. Savelsbergh, A. Campbell and L. Clarke. The inventory routing problem. in: T. Crainic and G. Laporte (Eds.), Fleet Management and Logistics, Kluwer Academic Publishers, 95-113, 1998.

[100] L. Purpura. Vendor-run inventory: Are its benefits exaggerated? *Supermarket News*, 47:59-60, 1997.

[101] C.Radice. Two heads are better than one. *Chain store executive with shopping center age*. 74:60-61, 1999.

[102] B. Mongelluzzo, Shippers let vendors manage the stock: Wal- mart's suppliers share in databases. *Journal of Commerce and Commercial*, 12A:417, 1998.

[103] Brumback N. Rubbermaid's new system scores. Home Furnishing *Network*, 69:11, 1995.

[104] M. Nannery. Aama promotes vendor-managed inventory; manufacturers say they can replenish better than stores. *Daily News Record*, 24:8, 1994.

[105] C. C. White III, C. G. Lee and Y. A. Bozer. A heuristic approach and properties of optimal solutions to the dynamic inventory routing problem. Working Paper, University of Toronto, Toronto, Ontario, Canada., 2003.

[106] D. M. Miller. An interactive, computer-aided ship scheduling system. *European Journal of Operational Research*, 32(3):363-379, 1987.

[107] S. Dauzere-Peres, A. Nordli, A. Olstad, K. Haugen, U. Koester, M. P. Olav, G. Teistklub, and A. Reistad. Omya Hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to European paper manufacturers. *Interfaces*, 37(1):39-51, 2007.

[108] F. Al-Khayyal and S.-J. Hwang. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part I: Applications and model. *European Journal of Operational Research*, 176(1):106-130, 2007

[109] R. Grønhaug, M. Christiansen, G. Desaulniers, and J. Desrosiers. A branch-and-price method for a liquefied natural gas inventory routing problem. *Transportation Science*, 44(3):400-415, 2010.

[110] J. A. Persson and M. Göthe-Lundgren. Shipment planning at oil refineries using column generation and valid inequalities. *European Journal of Operational Research*, 163(3):631-652, 2005.

[111] J. G. Rakke, M. Stalhane, C. R. Moe, M. Christiansen, H. Andersson, K. Fagerholt, and I. Norstad. A rolling horizon heuristic for creating a liquefied natural gas annual delivery program. *Transportation Research Part C: Emerging Technologies*, 19(5):896-911, 2011.

[112] Q. Shen, F. Chu, and H. Chen. A lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. *Computers & Chemical Engineering*, 35(10):2113-2123, 2011

[113] J. H. Song and K. C. Furman. A maritime inventory routing problem: Practical approach. *Computers & Operations Research*, 40(3):657-665, 2013.

[114] L. C. Coelho, J.-F. Cordeau and G. Laporte. Thirty Years of Inventory-Routing. *Transportation Science*, 2013.

[115] H. Andersson, A. Hoff, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515-1536, 2010.

[116] W. J. Bell, L. M. Dalberto, M. L. Fisher, A. J. Greenfield, R. Jaikumar, P. Kedia, R. G. Mack, and P. J. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4-23, 1983

[117] A. Federgruen and P. H. Zipkin. A combined vehicle-routing and inventory allocation problem. *Operations Research*, 32(5):1019-1037, 1984.

[118] M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle-routing. *Networks*, 11(2): 109-124, 1981.

[119] D. E. Blumenfeld, L. D. Burns, J. D. Diltz, and C. F. Daganzo. Analyzing trade-offs between transportation, inventory and production costs on freight networks. *Transportation Research Part B: Methodological*, 19(5):361-380, 1985.

[120] L. D. Burns, R. W. Hall, D. E. Blumenfeld, and C. F. Daganzo. Distribution strategies that minimize transportation and inventory costs. *Operations Research*, 33(3):469-490, 1985.

[121] M. Dror, M. O. Ball, and B. L. Golden. A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research*, 4(1):3-23, 1985.

[122] M. Dror and M. O. Ball. Inventory/routing: Reduction from an annual to a short-period problem. *Naval Research Logistics*, 34(6):891-905, 1987.

[123] M. Dror and L. Levy. A vehicle routing improvement algorithm comparison of a \greedy" and a matching implementation for inventory routing. *Computers & Operations Research*, 13(1):33-45, 1986.

[124] S. Anily and A. Federgruen. One warehouse multiple retailer systems with vehicle routing costs. *Management Science*, 36(1):92-114, 1990.

