ARISTOTLE UNIVERSITY OF THESSALONIKI

DOCTORAL THESIS

Solution methods for complex supply chain network optimization problems

By

Panagiotis KARAKOSTAS

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Chemical Engineering

June 2020

Declaration of Authorship

I, Panagiotis KARAKOSTAS, declare that this thesis titled, “Solution methods for complex supply chain network optimization problems” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Solution methods for complex supply chain network optimization problems

by Panagiotis KARAKOSTAS

Examination Committee Members

Michael C. Georgiadis
Professor
Supervisor
Dept. of Chemical Engineering
Aristotle University of Thessaloniki

Angelo Sifaleras
Associate Professor
Advisory Committee Member
Dept. of Applied Informatics
University of Macedonia

Eustathios S. Kikkinides
Professor
Advisory Committee Member
Dept. of Chemical Engineering
Aristotle University of Thessaloniki

Vlachos Dimitrios
Professor
Examiner
Dept. of Mechanical Engineering
Aristotle University of Thessaloniki

Hristu - Varsakelis Dimitrios
Professor
Examiner
Dept. of Applied Informatics
University of Macedonia

Nenes George
Associate Professor
Examiner
Dept. of Mechanical Engineering
University of Western Macedonia

Zikopoulos Christos
Assistant Professor
Examiner
Dept. of Economics
Aristotle University of Thessaloniki

"We must free ourselves of the hope that the sea will ever rest. We must learn to sail in high winds."

Aristotle Onassis

ARISTOTLE UNIVERSITY OF THESSALONIKI

Abstract

Faculty of Engineering
Department of Chemical Engineering

Doctor of Philosophy

Solution methods for complex supply chain network optimization problems

by Panagiotis KARAKOSTAS

The intertemporal integration of supply chain activities is crucial in developing sustained competitive advantage in the modern entrepreneurial environment. This integration refer to the simultaneous optimization of strategic, tactical and operational decisions in complex supply chain networks. Moreover, due to the fact that the supply chain activities emit pollutants, like carbon dioxide (CO_2), increased socio-environmental concerns have shifted the focus to a balanced goal, which integrates economic, environmental and social goals. To achieve these goals, the solution of complex supply chain optimization problems is required. New advanced optimization techniques and tools must be developed to assist decision makers.

This thesis studies and investigates new complex and integrated supply chain network optimization problems under economic and environmental. Efficient metaheuristic-based solution methods are developed for the solution of these problems to derive useful managerial insights.

More specifically, a well-known combinatorial optimization problem which integrates strategic, tactical and operation decisions is the Location-Inventory-Routing Problem. Herein, several variants of this complex problem are addressed. Initially, a Location-Inventory-Routing Problem with Distribution Outsourcing is introduced to describe cases where the proprietary fleet of vehicles is either cost-inefficient or specific fleet of vehicles is required (i.e. customer-specific). Furthermore, a green variant of the Location-Inventory-Routing Problem, the Pollution-Location-Inventory-Routing Problem is addressed by considering both economic and environmental concerns. In this new problem, decisions related to fuel consumption and CO_2 emissions are considered. Next, another green variant of the basic problem is proposed.

This problem is called the Fleet Size and Mix Pollution Location Inventory Routing Problem and considers more realistic features, such as fleet composition and capacity planning decisions. Furthermore, a healthcare supply chain network optimization problem related to innovative CAR T-cell cancer therapies is modelled and investigated. In this, a novel network structure is proposed to manage challenges associated to the increased demand for these therapies.

To obtain high-quality solutions of these problems in short computational times, several heuristic algorithms, based on the Variable Neighborhood Search metaheuristic framework, are proposed. Solution achieved using these methods are compared to the corresponding ones using state-of-the-art exact commercial solver, such as CPLEX. Extensive comparisons and numerical tests illustrate the efficiency of the proposed metaheuristic methods using key performance indicators.

Περίληψη

Η ενοποίηση των δραστηριοτήτων μίας εφοδιαστικής αλυσίδας αποτελεί κρίσιμο παράγοντα επίτευξης αειφόρου ανταγωνιστικού πλεονεκτήματος εντός των πλαισίων του σύγχρονου επιχειρησιακού περιβάλλοντος. Η εν λόγω ενοποίηση αφορά την ταυτόχρονη λήψη αποφάσεων στρατηγικού, τακτικού και επιχειρησιακού επιπέδου μέσω της αποδοτικής διαχείρισης της εφοδιαστικής αλυσίδας. Επιπρόσθετα, δεδομένης της εκπομπής ρύπων από τις δραστηριότητες της εφοδιαστικής αλυσίδας, όπως το διοξείδιο του άνθρακα, αυξημένες κοινωνικές και περιβαλλοντικές ανησυχίες έχουν μετατοπίσει την εστίαση στην επίτευξη ενός ισορροπημένου στόχου, ο οποίος αποτελεί ένα μίγμα οικονομικών, περιβαλλοντικών και κοινωνικών στόχων.

Η παρούσα διδακτορική διατριβή μελετά και διερευνά νέα σύνθετα προβλήματα βελτιστοποίησης δικτύων ενοποιημένων εφοδιαστικών αλυσίδων με αντικειμενικό στόχο την οικονομική, την περιβαλλοντική ή τη συνδυαστική απόδοση. Επίσης, νέες μεθευρετικές τεχνικές επίλυσης έχουν αναπτυχθεί ώστε να λυθούν τα νέα προβλήματα και να καταγραφούν χρήσιμες παρατηρήσεις αποδοτικής διαχείρισης.

Ακριβέστερα, ένα γνωστό πρόβλημα συνδυαστικής βελτιστοποίησης, το οποίο συνδυάζει στρατηγικές, τακτικές και λειτουργικές αποφάσεις είναι το σύνθετο Πρόβλημα Χωροθέτησης Εγκαταστάσεων, Ελέγχου Αποθεμάτων και Δρομολόγησης Οχημάτων. Στην παρούσα διατριβή εισάγονται αρκετές επεκτάσεις του συγκεκριμένου σύνθετου προβλήματος. Αρχικά, εισάγεται το σύνθετο Πρόβλημα Χωροθέτησης Εγκαταστάσεων, Ελέγχου Αποθεμάτων και Δρομολόγησης Οχημάτων με Εξωτερική Ανάθεση Διανομής για την περιγραφή περιπτώσεων όπου είτε ο ιδιόκτητος στόλος οχημάτων είναι μη οικονομικώς συμφέρουσα επιλογή ή απαιτείται ειδικού σκοπού στόλος οχημάτων (π.χ. στόλος οχημάτων προσαρμοσμένος στις ανάγκες των πελατών). Επιπλέον, προτείνεται μία επέκταση πράσινης διαχείρισης του σύνθετου Προβλήματος Χωροθέτησης Εγκαταστάσεων, Ελέγχου Αποθεμάτων και Δρομολόγησης Οχημάτων, ονομαζόμενο ως σύνθετο Πρόβλημα Χωροθέτησης Εγκαταστάσεων, Ελέγχου Αποθεμάτων, Δρομολόγησης Οχημάτων, Κατανάλωσης Καυσίμου και Εκπομπής Ρύπων Διοξειδίου του Άνθρακα, το οποίο λαμβάνει ταυτόχρονα οικονομικές και περιβαλλοντικές αποφάσεις, όπως η κατανάλωση καυσίμου και η εκπομπή ρύπων CO_2. Στη συνέχεια, ένα ακόμη πρόβλημα πράσινης διαχείρισης, επέκταση του βασικού προβλήματος, εισάγεται. Το νέο πρόβλημα καλείται σύνθετο Πρόβλημα Καθορισμού Μεγέθους και Σύνθεσης Στόλου Οχημάτων, Χωροθέτησης Εγκαταστάσεων με Πολλαπλές Επιλογές Χωρητικότητας, Ελέγχου Αποθεμάτων, Δρομολόγησης Οχημάτων, Κατανάλωσης Καυσίμου και Εκπομπής Ρύπων Διοξειδίου του Άνθρακα και λαμβάνει υπόψη επιπλέον ρεαλιστικά χαρακτηριστικά, όπως η σύνθεση στόλου και ο σχεδιασμός

χωρητικότητας εγκαταστάσεων. Επιπροσθέτως, ένα πρόβλημα βελτιστοποίησης δικτύων υγειονομικών εφοδιαστικών αλυσίδων σχετιζόμενων με τις καινοτόμες ανοσολογικές κυτταρικές θεραπείες του καρκίνου. Στο νέο αυτό πρόβλημα προτείνεται μία καινοτόμα δομή δικτύου για τη διαχείριση των προκλήσεων που προκύπτουν από την αυξανόμενη ζήτηση τέτοιων θεραπειών.

Για την αποδοτική επίλυση αυτών των προβλημάτων σε σύντομο υπολογιστικό χρόνο, προτείνονται αρκετοί ευρετικοί αλγόριθμοι βασιζόμενη στο μεθευρετικό πλαίσιο Αναζήτησης Μεταβλητής Γειτνίασης. Οι λύσεις που παρήχθησαν από τους προτεινόμενους αλγόριθμους συγκρίθηκαν με εκείνες του state-of-the-art εμπορικού λύτη, CPLEX. Πληθώρα συγκρίσεων και υπολογιστικών πειραμάτων υποδεικνύουν την αποδοτικότητα των προτεινόμενων μεθόδων και κύριους παράγοντες απόδοσης, ώστε τελικώς να κατασκευασθούν ταχείες και ισχυρές υπολογιστικές τεχνικές επίλυσης.

Acknowledgements

Several reasons can lead someone to commence a PhD study, such as the passion for research or the existence of a particular topic which rouses his curiosity. However, the journey toward completing a doctoral degree requires sacrifices and hard work. To overcome these difficulties, PhD candidates need appropriate support and guidance.

This page is devoted to people who have supported me in this long journey. To begin with, I would like to express my deepest gratitude and appreciation to my supervisor Professor Michael C. Georgiadis, who convincingly guided and encouraged me to be professional and focus on what really matters. His unrestricted support and constant feedback have made me to exert myself and complete this tough project. I would like to recognize the invaluable assistance that Professor Angelo Sifaleras provided during my research. Also, I am grateful to Professor Eufstathios Kikkinides for his courtesy to participate in my thesis advisory committee. Moreover, I would like to thank Professor Nicki Panoskaltsis and Professor Athanasios Mantalaris for their help and valuable feedback.

My friends and colleagues at the Department of Chemical Engineering in Aristotle University of Thessaloniki had a significant role in my long learning process of PhD. My heartfelt appreciation goes to Dr. George Nikolaidis and Dr. Nikolaos Koltsaklis for their warm encouragement and help. Moreover, I would like to thank my partners in crime, PhD candidates George Georgiadis and Apostolis Elekidis. I had great luck to share with them thoughts, knowledge and emotions. Special thanks also to Dr. Arsenios Chatzimichailidis and PhD candidates Evaggelos Tsimopoulos, Romuald Gyorgy and Nikolaos Koukovinos for their support.

Additionally, I owe a very important debt to PhD candidate Aggelos Kekempanos for sharing with me his knowledge on statistical analysis. Furthermore, I would like to thank Dr. Vitor Nazário Coelho and PhD candidate Christos Papalitsas for the productive discussions.

Of course, I would like to express my deepest and sincere gratitude to my family for their continuous and unconditional support and love. Particularly, I am deeply indebted to my wife Maria and to my furry son Rocky for their constant emotional support and their understanding. They have been a great source of motivation and encouragement during this cognitional marathon. Finally, I would like to thank my friends for all the wonderful times and their encouragement.