[125] C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382-391, 2007.

[126] O. Solyalı and H. Süral. A branch-and-cut algorithm using a strong formulation and an a priori tour based heuristic for an inventory-routing problem. *Transportation Science*, 45(3):335-345, 2011.

[127] L. C. Coelho and G. Laporte. The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, 40(2):558-565, 2013a.

[128] Y. Adulyasak, J.-F. Cordeau, and R. Jans. Formulations and branch-and-cut algorithms for multi-vehicle production and inventory routing problems. *INFORMS Journal on Computing*, 2013.

[129] M. Gendreau and J.-Y. Potvin. Handbook of Metaheuristics, *International Series in Operations Research and Management Science*. 146, Springer, New York, 2010.

[130] G. R. Raidl, J. Puchinger, and C. Blum. Metaheuristic hybrids. M. Gendreau and J.-Y. Potvin (Eds.), Handbook of Metaheuristics, 469-496. Springer, New York, 2010.

[131] V. Maniezzo, T. Stützle, and S. Voß. *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*. Springer, New York, 2009.

[132] A. M. Campbell and M. W. P. Savelsbergh. A decomposition approach for the inventory-routing problem. *Transportation Science*, 38(4):488-502, 2004.

[133] G. Gallego and D. Simchi-Levi. On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer r-systems. *Management Science*, 36(2):240-243, 1990.

[134] E.-H. Aghezzaf, B. Raa, and H. van Landeghem. Modeling inventory routing problems in supply chains of high consumption products. *European Journal of Operational Research*, 169(3):1048-1063, 2006.

[135] B. Raa and E.-H. Aghezzaf. A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research*, 192(2):429-441, 2009.

[136] T. W. Chien, A. Balakrishnan, and R. T. Wong. An integrated inventory allocation and vehicle routing problem. *Transportation Science*, 23(2):67-76, 1989.

[137] T. F. Abdelmaguid. *Heuristic approaches for the integrated inventory distribution problem.* Ph.D. dissertation, University of Southern California, Los Angeles, 2004.

[138] T. F. Abdelmaguid and M. M. Dessouky. A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Research*, 44(21):4445-4464, 2006.

[139] T. F. Abdelmaguid, M. M. Dessouky, and F. Ordonez. Heuristic approaches for the inventory-routing problem with backlogging. *Computers & Industrial Engineering*, 56(4):1519-1534, 2009.

[140] M. W. P. Savelsbergh and J. H. Song. An optimization algorithm for the inventory routing problem with continuous moves. *Computers & Operations Research*, 35(7):2266-2282, 2008.

[141] B. Raa and E.-H. Aghezzaf. Designing distribution patterns for long-term inventory routing with constant demand rates. *International Journal of Production Economics*, 112(1):255-263, 2008.

[142] M. J. Geiger and M. Sevaux. On the use of reference points for the biobjective Inventory Routing Problem. S. Ceschia, L. Di Gaspero, M. Loghi, A. Schaerf, and T. Urli (Eds.), Proceedings of the MIC 2011: The IX Metaheuritics International Conference, 141-149, Udine, 2011b.

[143] S. Michel and F. Vanderbeck. A column-generation based tactical planning method for inventory routing. *Operations Research*, 60(2):382-397, 2012.

[144] A. M. Campbell, L. Clarke, A. J. Kleywegt, and M. W. P. Savelsbergh. The inventory routing problem. T. G. Crainic and G. Laporte (Eds.), Fleet Management and Logistics, 95-113. Springer, Boston, 1998.

[145] L. Bertazzi and M. G. Speranza. Continuous and discrete shipping strategies for the single link problem. *Transportation Science*, 36(3):314-325, 2002.

[146] C. Archetti, L. Bertazzi, A. Hertz, and M. G. Speranza. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101-116, 2012.

[147] L. C. Coelho, J.-F. Cordeau, and G. Laporte. The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537-2548, 2012a.

[148] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24(1):270-287, 2012b.

[149] M. Hewitt, G. L. Nemhauser, M. W. P. Savelsbergh, and J.-H. Song. A branch-and-price guided search approach to maritime inventory routing. *Computers & Operations Research*, 40(5):1410-1419, 2013.

[150] P. Chandra. A dynamic distribution model with warehouse and customer replenishment requirements. *Journal of the Operational Research Society*, 44(7):681-692, 1993.

[151] P. Chandra and M. L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503-517, 1994.