Contents

List of Figures

List of Tables

List of Abbreviations

AMOSA	**A**rchived **M**ulti-**O**bjective **S**imulated **A**nnealing
ALNS	**A**daptive **L**arge **N**eighborhood **S**earch
APCs	**A**ntigen-**P**resenting **C**ells
aVND	**a**daptive **V**ariable **N**eighborhood **D**escent
bVND	**b**asic **V**ariable **N**eighborhood **D**escent
BVNS	**B**asic **V**ariable **N**eighborhood **S**earch
B2C	**B**usiness **T**0 **C**onsumer
CO_2	**C**arbon **D**ioxide
CAR	**C**himeric **A**ntigen **R**eceptor
CO	**C**ombinatorial **O**ptimization
cVND	**c**yclic **V**ariable **N**eighborhood **D**escent
eVND	**e**xtended **V**ariable **N**eighborhood **D**escent
FLP	**F**acility **L**ocation **P**roblem
FA	**F**irefly **A**lgorithm
GBD	**G**eneral **B**enders **D**ecomposition
GVNS	**G**eneral **V**ariable **N**eighborhood **S**earch
GA	**G**enetic **A**lgorithm
GSCM	**G**reen **S**upply **C**hain **M**anagement
hSCM	**healthcare** **S**upply **C**hain **M**anagement
ICA	**I**mperialist **C**ompetitive **A**lgorithm
IACO	**I**mproved **A**nt **C**olony **O**ptimization
ICO	**I**nventory **C**ontrol **P**roblem
IRP	**I**nventory **R**outing **P**roblem
ILS	**I**terative **L**ocal **S**earch
JiT	**J**ust **i**n **T**ime
LARP	**L**ocation **A**rc **R**outing **P**roblem
LIP	**L**ocation **I**nventory **P**roblem
LIRP	**L**ocation **I**nventory **R**outing **P**roblem
LRP	**L**ocation **R**outing **P**roblem
MILP	**M**ixed **I**nteger **L**inear **P**rogramming
MILNP	**M**ixed **I**nteger **N**on-**L**inear **P**rogramming
MCDA	**M**ulticriteria **D**esicion **A**nalysis

MDLRP	**M**ulti-**D**epot **L**ocation **R**outing **P**roblem
MOMILP	**M**ulti-**O**bjective **M**ixed **I**nteger **L**inear **P**rogramming
MOMINLP	**M**ulti-**O**bjective **M**ixed **I**nteger **N**on-**L**inear **P**rogramming
MOPSO	**M**ulti-**O**bjective **P**article **S**warm **O**ptimization
MPCLIRP	**M**ulti-**P**roduct **C**apacitated **L**ocation **I**nventory **R**outing **P**roblem
NP	**N**on-**D**eterministic **P**olynomial
NP-complete	**N**on-**D**eterministic **P**olynomial **c**omplete
NP-hard	**N**on-**D**eterministic **P**olynomial **h**ard
NSGA-II	**N**on-dominated **S**orted **G**enetic **A**lgorithm-II
OR	**O**perational **R**esearch
OECD	**O**rganization for **E**conomic **C**o-Operation and **D**evelopment
PSO	**P**article **S**warm **O**ptimization
pVND	**p**ipe-**V**ariable **N**eighborhood **D**escent
P	**P**olynomial
PROMETHEE	**P**reference **R**anking **O**rganization **M**ethod for **E**nrichment **E**valuations
RPBNSGA-II	**R**eference **P**oint-**B**ased **N**on-dominated **S**orted **G**enetic **A**lgorithm-II
SA	**S**imulated **A**nnealing
SoSCM	**S**ocial **S**upply **C**hain **M**anagement
SCM	**S**upply **C**hain **M**anagement
SCN	**S**upply **C**hain **N**etwork
SCO	**S**upply **C**hain **O**ptimization
SSCM	**S**ustainable **S**upply **C**hain **M**anagement
TS	**T**abu **S**earch
TOPSIS	**T**echnique for **O**rder of **P**reference by **S**imilarity to **I**deal **S**olution
uVND	**u**nion **V**ariable **N**eighborhood **D**escent
VND	**V**ariable **N**eighborhood **D**escent
VNS	**V**ariable **N**eighborhood **S**earch
VRP	**V**ehicle **R**outing **P**roblem
VRPTW	**V**ehicle **R**outing **P**roblem with **T**ime **W**indows

Dedicated to Maria and Rocky

Chapter 1

Introduction

1.1 Supply Chain Optimization

The modern entrepreneurial environment constitutes a mixture of several interrelated components, such as social and technological components, in a constantly changing process. The inherent characteristics of this dynamic system are high complexity and uncertainty. Companies, to initially secure their survival and further their growth, have to develop a competitive advantage (Shadid, 2017). To achieve that, a company should manage efficiently its whole supply chain, which consists of all entities involved in fulfilling the demand requests of customers. Those entities are suppliers, facilities of the company (plants, warehouses and distribution centres) and customers (immediate customers and final consumers). They are connected by material, information and financial flows in an network representation, known as Supply Chain Network (SCN) (Stadler et al., 2015). Entities of a typical SCN are illustrated in Figure 1.1.

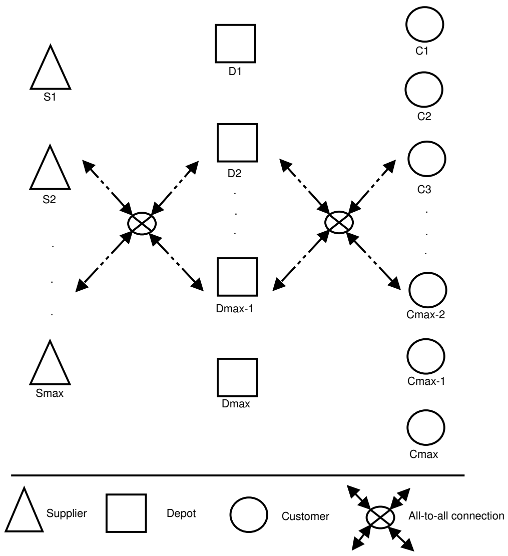

FIGURE 1.1: The entities of a typical SCN

The coordination of the necessary activities, associated with converting natural resources, raw materials and semi-finished product components into final products delivered to consumers, called Supply Chain Management (SCM) (Stadler et al., 2015). The enrichment of SCM with Operational Research (OR) techniques is known as Supply Chain Optimization (SCO). SCO constitutes a set of processes and tools developed and applied for optimizing the decision making process, in order to configure an efficient SCN (Speranza, 2018). An example of a configured SCN is depicted in Figure 1.2.

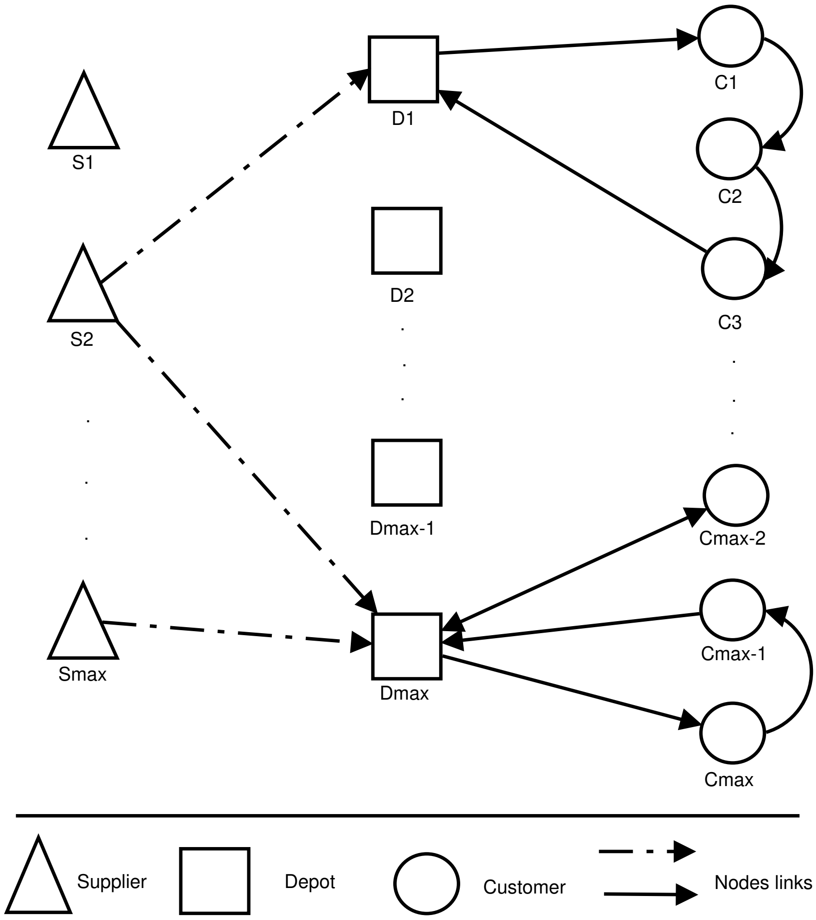

FIGURE 1.2: An example of a configured SCN

A relative recent trend in SCM refer to the systematic focus of the SCN (Speranza, 2018). The intertemporal integration of supply chain activities is crucial in developing sustained competitive advantage, as it leads to significant reduction of costs and increase of overall supply chain responsiveness (Aguirre et al., 2018; Vicente et al., 2015). More specifically, this integration is characterized by the simultaneous tackling of strategic, tactical and operational decisions (Hiassat et al., 2017). Strategic-level decisions refer to long-term planning such as resource acquisition decisions. Decisions in a medium-term planning horizon are characterized as tactical. For instance, the inventory planning is a tactical level decision. Operational decisions have a short execution horizon, such as the daily delivery schedules. The classic hierarchical supply chain decision-making process is depicted in Figure 1.3, while the integrated decision-making approach is illustrated in Figure 1.4.

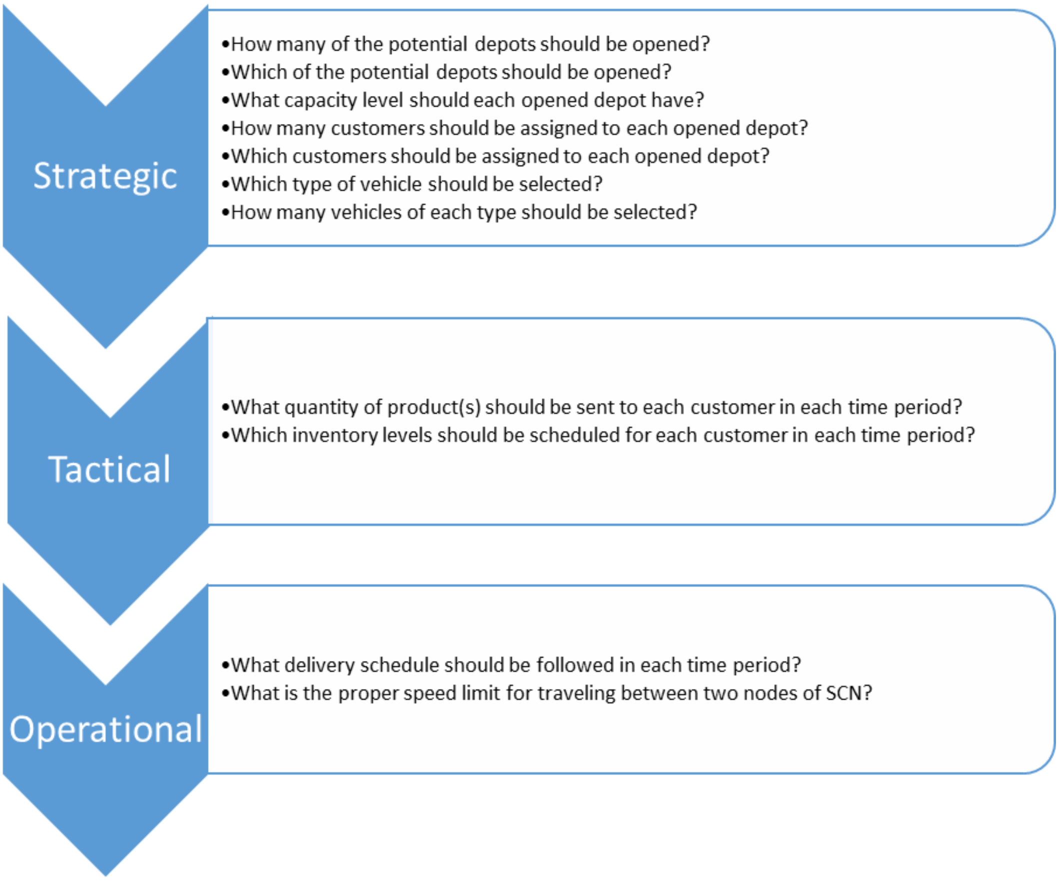

FIGURE 1.3: Hierarchical supply chain decision-making process

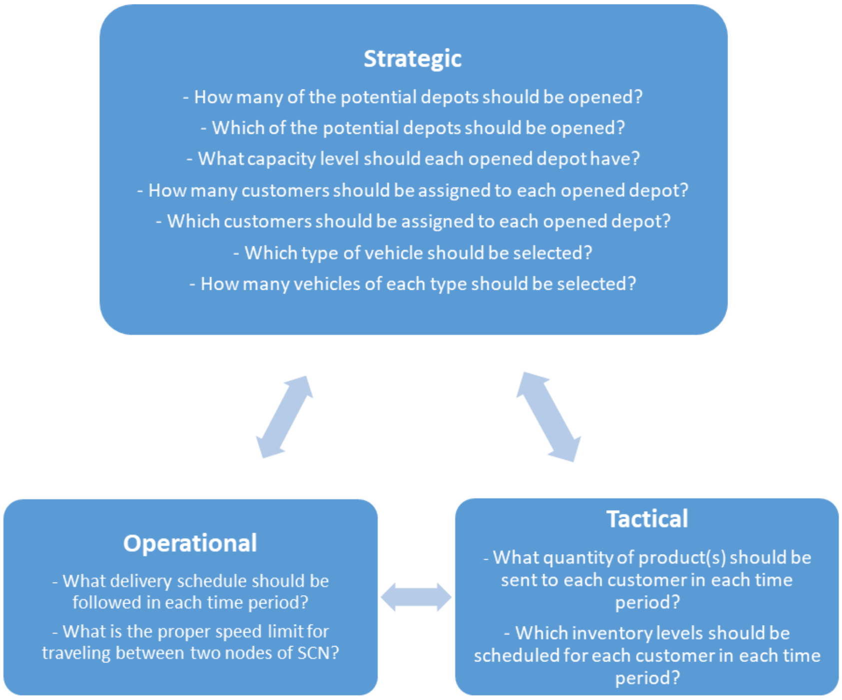

FIGURE 1.4: Integrated supply chain decision-making process

Three classic SCO problems have been used to address the three previously mentioned decision levels, separately:

- The *Facility Location Problem (FLP)* tackles the strategic-level decision of optimal designing a SCN.
- The *Inventory Control Problem (ICP)* refers to selecting the optimal inventory policy by determining tactical decisions, such as the delivery quantity and frequency.
- The *Vehicle Routing Problem (VRP)* which tackles operational-level decisions of scheduling optimal delivery routes.

Therefore, the hierarchical decision-making process refers to the sequential solution of these problems, which may lead to suboptimality (Zhang et al., 2014). Several contributions have been combined two of these classic problems in a single model, following the idea of integration, in an effort to overcome the potential sub-optimality. Traditionally, this two-level integration is addressed by the following complex SCO problems:

- The *Location Routing Problem (LRP)*, which tackles both strategic and operational decision levels (Cuda et al., 2015; Drexl & Schneider, 2015).

- The *Inventory Routing Problem (IRP)*, which tackles tactical and operational decisions (Dong et al., 2017; Soysal et al., 2019).
- The *Location Inventory Problem (LIP)*, which integrates strategic and tactical decisions (Farahani et al., 2015).

Amiri-Aref et al. (2018) studied a multi-period LIP considering multi-echelon SCN, demand uncertainty and a multi-sourcing strategy. They developed a two-stage stochastic mathematical model to maximize the total profit. A linearization method was applied to make the model tractable and a Sample Average Approximation algorithm, a Monte-Carlo simulation-based approach, is used to produce near-optimal solutions. Yu et al. (2019a) studied a capacitated LRP with tight capacity constraints on both depots and vehicles. To solve the problem, a hybrid Genetic Algorithm (GA) was developed which tackles feasible and infeasible solutions, by using a customized population management mechanism. Amiri et al. (2019) developed a Mixed Integer Non-Linear Programming (MINLP) model for the periodic LRP with time windows and fleet composition decisions. For the solution of the problem, a two-phase Lagrangean decomposition method was proposed. In the first phase, two sub-problems were solved. The first referred to products distribution from suppliers to onshore bases, while the second one to the delivery of products from onshore bases to offshore units. Then, a VRP with Time Windows (VRPTW) was solved to complete the overall problem solution. Darvish et al. (2019) investigated a flexible LRP by considering two sources of flexibility, the network design flexibility and the flexibility in due-dates. The first one refer to the potential selection of different facilities in a daily basis, while the second source of flexibility depends on the frequency of deliveries. Authors proposed a parallel exact algorithm to efficiently solve small- and medium-sized problem instances. They concluded that both sources of flexibility can lead to significant cost savings in comparison with fixed problem cases. A multi-depot IRP was studied by Bertazzi et al. (2019). They developed a Mixed Integer Linear Programming (MILP) model as well as a matheuristic algorithm based on a three-phase decomposition approach. The first phase is a clustering phase which build clusters of customers for each depot. The next is a route construction phase applied in each cluster and finally, an optimization phase which improves routing and delivery schemes. The proposed matheuristic outperformed a branch-and-cut algorithm with several families of cuts. Archetti et al. (2019) studied a variant of the IRP where the ratio between the total distribution cost and the total delivered quantity is minimized. They developed an exact algorithm based on the idea to sequentially solve different IRPs with a linear objective function. Also, two acceleration techniques were implemented to speed-up the proposed algorithm. A comparison between this enhanced version of the algorithm, called ACS, and another exact algorithm from the literature was performed. The results indicated that ACS outperformed the other algorithm in cases with small number of vehicles. Moreover, the proposed approach can solve problem cases with more customers.

Almouhanna et al. (2020) introduced a new variant of the LRP by considering electric vehicles and constrained distances. For the solution of the new problem, two heuristic algorithms were developed, a fast multi-start biased-randomized heuristic and a biased-randomized Variable Neighborhood Search (VNS) solution method. Numerical results showed that the proposed multi-start heuristic algorithm can generate quite good solutions in very short computational times. On the other hand, the VNS-based solution method leads to better solutions requiring more computational time. Oudouar et al. (2020) studied a capacitated LRP. They developed a two-steps hybrid solution approach to solve the problem under consideration. In its first step, location-allocations are made by using a self-organizing map algorithm, while routing decisions are tackled by applying a combination of the Clarke and Wright algorithm and the Or-opt local search method. The proposed solution approach was compared to the most effective heuristics in the literature. The results indicated that the hybrid algorithm can generate competitive algorithms and 38 new best found solutions were reported out of 79 problem instances. Alvarez et al. (2020) studied an IRP by considering a single type perishable commodity. More specifically, they considered perishability through a fixed shelf-life of an aging product under age-dependent inventory holding costs and sales revenues. Four mathematical formulations of the problem were presented and their advantages were further investigated. For the solution of small-sized problem instances exact algorithms were applied, while a hybrid Iterated Local Search (ILS) heuristic was developed for tackling efficiently larger problem cases. Markov et al. (2020) solved a recyclable waste collection IRP with heterogeneous fixed fleet of vehicles and stochastic demands. Authors developed an MINLP model as well as an Adaptive Large Neighborhood Search (ALNS) algorithm combined with a realistic demand forecasting model to solve the problem efficiently. Their proposed approach was compared to alternative deterministic policies for the control of occurrence of container overflows. This comparative study indicated the efficiency of their proposed approach. A cyclic IRP under Vendor Management Inventory policy was studied by Dai et al. (2020). They considered perishable products and price-dependent demand. To solve the problem, they developed MINLP models and a hybrid cuckoo algorithm with an improved Clarke-Wright savings algorithm. Their proposed algorithm proved quite efficient compared to optimization solver, CLPEX. Liu et al. (2020) studied an LIP in a stochastic supply chain system where random supply disruptions and stochastic demands and replenishment lead-times were considered. They developed a two-phase queuing theory-optimization model and a tailored hybrid GA to solve the problem. Araya-Sassi et al. (2020) introduced two multi-commodity LIP considering continuous and periodic review inventory control policies and modular stochastic capacity constraints. These new problems were formulated as MINLP models and a Lagrangian relaxation with a subgradient method were developed to tackle the high complexity of the problem under consideration. Tirkolaee et al. (2020) investigated a green LIP in order to design an efficient municipal solid waste management system. The problem was formulated as an MILP and several problems with real-life data were solved using CPLEX.

Other studies proposed the integration of two of the decision levels, while considering selected priorities of the third level. More specifically, a Multi-depot Location Routing Problem (MDLRP) taking into account inventory costs was considered by Liu and Lee (2003). They developed a two-phase heuristic applied on 144 random generated test instances. A hybrid Tabu Search (TS)/Simulated Annealing (SA) approach was then proposed for solving the same MDLRP (Liu & Lin, 2005). Furthermore, a linear programming model incorporating location, routing, and inventory decisions was proposed by Ambrosino and Scutella (2005), but feasible solutions were presented only for the case of the LRP on 12 single-period instances. Max Shen and Qi (2007) presented a nonlinear integer programming model for the integrated supply chain design. A Lagrangian-relaxation based algorithm was developed and its performance was evaluated on several randomly generated test instances. A Location Arc Routing Problem (LARP) with inventory constraints was studied by Riquelme-Rodriguez et al. (2016). They proposed two location constructive algorithms for building initial solutions and an ALNS algorithm for further improvement of solutions. Turan et al. (2017)

considered a two-echelon LIP within distribution decisions in order to rebalance inventory levels due to high demand uncertainty and high procurement costs. To efficiently tackle this problem, they developed a hybrid VNS algorithm with a dynamic programming method. A MDLRP considering inventory risks was studied by Zhao and Ke (2017) in the context of explosive waste management. They developed a multicriteria decision-making solution approach based on the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method in order to solve the studied problem in a reasonable computation time. By applying their method on a real application they observed 34% system costs decrease and 57% environmental risk reduction.

The integrated Location Inventory Routing Problem (LIRP) has received rather limited attention in the literature (Hiassat et al., 2017; Zhang et al., 2014). This problem considers simultaneous all three decisions levels and it is classified as an NP-Hard problem (Javid & Azad, 2010). Because of its computational complexity, large scale LIRP instances cannot be solved to optimality by exact solution methods (Eskandarpour et al., 2017). In order to overcome such computational limitations, heuristic and meta-heuristic approaches are often applied.