[152] Y. T. Herer and R. Roundy. Heuristic for one-warehouse multiretailer distribution problem with performance bounds. *Operations Research*, 45(1):102-115, 1997.

[153] F. Fumero and C. Vercellis. Synchronized development of production, inventory and distribution schedules. *Transportation Science*, 33(3):330-340, 1999.

[154] L. Bertazzi, G. Paletta, and M. G. Speranza. Minimizing the total cost in an integrated vendor-managed inventory system. *Journal of Heuristics*, 11(5-6):393-419, 2005.

[155] J. F. Bard and N. Nananukul. Heuristics for a multiperiod inventory routing problem with productiondecisions. *Computers & Industrial Engineering*, 57(3):713-723, 2009.

[156] J. F. Bard and N. Nananukul. A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37(12):2202-2217, 2010.

[157] C. Archetti, L. Bertazzi, G. Paletta, and M. G. Speranza. Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 12(38):1731-1746, 2011.

[158] Y. Adulyasak, J.-F. Cordeau, and R. Jans. Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, forthcoming, 2012. doi: 10.1287/trsc.1120.0443.

[159] M. G. Speranza and W. Ukovich. Minimizing transportation and inventory costs for several products on a single link. *Operations Research*, 42(5):879-894, 1994.

[160} M. G. Speranza and W. Ukovich. An algorithm for optimal shipments with given frequencies. *Naval Research Logistics*, 43(5):655-671, 1996.

[161] L. Bertazzi, M. G. Speranza, and W. Ukovich. Minimization of logistic costs with given frequencies. *Transportation Research Part B: Methodological*, 31(4):327-340, 1997.

[162] D. Popovic, M. Vidovic, and G. Radivojevic. Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. *Expert Systems with Applications*, 39(18):13390-13398, 2012.

[163] N. H. Moin, S. Salhi, and N. A. B. Aziz. An effcient hybrid genetic algorithm for the multi-product multiperiod inventory routing problem. *International Journal of Production Economics*, 133(1):334-343, 2011.

[164] A. Mjirda, B. Jarboui, R. Macedo, and S. Hanafi. A variable neighborhood search for the multi-product inventory routing problem. *Electronic Notes in Discrete Mathematics*, 39(1):91-98, 2012.

[165] N. Ramkumar, P. Subramanian, T. Narendran, and K. Ganesh. Mixed integer linear programming model for multi-commodity multi-depot inventory routing problem. *OPSEARCH*, 49(4):413-429, 2012.

[166] D. Ronen. Marine inventory routing: shipments planning. *Journal of the Operational Research Society*, 53 (1):108-114, 2002.

[167] L. C. Coelho and G. Laporte. A branch-and-cut algorithm for the multi-product multi-vehicle inventoryrouting problem. *International Journal of Production Research*, forthcoming, 2013b. doi: 10.1080/ 00207543.2012.757668.

[168] A. J. Kleywegt, V. S. Nori, and M. W. P. Savelsbergh. The stochastic inventory routing problem with direct deliveries. *Transportation Science*, 36(1):94-118, 2002.

[169] L. Bertazzi. Analysis of direct shipping policies in an inventory-routing problem with discrete shipping times. *Management Science*, 54(4):748-762, 2008.

[170] G. Gallego and D. Simchi-Levi. Rejoinder to 'A note on bounds for direct shipping costs'. *Management Science*, 40(10):1393, 1994.

[171] Q.-H. Zhao, S.-Y. Wang, and K. K. Lai. A partition approach to the inventory/routing problem. *European Journal of Operational Research*, 177(2):786-802, 2007.

[172] S. Viswanathan and K. Mathur. Integrating routing and inventory decisions in one warehouse multiretailer multiproduct distribution system. *Management Science*, 43(3):294-312, 1997.

[173] R. Roundy. 98%-effective integer-ratio lot-sizing for one-warehouse multi-retailer systems. *Management Science*, 31(11):1416-1430, 1985.

[174] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Dynamic and stochastic inventory-routing. *Technical Report CIRRELT-2012-37*, Montreal, Canada, 2012c.

[175] J. E. Beasley. Fixed routes. *Journal of the Operational Research Society*, 35(1):49-55, 1984.

[176] C. A. Barlett and S. Ghoshal. Building competitive advantage through people. *MIT Sloan Management Review*, 43(2):34-41, 2002.

[177] H. Zhong, R. W. Hall, and M. M. Dessouky. Territory planning and vehicle dispatching with driver learning. *Transportation Science*, 41(1):74-89, 2007.