The first attempt to tackle simultaneously location, inventory and routing decisions was presented by Javid and Azad (2010). They proposed a MINLP model and a hybridization of TS and SA for solving large sized problem instances. Tavakkoli-Moghaddam et al. (2013) presented an MINLP model for a stochastic distribution network and solved five examples using the Lingo software. A hybrid Variable Neighborhood Descent (VND) - ILS metaheuristic solution approach was applied by Guerrero et al. (2013) for solving LIRP cases, described by an MILP formulation. Reza Sajjadi et al. (2013) developed an MINLP model for the two-layer multi-product capacitated LIRP (MP-CLIRP) and they solved larger instances using a sequential heuristic. Seyedhosseini et al. (2014) developed an MINLP model for the three-level SCN design and solved random generated instances with the Lingo solver for the small cases and a GA for larger instances. A two-stage hybrid TS heuristic for solving the combined LIRP in Business to Consumer (B2C) e-Commerce distribution system was proposed by Chen et al. (2014). Nekooghadirli et al. (2014) studied a bi-objective LIRP and applied four evolutionary based metaheuristics for solving several test instances. Zhang et al. (2014) presented an MILP model for the multi-period LIRP with flexible replenishment policy and they developed a hybrid SA metaheuristic for solving the proposed problem. Liu et al. (2015) studied a stochastic LIRP for designing a logistic system for e-commerce and implemented a Pseudo Parallel hybridization of GA and SA. Zhalechian et al. (2016) presented an MINLP model for a sustainable closed-loop LIRP and they applied a hybrid two-phase stochastic-possibilistic programming method within a game theory approach, in order to manage the uncertainty and a Self-Adaptive GA for addressing efficient solutions on large instances. A hybrid SA and Imperialist Competitive Algorithm (ICA) for tackling the LIRP was presented by Ghorbani and Akbari Jokar (2016). Hiassat et al. (2017) proposed evolutionary based optimization metaheuristics and more precisely, different versions of GA based solution approaches. Rayat et al. (2017a) studied an LIRP with multiple commodities and multiple time periods. They formulated the problem as Multi-Objective MINLP (MOMINLP) and they developed an Archived Multi-Objective SA (AMOSA) algorithm for the solution of large problem cases. To further improve the performance of proposed algorithm, they used the Taguchi method for tuning parameters.

Rafie-Majd et al. (2018) studied the design of a supply chain system of perishable products under uncertainty and they employed a Lagrangian Relaxation heuristic for solving it. Guo et al. (2018) studied a single-period closed-loop LIRP. They formulated the problem as a Nonlinear Integer Programming Model and developed a hybrid SA with an adaptive GA to solve it efficiently. Extensive sensitivity analysis was performed to highlight the impact of key solution method parameters on its efficiency. A multi-period, multi-stage closed-loop LIRP was investigated by Forouzanfar et al. (2018). They developed a bi-objective nonlinear integer model to address the problem. For the solution of small-sized problem instances the applied the ε-constrained method, while for solving larger problem cases they proposed a hybrid metaheuristic algorithm which combines the Non-Dominated Sorting GA-II (NSGA-II) and a multi-objective Particle Swarm Optimization (MOPSO). To improve the performance of the proposed solution method, they used the Taguchi method during parameter setting. Habibi et al. (2018) modeled an LIRP as an MINLP to optimize a microalgae biofuel supply chain system. To solve problem instances of practical interest, they properly modified three well-known metaheuristic frameworks, a SA, a GA and a Firefly Algorithm (FA). The obtained numerical results showed that the SA was the most efficient solution method and the GA performed better than FA.

Tavana et al. (2018) developed a Multi-objective MILP (MOMILP) model to address a new humanitarian LIRP, which considers pre- and post-disaster planning. For the solution of the problem, an Epsilon-constraint method was initially developed. However, it was not able to produce any solution of the studied problem. Therefore, they proposed two metaheuristic solution schemes, an NSGA-II and a Reference Point based NSGA-II (RPBNSGA-II). They concluded, through a statistical analysis on the obtained solutions, that NSGA-II provided better solutions than RPBNSGA-II for the case of small-sized instances, while RPBNSHA-II outperformed NSGA-II in large problem cases. Asadi et al. (2018) presented a bi-objective LIRP model for the optimization of production and distribution of a microalgae biofuel SCN. They developed two metaheuristic solution methods, the MOPSO and the NSGA-II. Due to the fact that both of them includes several parameters, authors used the Taguchi method to properly set them. Several problem instances of different size were used to perform a comparison between the proposed methods. NSGA-II proved to be more efficient than MOPSO. Yuchi et al. (2018) studied an LIRP in a closed-loop SCN, which formulated as an MINLP. For the solution of computationally challenging problem instances, a hybrid TS-SA heuristic algorithm was proposed. As a SA-based algorithm includes several parameters, a sensitivity analysis was performed to indicate the potential impact of them on the total costs. A multi-period, multi-commodity LIRP was studied by Vahdani et al. (2018) for the design of a humanitarian logistics network. They developed a MOMIP model to address the problem and two multi-objective metaheuristic algorithms were proposed to solve it efficiently. More specifically, NSGA-II and MOPSO were implemented and tested on several problem cases. In problem cases with certain conditions, MOPSO was proved more efficient than NSGA-II, while the last one worked better under uncertain conditions.

Saif-Eddine et al. (2019) studied an LIRP with Vendor Managed Inventory strategy and developed an improved GA to solve it efficiently. A General Benders Decomposition (GBD) method was proposed by Zheng et al. (2019) to solve an integrated LIRP for the design of a passenger car SCN. Rabbami et al. (2019) investigated a multi-period LIRP in the context of of industrial hazardous waste management. They developed an MINLP model to mathematically formulate the problem and applied an exact linearization method to reduce the initial model to an MILP one. For the solution of medium- and large-sized problem instances a simheuristic, based on NSGA-II and Monte Carlo simulation method, was developed. Numerical tests indicated the efficiency of the proposed method. Furthermore, interesting insights were observed, such those about the relation between the uncertainty levels and the

CPU time requirements. More specifically, an increase on the uncertainty level lead to increase to the required CPU time. Saragih et al. (2019) formulated a three-echelon supply chain system as an MINLP by simultaneously considering location, inventory and routing decisions. They developed a two-stage SA-based heuristic solution method. The first stage consists of a constructive procedure for generating an initial solution, while a SA metaheuristic framework constitutes the improvement stage. For small-sized problem cases, the proposed heuristic produced near-optimal solutions. Authors, also, investigated the impact of tackling inventory decisions within the improvement stage. The obtained results with inventory decisions consideration were 17% better than those achieved without integrating inventory decisions. Chao et al. (2019) developed an MIP model to address a two stage LIRP with Time Windows for a food delivery system. They proposed an Improved Ant Colony Optimization (IACO) algorithm with a distance-based clustering approach. More specifically, the distance-based clustering method was applied to reduce the initial problem into several VRPs. Then, the IACO algorithm was used to for generating high quality routes. Finally, a relocate-exchange method was used to further improve solutions. Biuki et al. (2020) tackled a multi-period LIRP by considering sustainability issues, integrated decision-making and real-world assumptions. They proposed a two-phased approach. Initially, the Multicriteria Decision Analysis (MCDA) method, known as Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE) method was employed to rank the suppliers by a sustainability perspective. Next, an MOMILP was formulated in an effort to further enhance sustainability performance. Due to the high computational complexity of the problem, two hybrid metaheuristic solution methods based on different combinations of a GA and a PSO were implemented for the solution of test instances. Extensive numerical analyses indicated that further improvements on the supply chain performance can be achieved in a cost efficient manner. Table 1.1 summarizes the main LIRP contributions.

TABLE 1.1: Key literature contributions on LIRP

Reference	P.T.[1]	D.T.[2]	C.T.[3]	R.P.[4]	F.C.[5]	E.D.[6]	C.P.[7]	Model	S.M.[8]
(Javid & Azad, 2010)	Single	Stochastic	Single	(Q,R)	Homogeneous	x	x	MINLP	Metaheuristic
(Tavakkoli-Moghaddam et al., 2013)	Single	Stochastic	Single	(Q,R)	Heterogeneous	x	x	MINLP	Exact
(Guerrero et al., 2013)	Multiple	Deterministic/Variable	Single	Order up to level	Homogeneous	x	x	MIP	Metaheuristic
(Seyedhosseini et al., 2014)	Single	Stochastic	Single	(Q,R)	Homogeneous	x	x	MINLP	Exact/Metaheuristic
(Chen et al., 2014)	Single	Fuzzy	Single	(T,R) Periodic	Homogeneous	x	x	MIP	Metaheuristic
(Nekooghadirli et al., 2014)	Multiple	Stochastic	Multiple	(Q,R)	Heterogeneous	x	x	MINLP	Metaheuristic
(Zhang et al., 2014)	Multiple	Deterministic/Variable	Single	Flexible	Homogeneous	x	x	MIP	Metaheuristic
(Liu et al., 2015)	Single	Stochastic	Single	(Q,R)	Homogeneous	x	x	MINLP	Metaheuristic
(Zhalechian et al., 2016)	Multiple	Stochastic	Multiple	(Q,R)	Heterogeneous	x	x	MMINLP	Metaheuristic
(Ghorbani & Akbari Jokar, 2016)	Multiple	Deterministic/Variable	Multiple	(Q,R)	Homogeneous	x	x	MIP	Metaheuristic
(Hiassat et al., 2017)	Multiple	Deterministic/Variable	Single*	-	Homogeneous	x	x	MIP	Metaheuristic
(Rayat et al., 2017a)	Multiple	Stochastic	Multiple	Continuous Review	Heterogeneous	x	✓	BOMIP	Metaheuristic
(Vahdani et al., 2018)	Multiple	Deterministic	Multiple	-	Heterogeneous	x	✓	MOMIP	Metaheuristic
(Rafie-Majd et al., 2018)	Multiple	Stochastic	Multiple	(Q,R)	Heterogeneous	x	x	MINLP	Heuristic
(Asadi et al., 2018)	Single	Stochastic	Sinlge	(S-1,S)	Homogeneous	✓	x	BOMINLP	Metaheuristic
(Habibi et al., 2018)	Single	Stochastic	Single	(S-1,S)	Homogeneous	x	x	MINLP	Metaheuristic
(Tavana et al., 2018)	Multiple	Deterministic	Multiple	Order-based	Heterogeneous	x	x	MOMILP	Metaheuristic
(Saragih et al., 2019)	Single	Stochastic	Single	(Q,R)	Homogeneous	x	✓	MINLP	Metaheuristic
(Saif-Eddine et al., 2019)	Multiple	Deterministic	Single	-	Homogeneous	x	x	MIP	Metaheuristic
(Zheng et al., 2019)	Single	Stochastic	Single	(T,s)	Homogeneous	x	x	MINLP	Exact
(Yuchi et al., 2018)	Single	Stochastic	Single	(Q,R)	Homogeneous	x	x	MINLP	Metaheuristic
(Habibi et al., 2018)	Single	Stochastic	Single	(S-1,S)	Homogeneous	✓	x	MINLP	Metaheuristic
(Forouzanfar et al., 2018)	Multiple	Deterministic	Single	-	Heterogeneous	x	✓	MOMINLP	Metaheuristic
(Rabbami et al., 2019)	Multiple	Stochastic	Multiple	-	Heterogeneous	✓	x	MOMINLP	Metaheuristic
(Chao et al., 2019)	Single	Deterministic	Single	-	Homogeneous	✓	x	MILP	Metaheuristic
(Biuki et al., 2020)	Multiple	Stochastic	Multiple	-	Heterogeneous	✓	✓	MOMILP	Metaheuristic

[1] *Period Type*, [2] *Demand Type*, [3] *Commodity Type*, [4] *Replenishment Policy*, [5] *Fleet Composition*, [6] *Environmental Decisions*, [7] *Capacity Planning*, [8] *Solution Method*

1.2 Sustainable Supply Chain Optimization

Despite SCM traditionally focuses on cost-efficient practices in order to achieve high profit levels and consequently, continuous competitive advantage, the environmental and social dimensions of its activities should be considered (Hristu-Varsakelis et al., 2012; Mallidis et al., 2020; Mallidis et al., 2014). Moreover, due to the fact that the supply chain activities emit pollutants, like carbon dioxide (CO_2), increased socio-environmental concerns have shifted the focus to a balanced goal which integrates economic, environmental and social goals (Foo et al., 2018; Koç et al., 2014). These three dimensions are the major pillars of sustainability.

According to the definition provided by the United Nations in the World Commission on the Environment and Development, sustainability is the effort to achieve the current performance objectives with respect to future generations needs [1]. Obviously, this definition is quite general and cannot provide the sustainable philosophy in clarity.

The sustainable philosophy refers to proper incorporation of economic, environmental and social aspects in the operation of an organization (Iakovou et al., 2016). For instance, the environmental sustainability refers to efforts on reducing the environmental footprint of an organization (i.e. minimize carbon emissions). The social sustainability consists of actions made to guarantee societal requirements, such as the promotion of health or the creation of working capabilities. Moreover, the economic dimension of sustainability refers to cost-efficient application of the proper sustainable operations in order to secure the competitive advantage of the organization. Figures 1.5, 1.6 and 1.7 summarize the most significant indicators of the three dimensions of sustainability (Barbosa-Povoa & Pinto, 2018).

[1] https://sustainabledevelopment.un.org/milestones/wced

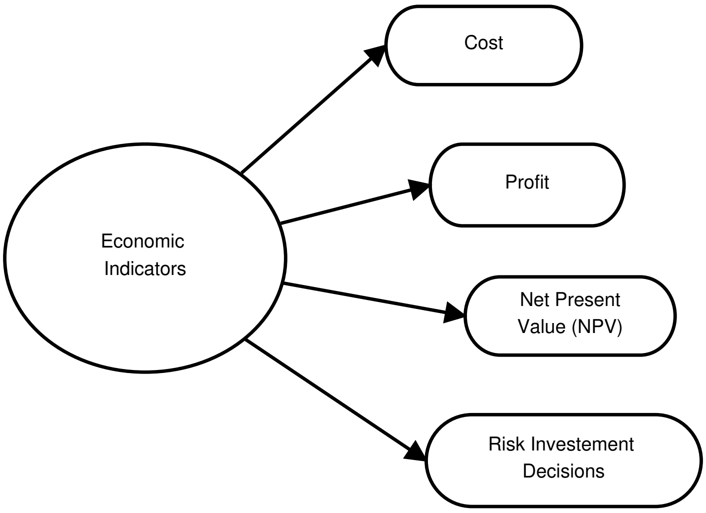

FIGURE 1.5: Economic indicators of sustainability

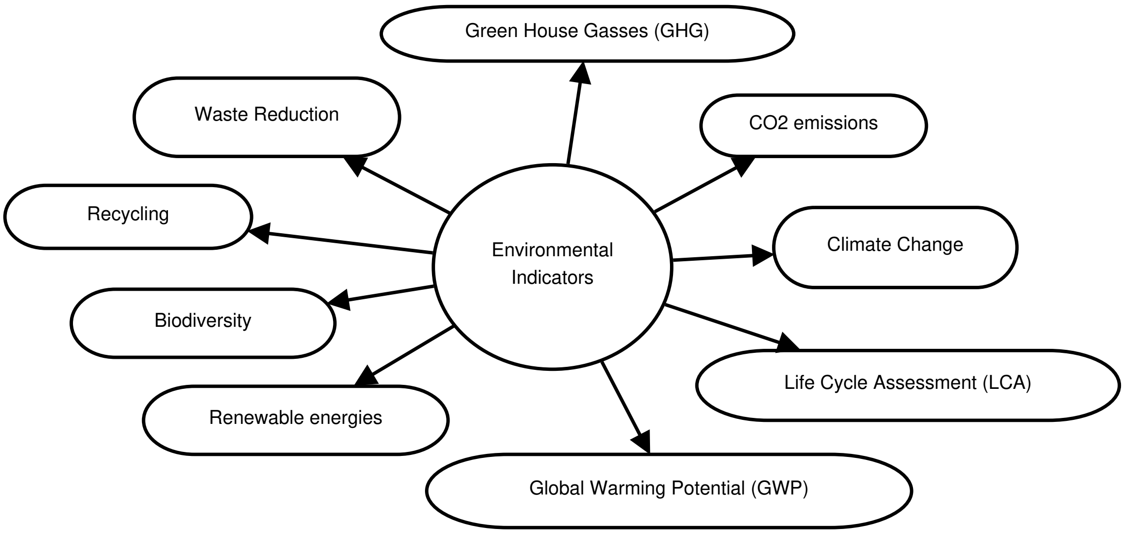

FIGURE 1.6: Environmental indicators of sustainability

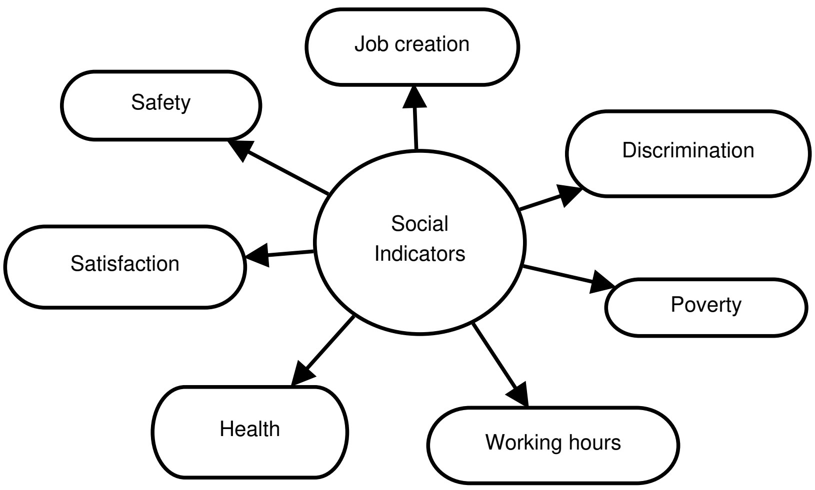

FIGURE 1.7: Social indicators of sustainability

Despite SCM constitutes a crucial element of economic efficiency in an organization, it cannot guarantee sustainable development without considering environmental and social aspects (Hong et al., 2018). To manage the three dimensions of sustainability in perfect balance, a new wider systematic coordination of supply chain activities needs to be addressed (Vivas et al., 2020). This new type of management is known as Sustainable SCM (SSCM) (Fritz, 2019). To successfully achieve sustainable development through SSCM, entails the efficient application of sustainable supply chain practices (Jia et al., 2015). However, it should be clarified that SSCM does not extend the SCM as depicted in Figure 1.8, but it consists of the following three interdependent supply chain management components:

- *SCM*. It deals with the economic aspect of SSCM.
- *Green SCM (GSCM)*. It refers to the environmental aspect of SSCM.
- *Social SCM (SoSCM)*. It deals with the social aspect of SSCM.

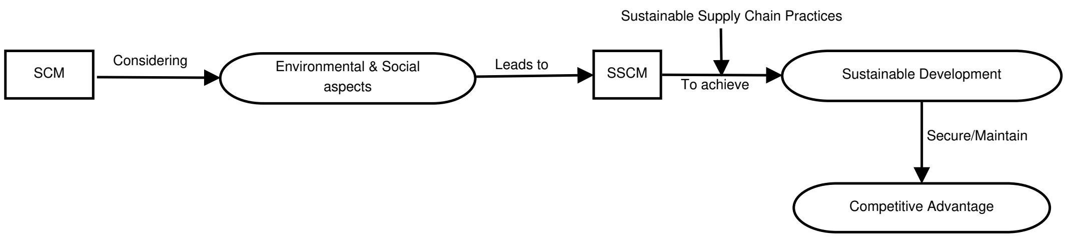

FIGURE 1.8: A false approach of SSCM

The proper approach of the SSCM is illustrated in Figure 1.9.

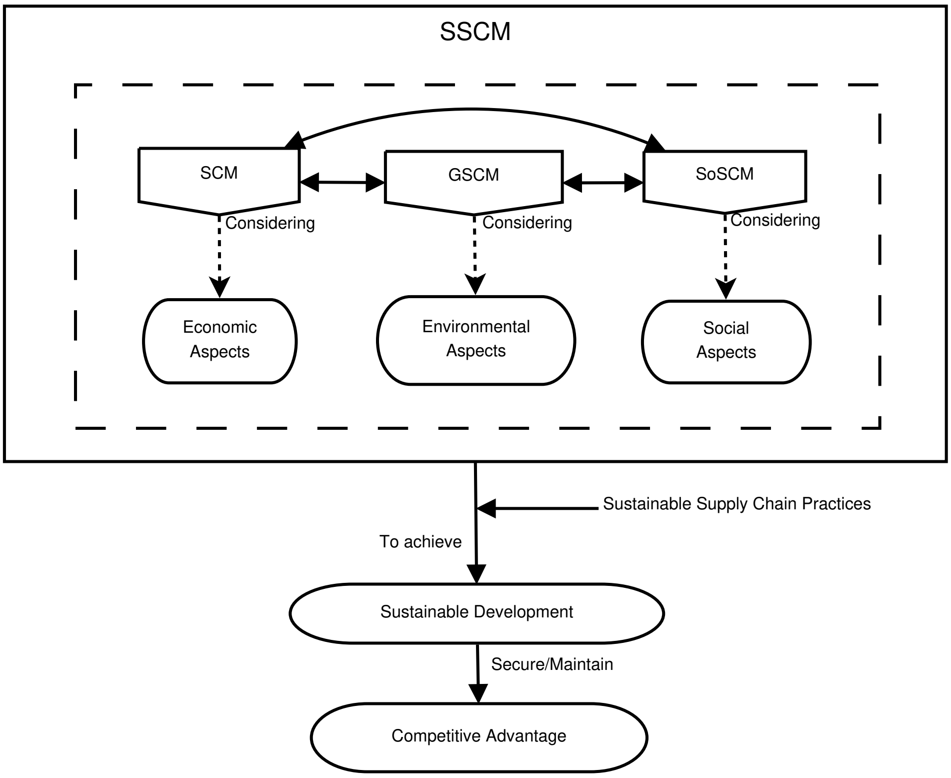

FIGURE 1.9: The SSCM approach

Sustainability can be achieved by the adoption of proper sustainable supply chain practices. These practices aim to address specific characteristics in a supply chain to reform it into a lean, resilient and green system (Govindan et al., 2014). Some examples of such practices are:

- Lean practices
  - Supplier compliance auditing.
  - Total quality management.
  - Just-in-Time (JiT).
- Resilient practices
  - Flexible sourcing.
  - Supply chain risk management.
  - Flexible transportation.
- Green practices
  - Environmental collaboration with suppliers.
  - To decrease the consumption of hazardous and toxic materials.

– Reverse logistics.

Obviously several sustainable supply chain practices can contribute to two or even to all three sustainable characteristics. The impact of several lean, resilient and green SSCM practices on the sustainability of a supply chain has been thoroughly investigated in the recent literature (Govindan et al., 2014; Jia et al., 2015).

From an optimization perspective, problems which tackle environmental-related decisions are characterized as green optimization problems (Bektaş et al., 2019; Martins & Pato, 2019; Rafie-Majd et al., 2018; Skouri et al., 2018). Freight transportation is mentioned as the main source of CO_2 (Bektaş et al., 2019; Leenders et al., 2017). Especially, road transportation generates more than 20% of the total CO_2 emissions in European Union (Leenders et al., 2017) and 30% in the Organization for Economic Co-operation and Development (OECD) countries (Reichert et al., 2016). Therefore, the majority of previous works in this area has focused on green routing optimization problems (Li et al., 2018; Soon et al., 2019; Yu et al., 2019b).

However, as recently noticed by Koç et al. (2016) depot- and fleet composition-related decisions also affect emissions. In this direction, several contributions have studied more complex supply chain optimization problems within environmental considerations. Dukkanci et al. (2019) addressed a green LRP. They used a comprehensive modal emission model in order to estimate the emitted pollutants. Zhang et al. (2018) studied a multi-depot emergency LRP with carbon dioxide emissions. Cheng et al. (2017) proposed a green IRP with fleet heterogeneity. They highlighted the benefits of using a mixed fleet. Toro et al. (2017) studied the multi-objective green LRP and they highlighted the importance of using more vehicles in shorter routes to minimize both fuel consumption and emissions. Micheli and Mantella (2018) studied an environmentally extended IRP with heterogeneous fleet and they examined the effect of different carbon control policies on emissions reduction. Even though the environmental-related decisions are critical in achieving sustainability, limited contributions of green LIRP cases have been reported. More specifically, a MOMINLP model for the closed-loop LIRP was proposed (Zhalechian et al., 2016). They used a stochastic-possibilistic approach in order to tackle uncertainty and they developed a hybrid self-adaptive GA-VNS metaheuristic algorithm to solve large-sized instances. A bi-objective stochastic LIRP, which considers CO_2 emissions at distribution facilities, was proposed by Asadi et al. (2018).