[178] K. Smilowitz, M. Nowak, and T. Jiang. Workforce management in periodic delivery operations. *Transportation Science*, 47(2): 214-230, 2013.

[179] J. F. Bard, L. Huang, P. Jaillet, and M. Dror. Decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science*, 32(2):189-203, 1998.

[180] A. Federgruen, G. Prastacos, and P. H. Zipkin. An allocation and distribution model for perishable products. *Operations Research*, 34(1):75-82, 1986.

[181] B. L. Golden, A. A. Assad, and R. Dahl. Analysis of a large-scale vehicle-routing problem with an inventory component. *Large Scale Systems in Information and Decision Technologies*, 7(2-3):181-190, 1984.

[182] P. Jaillet, J. F. Bard, L. Huang, and M. Dror. Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science*, 36(3):292-300, 2002.

[183] F. Li, B. L. Golden, and E. A. Wasil. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34(9):2734-2742, 2007.

[184] S.-C. Liu and W.-T. Lee. A heuristic method for the inventory routing problem with time windows. *Expert Systems with Applications*, 38(10):13223-13231, 2011.

[185] A. S. Minkoff. A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. *Operations Research*, 41(1):77-90, 1993.

[186] O. Berman and R. C. Larson. Deliveries in an inventory/routing problem using stochastic dynamic programming. *Transportation Science*, 35(2):192-213, 2001.

[187] A. J. Kleywegt, V. S. Nori, and M. W. P. Savelsbergh. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science*, 38(1):42-70, 2004.

[188] D. Adelman. A price-directed approach to stochastic inventory/routing. *Operations Research*, 52(4):499-514, 2004.

[189] W. W. Qu, J. H. Bookbinder, and P. Iyogun. An integrated inventory-transportation system with modified periodic policy for multiple products. *European Journal of Operational Research*, 115(2):254-269, 1999.

[190] S.-H. Huang and P.-C. Lin. A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):598-611, 2010.

[191] L. M. Hvattum and A. Lokketangen. Using scenario trees and progressive hedging for stochastic inventory routing problems. *Journal of Heuristics*, 15(6):527-557, 2009.

[192] L. M. Hvattum, A. Lokketangen, and G. Laporte. Scenario tree-based heuristics for stochastic inventoryrouting problems. *INFORMS Journal on Computing*, 21(2):268-285, 2009.

[193] E.-H. Aghezzaf. Robust distribution planning for the supplier-managed inventory agreements when demand rates and travel times are stationary. *Journal of the Operational Research Society*, 59(8):1055-1065, 2008.

[194] O. Solyalı, J.-F. Cordeau, and G. Laporte. Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327-340, 2012.

[195] D. Pisinger, S. Ropke; Large neighborhood search, F. Glover, G.A. Kochenberger (Eds.), Handbook of Metaheuristics, 2nd edition, 399-419, Springer, New York; 2010.

[196] P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. *Principles and Practice of Constraint Programming*, *Lecture Notes in Computer Science*, 1520:417–431, 1998.

[197] R.K. Ahuja, Ö. Ergun, J.B. Orlin, and A.P. Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123(1-3): 75–102, 2002.

[198] S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4): 455–472 2006.

[199] L.C. Coelho, J.-F. Cordeau, G. Laporte, The inventory-routing problem with transshipment. *Computers & operations Research*, 39(11): 2537-2548, 2012.

[200] M. Chiarandini, Vehicle Routing Local Search based Metaheuristics. *Lecture Notes in Scheduling, Timetabling and Routing*, 2011.

[201] G. Laporte and F. Semet, Classical Heurisrics for CVRP. P. Toth and D. Vigo (Eds.), The Vehicle Routing Problem, Monographs on Discrete Mathematics and Applications 9, SIAM. 2002.

VITA

Özge Tüncel was born in Sakarya, Turkey on October 29, 1987. She graduated from Kırşehir Science High School in 2005. She received her BSc. degrees in Industrial Engineering from Koç University, Istanbul, in 2010. During her BSc. degree she completed four internships in Otokar, IBM, İş Yatırım, and Koç Net and two part-time jobs in Pfizer and Finansbank. In 2010 after completing BSc. degree, she joined the MSc. program in Industrial Engineering at Koç University as a research and teaching assistant. Currently, she is a Ph.D. student in Engineering Systems and Design programme in Singapore University of Technology and Design.