1.3 Cell Therapies Supply Chain Optimization

Healthcare Supply Chains

Healthcare or medical supply chain refers to the flow of medical products and services between several independent parts, such as hospitals, drug manufacturers and patients (Imran et al., 2018). An illustration of a typical healthcare SCN (hSCN) is provided in Figure 1.10.

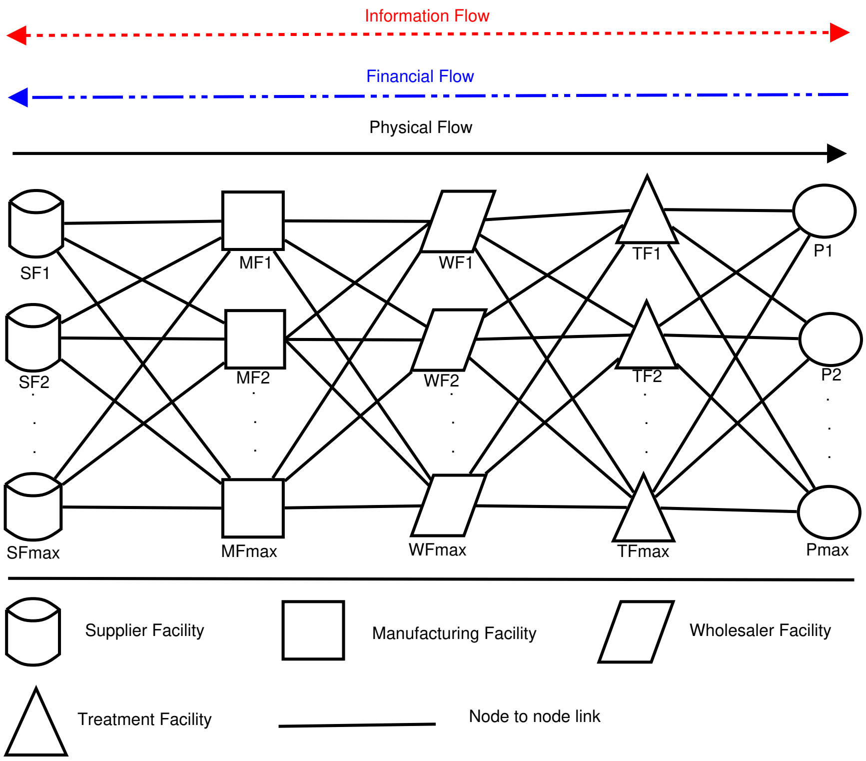

FIGURE 1.10: An example of an hSCN

The management of hSCN includes the activities, required for the efficient tackling of physical, information and financial flows.

The physical flow may contain (Imran et al., 2018):

- pharmaceuticals,
- surgical consumables,
- medical devices,
- hygiene consumables.

Some examples of information flow are the following (Gonul Kochan et al., 2018):

- Order entry and processing data,
- patients' medical information,
- inventory information,
- pricing data.

The financial flow commonly involves:

- payment schedules,
- credit terms,
- consignment agreements.

Despite both physical and information flows can be clearly understood to the reader, further details may be needed for the examples of financial flows due to their technical terminology. The payment schedules are typical payments for purchased products and services. Credit terms are payment requirements which are declared on invoices according to early payment terms (Li et al., 2019). An invoice is a formal financial document which summarizes identification-oriented data of a transaction such as transaction number, seller and buyer details, quantities of items and payment amount and terms. Consignment agreement is a contract signed by a seller, the consignor, and a buyer, the consignee. According to that signed agreement, the consignor delivers products to consignee but their title remains with the consignor. For the sold or used products, the consignee executes a payment, as denoted by a commission. Non- sold or used products can be returned to the consignor (Gharaei et al., 2019). This financial agreement can be adopted by healthcare entities, especially for the case of pharmaceuticals supply as a tool for handling demand variability.

Cell Therapies

Biopharmaceuticals constitute sophisticated medical products with pure biological basis, such as cells, bacteria and enzymes (Holder et al., 2019). The main scope of biopharmaceutical industry is to handle the recent advances in fields of bioengineering and biomanufacturing, in order to control and guide the behavior and function of a living cell (Xie & Murphy, 2019). The branch of medicine which utilizes this new advanced pharmaceuticals is known as biomedicine or more commonly as regenerative medicine (Gardner & Webster, 2016).

Cells are critical components of those new products. The cellular produced therapies are classified into (Wei Teng et al., 2014):

- **Allogeneic**. In these therapies, different donor and recipient are considered.
- **Autologous**. Cells are removed from a patient, re-engineered and returned to the same patient.

To better understand the essential difference between the two classes of cellular therapies, a comparative illustration is provided in Figure 1.11.

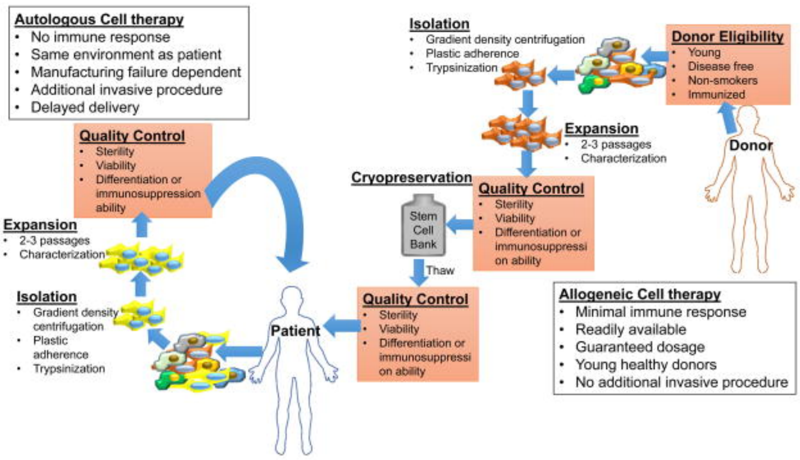

FIGURE 1.11: A comparison between autologous and allogeneic cellular therapies (Karantalis et al., 2015)

Malik and Durby (2015) attempted a more detailed comparison between allogeneic and autologous cell therapies on the base of the following factors:

- Immunological issues,
- patient-centric factors,
- commercial-scale manufacturing,
- business model,
- reimbursement price potential.

Some of the key differences, according to the mentioned factors, are:

Immunological issues. An allogeneic cell therapy entails a higher risk of immune rejection than an autologous cell therapy. A potential immunological rejection may cause a significant limitation of the persistence of administered cells and consequently to eliminate any possible therapeutic impact, or an inflammatory reaction which leads to patient tissue damage (Malik & Durby, 2015; Mo et al., 2019).

Patient-centric factors. In case of allogeneic cell therapies, the cells can be donated by selected young and healthy adults. On the other hand, in autologous therapies cells taken from older patients have lower proliferative capacity, which may negatively affect the therapeutic benefit (Narbonne, 2018; Scruggs et al., 2013).

Commercial-scale manufacturing. The cell-based products, such as the cell therapies have higher manufacturing costs compared to other biopharmaceuticals (Malik & Durby, 2015). Focused on the two classes of cell therapies, the personalized manufacturing character of autologous cell therapies thwarts their mass production and increases their manufacturing cost. Moreover, in allogeneic approach a potential damage of the product can be restored in significantly shorter time than in the autologous therapies (Morizane et al., 2013).

Business model. Allogeneic cell therapies follow the well-known "Off-the-Shelf" production model, which refers to non-customized products. Therefore, they properly fit in the current model of biomanufacturing industry. Also, they can be applied both in acute and chronic diseases (Malik & Durby, 2015). In contrast, the autologous cell therapies are restricted to a product-customized model and applied to non-acute diseases (Morizane et al., 2013).

Reimbursement price potential. The worldwide financial crisis leads to significant limitations on the healthcare budgets. Thus, securing reimbursement is almost crucial in the successful early life-cycle of a medical product. For both allogeneic and autologous therapies, the insurer is expected to pay a high reimbursement, especially in case of life-threatening diseases (Malik & Durby, 2015; Pereira Chilima et al., 2018).

Chimeric antigen receptor (CAR) T-cell therapy

According to the American Cancer Society and National Cancer Institute, immunotherapy represents a biological cancer therapy, which can either enhance the immune system of a patient or lead to the training of the immune system in order to attack specific cancer cells [2], [3]. Chimeric antigen receptor (CAR) T-cell therapy is a promising new type of immunotherapy (Baboo et al., 2019). The pillar of this cellular therapy is T-cells, which are re-engineered in specialized laboratories in order to be able to identify and destroy cancer cells. The re-engineering process is achieved by performing critical steps (Vormittag et al., 2018). In the first step, blood is removed from the patient and through leukapheresis, separation of leukocytes is carefully performed and the rest of the blood is returned to the patient. This initial process is completed when a sufficient number of leukocytes is collected. Next, anticoagulants are added in the leukapheresis buffer and a washing step is performed by removing the remaining red blood cells and platelets followed by T-cell selection, which separates cells based on their density. The next step includes activation of T-cells by using soluble monoclonal antibodies (anti-CD3/anti-CD8), coated magnetic beads or artificial autologous antigen-presenting cells (APCs). CAR is then delivered to T-cells by using either viral or non-viral methods (Vormittag et al., 2018). Following gene delivery, expansion of CAR T-cells is required, which can take up to few weeks since millions of CAR T-cells need to be generated for each therapy. The expansion process is performed in different culture platforms (T-Flasks, Static Culture Bags, Rocking Motion Bioreactors). Finally, the appropriately CAR T-cells are either immediately administered, through infusion, to patients or are frozen. For clarity for the reader, both the overall CAR T-cell manufacturing and administration process and its major steps are illustrated in Figures 1.12 and 1.13.

[2]https://www.cancer.org/treatment/treatments-and-side-effects/treatment-types/immunotherapy/car-t-cell1.html#written_by

[3]https://www.cancer.gov/about-cancer/treatment/research/car-t-cells

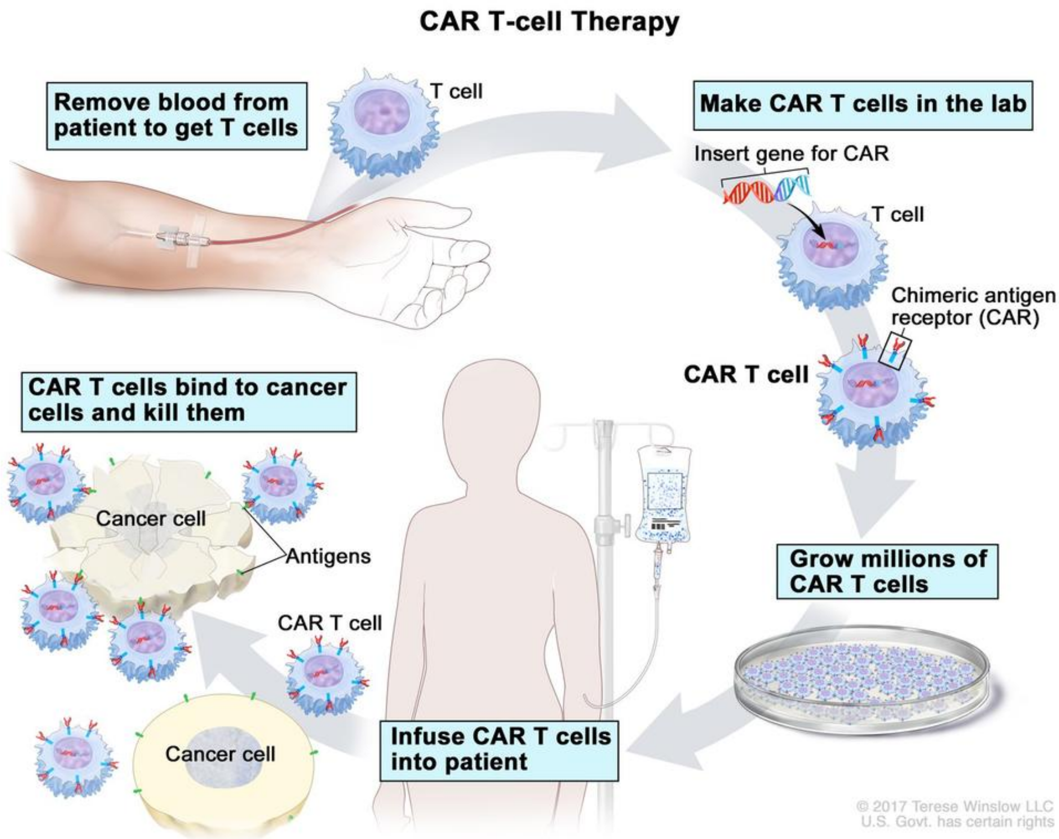

FIGURE 1.12: The overall process of CAR T-cell therapy
source : National Cancer Institute[4]

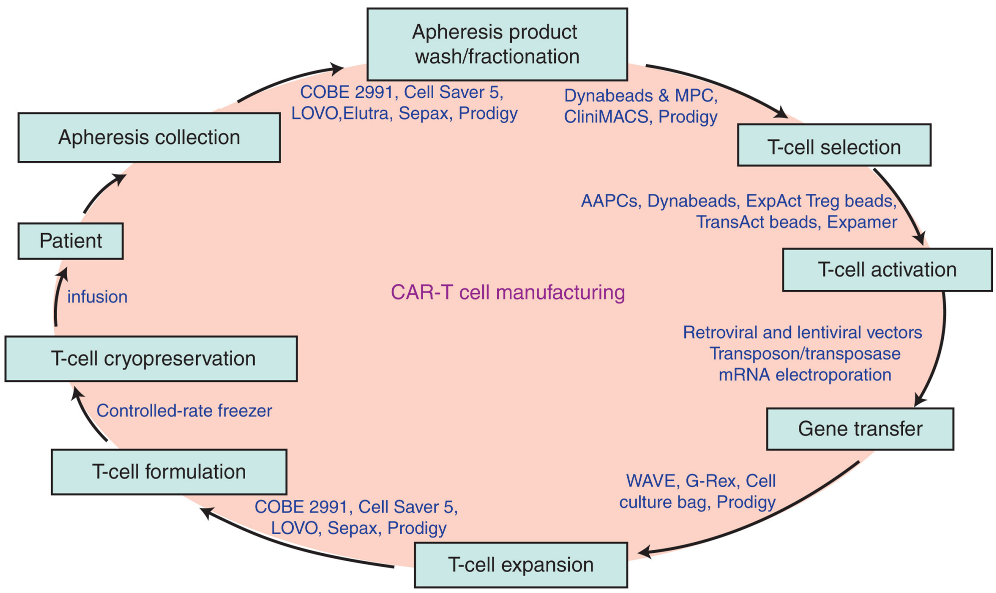

FIGURE 1.13: Key steps in CAR T-cell therapy manufacturing
(Wang & Riviere, 2016)

[4]https://www.cancer.gov/about-cancer/treatment/types/immunotherapy/t-cell-transfer-therapy

Currently, very limited attention has been placed on the design and operation of CAR T-cell therapies supply chains, while the network representation is still an open research challenge. The overall process of CAR T-cell therapies relies on four main entities (Boyiadzis et al., 2018). A cancer patient visits a specialized treatment facility and a blood sample is collected. Then, the collected blood is sent to the manufacturing centre where the therapy is produced. Finally, the patient revisits the treatment facility to receive the required bridge chemotherapy (a chemotherapy administered in the time between collection of autologous T-cells and the infusion of CAR T-cell therapy) and the cell therapy is delivered to the treatment facility in order to be administered to the patient.

The complex and precise biomanufacturing of CAR T-cells requires highly-specialized staff, facilities and equipment (Harrison et al., 2019), which result in high costs associated with CAR T-cell therapy. Specifically, 2019 reports [5] place cost of CAR T-cell therapy to $375000. Certain treatments, such as Novartis' Kymriah cost up to $475000 (Vormittag et al., 2018). A major cost contributor is logistics, such as facilities selection costs, consumables, and transportation costs (Harrison et al., 2019). As such, supply chain costs represent approximately 30% of the total cost of treatment (Franco & Alfonso-Lizarazo, 2019). Consequently, SCO is crucial in reducing costs and rendering the design and operation of the CAR T-cell therapies SCN financially sustainable and competitive (Barbosa-Povoa & J.M., 2019; Numbi & Kupa, 2017).

Although significant progress has been made in the optimal design and operation of traditional supply chain production and distribution networks, the integrated CAR T-cell supply chain network problem has not been studied in the open literature. A couple of recent contributions have solely focused on studying specific aspects of the underlying supply chain problem of CAR T-cell therapies without considering the integrated optimization of design and operational aspects (Wang et al., 2019; Wang et al., 2018).

A multi-objective stochastic programming model for the optimal network design of CAR T-cell supply chain was recently proposed by Wang et al. (2018). They highlighted the benefits that can be achieved by optimizing several design aspects of these supply chains. Wang et al. (2019) presented a multiscale logistics simulation framework to address emergent challenges of autologous cell therapies. They investigated two scenarios; the first, focused on the selection of a proper inventory policy, while the second one

[5] https://www.reuters.com/article/us-gilead-novartis-trials-focus/gilead-novartis-cancer-therapies-losing-patients-to-experimental-treatments-idUSKCN1UP10O

evaluated the impact of a possible supplier disruption. Papathanasiou et al. (2020) recently presented a review of the main challenges associated with the CAR T-cell supply chains, such as the capacity bottleneck in hospitals. They thoroughly highlighted several issues which affect the efficient design and operation of the underlying supply chain network. Harrison et al. (2019) used a process economics modelling and calculation tool to investigate the impact of different factors on the production costs of CAR T-cell therapies (donor cells, geographic dispersion of production and consumption points, etc)

CAR T-cell supply chains share key characteristics with other process supply chains, such as the vaccine supply chain and the blood supply chain. Some of these characteristics are as follows:

- management of medical-oriented systems,
- distribution of temperature-sensitive products,
- strict lead times,
- strict replenishment rates.

Healthcare & Pharmaceutical Supply Chains.

Liu et al. (2014) proposed a periodic vehicle routing problem for home healthcare logistics in which deliveries, either of special drugs from hospitals to patients or blood samples from patients to labs, were evaluated. They developed a tabu search metaheuristic algorithm by incorporating feasible and infeasible intra-route local search schemes for the solution of several logistics problems. Zahiri et al. (2017) studied an integrated sustainable-resilient pharmaceutical supply chain under uncertainty. They formulated the problem as a multi-objective mixed-integer linear programming model and proposed a possibilistic-stochastic programming approach to handle uncertainty. Moreover, a game-based differential evolution-variable neighborhood search metaheuristic algorithm was developed for the solution of large problems. Savadkoohi et al. (2018) developed a possibilistic location-inventory model to optimize a pharmaceutical supply chain by evaluating the supply chain network design under perishability issues. Jankauskas et al. (2019) presented a genetic algorithm (GA) for solving the integrated problem of capacity planning and scheduling of a biopharmaceutical manufacture. They also, developed a particle swarm optimization as a post-optimization procedure for fine tuning GA's hyperparameters in an attempt to solve even larger problem cases than the developed GA. Kramer et al. (2019) addressed a variant of vehicle routing problem for delivering pharmaceutical products to healthcare facilities. They developed a multi-start iterated local search algorithm solving of both realistic and artificial problem cases. Moussavi et al. (2019) proposed a Gurobi-based matheuristic to solve integrated worker assignment and vehicle routing problems of a home healthcare system.

Blood & Vaccine Supply Chains.

Sadjadi et al. (2019) studied a vaccine supply chain network design problem under uncertainty. They proposed a deterministic MILP and a robust mathematical programming approach applied to a real-world case study, in order to evaluate performance. Lin et al. (2020) focused on a vaccine supply chain in which vaccines are delivered from a distributor to a hospital. They examined the distributor's decision-making on using or not temperature-controlled transportation and the potential impact of the retailer's inspection policy on that decision. A reliable blood supply chain network design with facility disruption was recently studied by Haghjoo et al. (2020). They proposed a scenario-based robust approach to address uncertainty and developed two metaheuristic algorithms for the solution of large-sized problems.

1.4 Optimization Techniques

1.4.1 Optimization Problem

An optimization problem has potentially the following general form:

$$\min\{f(x)|x \in X \subseteq S\} \tag{1.1}$$

where S denotes the solution space, X is the set of feasible solutions -solutions which respect a number of given constraints, x represents an obtained feasible solution and f is an objective function.

A classification of optimization problems can be performed by considering the nature of elements in set S. More specifically, an optimization problem is characterized as continuous when S contains real values and as discrete when S includes discrete values, such as integers (Baron et al., 2019; Kidd et al., 2020).

Combinatorial Optimization (CO) refers to discrete optimization problems where the discrete set of solution space is finite, commonly huge and contains combinatorial structures, such as permutations or assignments (Goerigk et al., 2020). Typically, a CO problem is mathematically formulated as an MILP model, which means that the model includes both variables receiving values in an integer and a continuous domain (Della Croce, 2014).

An alternative taxonomy classifies the optimization problems based on their requirements on computational resources needed for their solution, known as computational complexity. The computational time is one of the most critical computational resource (Hoos & Stützle, 2005). Therefore, following a time complexity taxonomy, some crucial complexity classes are met:

- Complexity class P (Polynomial). *This class includes problems which can be solved in polynomial time.*
- Complexity class NP (Non-deterministic Polynomial). *It contains problems which requires commonly exponential time to be solved, but their solutions can be verified in polynomial time.*
- Complexity class $NP - hard$ (Non-deterministic Polynomial-hard). *Problems in this class are at least as difficult to be solved as the most difficult problems in NP, but they do not necessarily belong to NP.*

It is obvious that class P is a subset of class NP, as each problem which can be solved polynomially, its solution can also be verified in polynomial solution. Another significant subset of class NP is the class $NP - complete$ (Non-determinitistic Polynomial-complete). This class contains problems which they belong both in NP and $NP - hard$. Consequently, those problems included in $NP - complete$ are the hardest problems in NP (Huang et al., 2009).

1.4.2 Solution Methods

An optimization method is a technique used for solving an optimization problem (Zgaya & Hammadi, 2016). Optimization methods are divided into four main categories:

- Exact algorithms.
- Approximation algorithms.
- Heuristic methods.
- Metaheuristic methods.

Exact algorithms are pure mathematical techniques for solving optimization problems with proved optimality. Some well-known exact methods are *Enumeration*, *Branch & Bound algorithm* and *Branch & Cut* (Festa, 2014). Enumeration method investigates and compares all possible feasible solutions. Branch & Bound methods examine all possible solutions by exploring the solutions of relaxed subproblems in a tree structure. They are characterized by two key processes, the branching which applied to generate subproblems and bounding, which uses rules for pruning potential low quality branches. Branch & Cut algorithms are a combination of Branch & Bound thechniques with an iterative method which refines the search space, known as cutting-planes method. This iterative use of cutting-planes aims to reduce the search space. **Approximation algorithms** are solution methods which provide near optimal solution with a guarantee of their performance e.g. a found solution has a cost 1.5 times of the optimal one. *Dynamic programming methods* and *Relaxation methods* are two of the most commonly used categories of approximation solution methods (Williamson & Shmoys, 2011). Dynamic programming technique divide a given problem into simpler subproblems. These problems are solved in a bottom-up approach by combining their solutions. Lagrangian relaxation is a well-known relaxation method in which complex constraints can be removed by being transferred to the objective function, weighted with a proper Lagrange multiplier. Despite exact methods guarantee the optimality of best found solutions, limited problems can be tackled in optimality in reasonable time. Therefore, the development of approximation algorithms is an effort to manage intractable problems in accepting computational time, by sacrificing optimality. However, hard optimization problems, such those in $NP - hard$ complexity class, is quite challenging to be solved either by exact or approximation algorithms. Particularly, the solution of large scale instances of $NP - hard$ problems requires the development of fast and efficient computational methods.

Heuristics are solution approaches for obtaining good-quality solutions in significantly short computational times without any optimality guarantee (Zgaya & Hammadi, 2016). Heuristic solution methods are divided into two categories:

- *Construction heuristics*, which build feasible solution for a specific optimization problem. These methods usually are quite fast and simple procedures for the solution of a problem (Blocho, 2020).

- *Improvement heuristics* are iterative procedures which receive an already constructed solution and further improve it. Well-known local search operators, such as 2-Opt or 1-1 node exchange, are considering as classic examples of improvement heuristics (Blocho, 2020).

Metaheuristics are general frameworks, which imply specific strategies in order to perform a deeper exploration of the search space than simple heuristics (Deroussi, 2016). The solutions achieved by using a metaheuristic solution method are often better than those obtained by simple heuristics. Furthermore, metaheuristics are classified into trajectory-based (also known as individual-based) and population-based methods. The first category includes methods which tackle a single solution in each step of their solution process, while the second one contains methods that they tackle a number of solutions simultaneously (Deroussi, 2016). Typically, metaheuristics consist of two main steps, the *intensification* and the *diversification* steps. Intensification refers to the exploitation in specific locations in the search space which seems to be quite promising. On the other hand, diversification is an exploration step in an effort to shift in new promising regions of search space (Blocho, 2020). An appropriate balance between these two strategies is essential, as it significantly affects the performance of a metaheuristic solution method. To successfully structure an efficient metaheuristic scheme, a deep knowledge of the tackled problem is required (Deroussi, 2016).

Hansen and Mladenović (2003) underlines that a metaheuristic algorithm should be characterized by the following properties:

1. **Simplicity**. This feature refers to the idea behind the design of a metaheuristic algorithm, which should be simple and comprehensive.

2. **Precision**. A metaheuristic algorithm should be clearly defined in mathematical terms, which means that redundant details should be eschewed.

3. **Coherence**. The design of a problem-specific heuristic based on a metaheuristic framework should follow the principle of this framework in each step.

4. **Efficiency**. A metaheuristic algorithm should be able to provide optimal or near-optimal solutions for instances of a specific problem.

5. **Effectiveness**. A metaheuristic algorithm should provide a high-quality solution in a reasonable computing time.

6. **Robustness**. The performance of a metaheuristic solution method should be stable solving different instances of a specific problem.

7. **User-friendliness**. The designed metaheuristic algorithms should be intelligible and well-expressed.

8. **Innovation**. The principle of the metaheuristic algorithm should provide the chance of innovative applications.

1.4.3 Variable Neighborhood Search solution framework

VNS proposed by Mladenović and Hansen (1997), is a trajectory-based metaheuristic which has all the desired properties mentioned in the previous section. Its main principle is the systematic change of predefined local search operators (also known as neighborhood structures) during the search for an optimal or approximately optimal solution (Hansen et al., 2010). This systematic process is applied as a repeated execution of three basic search ingredients, until a stopping criterion is met. These three search steps are (Hansen et al., 2017):

- Shaking Procedure (as a diversification phase for escaping locally optimal solutions).
- Neighborhood Change Step (for guiding purposes while VNS explores the solution space).
- Improvement Procedure (as an intensification phase for improving the incumbent solution).

Figure 1.14 provides an illustration which indicates how the use of different neighborhoods and their successive application lead to efficient exploration of the search space.

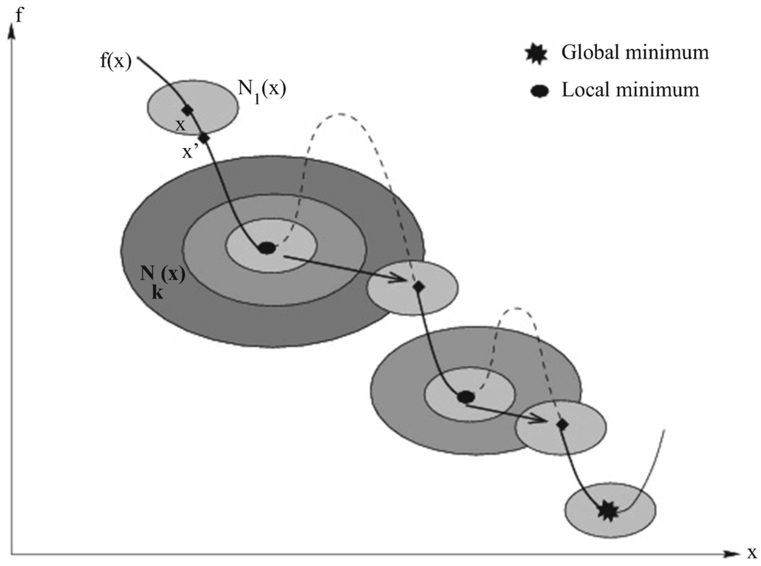

FIGURE 1.14: The application of VNS in the search space (Hansen & Mladenović, 2018)

The most commonly used diversification method within VNS is the intensified shaking, which randomly selects a shaking operator and applies it k times, where k denotes the intense of diversification and it is $1 \leq k \leq k_{max}$, with k_{max} being the shaking strength (Papalitsas et al., 2019). According to the neighborhood change step, several procedures have been proposed. Their aim is the guidance of the VNS algorithm during the exploration in the solution space (Hansen et al., 2010). Four of the most commonly used neighborhood change procedures are the following (Hansen et al., 2017):

- **Sequential neighborhood change step**. The search continuous to the next neighborhood when no improvements occurred within the current neighborhood. In case of reaching an improved solution, the search continues with the first neighborhood structure.

- **Cyclic neighborhood change step**. In this neighborhood change strategy the search continues to the next neighborhood regardless of any improvement.

- **Pipe neighborhood change step**. The search continues in the same neighborhood, while it produces improvements.

- **Skewed neighborhood change step**. This neighborhood change step aims to lead the search in distant locations in the search space regarding the incumbent solution. To achieve that, they accept not only improving new solutions, but also solutions which are worse than the current one. This relaxed acceptance of solutions is denoted by a relative parameter.

The combination of simplicity and efficiency, which are characterized VNS, has attracted the attention of researchers. Thus, several variants of this metaheuristic framework have been addressed in the literature. Three of the most well-known VNS schemes are the Basic VNS (BVNS), the Variable Neighborhood Descent (VND) and the General VNS (GVNS) (Brimberg et al., 2017; Mladenović et al., 2020; Sánchez-Oro et al., 2017).

BVNS alternates a shaking procedure with a local search operator, until a stopping criterion is met (Hansen et al., 2010). A commonly used termination criterion is an upper CPU time limit (Hansen et al., 2017). Several interesting contributions using the BVNS have been proposed in the recent literature (Costa et al., 2017). An illustration of BVNS is provided in Algorithm 1. BVNS receives an initial feasible solution S, the strength of shaking k_{max}, the upper CPU execution time limit max_time, the ordered set of local search operators N and the set of shaking operators $N_{Shaking}$.

Algorithm 1 Basic VNS

```
procedure BVNS(S, k_max, max_time, N, N_Shaking)
    l ← 1
    while time ≤ max_time do
        for each neighborhood structure l do
            for k ← 1, k_max do
                S' ← Shake(S, k, N_Shaking)
                S'' ← Local_Search(S', l)
                if f(S'') < f(S) then
                    S ← S''
                    l ← 1
                else
                    l ← l + 1
                end if
                if l > |N| then
                    l ← 1
                end if
            end for
        end for
    end while
    Return S
end procedure
```

VND is fully deterministic VNS variant in which a number of local search operators are applied iteratively with respect to an adopted neighborhood change strategy (Hansen et al., 2017; Mjirda et al., 2017). According to the neighborhood change strategy, the following sequential VND schemes are formed:

- **Basic VND (bVND)**. Each time an improved solution is found, the search continues with the first operator (Hansen et al., 2017).
- **Pipe VND (pVND)**. If an improved solution is found within an operator, the search continues with that operator (Hansen et al., 2017).
- **Cyclic VND (cVND)**. The search continues with the next operator regardless the improvements (Hansen et al., 2017).
- **Union VND (uVND)**. It is also known as Multiple neighborhood search. The search is applied in the union of all neighborhood structures (Hansen et al., 2017).
- **Extended VND (eVND)**. This VND variant extends bVND by specifying a parameter (m) which indicates the improvement depth. More specifically, the search switches to the first operator either when m improvements are achieved by the current operator or exactly one improvement is made within the current operator (Lai & Hao, 2016).
- **Adaptive VND (aVND)**. This variant uses one of the previous VND schemes but in each iteration the order of the neighborhoods is changed mainly according to their success in the previous iteration (Todosijević et al., 2016).

VND has been successfully applied on many combinatorial optimization problems related to logistics activities (Larrain et al., 2017; Sifaleras & Konstantaras, 2017). A VND method receives an initial feasible solution S and an ordered set of neighborhood structures N as inputs. The steps of a pVND solution method are provided in Algorithm 2.

Algorithm 2 pipe-VND

```
procedure PVND(S, N)
    l = 1
    while l ≤ l_max do
        select case(l)
        case(l)
        S′ ← N_l(S)
        end select
        if f(S′) < f(S) then
            S ← S′
        else
            l = l + 1
        end if
    end while
    Return S
end procedure
```

GVNS is a widely used VNS variant, which extends BVNS by using a VND method as its main improvement phase (Hansen et al., 2017). GVNS combines a deterministic with a stochastic phase. More specifically, it consists of the stochastic shaking part of the BVNS and a deterministic VND procedure. Therefore, GVNS constitutes a powerful solution scheme which has been used to solve NP-hard problems (Derbel et al., 2019; Sifaleras & Konstantaras, 2015, 2018; Smiti et al., 2020; Todosijević et al., 2017). The main form of a GVNS solution method is provided in Algorithm 3. The inputs of this method are an initial feasible solution S, the shaking strength k_{max}, an upper CPU time limit max_time, the ordered set of local search operators N and the set of shaking operators $N_{Shaking}$.

Algorithm 3 General VNS

```
procedure GVNS(S, k_max, max_time, N, N_Shaking)
    while time ≤ max_time do
        for k ← 1, k_max do
            S* = Shake(S, N_Shaking)
            S' = pVND(S*, N)
            if f(S') < f(S) then
                S ← S'
            end if
        end for
    end while
    return S
end procedure
```

1.5 Research Objectives and Thesis Outline

The main research objectives of this dissertation, which are summarized as follows:

- Modeling extensions of integrated supply chain optimization problems, by considering further realistic aspects, to address new problems of industrial and practical interest.
- The development of powerful metaheuristic algorithms for the efficient solution of large-scale supply chain problems.
- To derive managerial insights based on the optimization of the underlying supply chain problems.

The rest of the thesis is structured as follows:

- **Chapter 2**: A new complex SCN optimization problem is addressed. It extends the well-known LIRP by considering distribution outsourcing decisions.
- **Chapter 3**: A green variant of the LIRP is introduced. A comprehensive emission model is adopted to address fuel consumption and CO_2 emissions in the operation of this supply chain problem.
- **Chapter 4**: A new LIRP variant is studied by considering further advanced strategic decisions, such as fleet composition and capacity planning. Moreover, the Just-in-Time replenishment policy is adopted as it

is considered as the most appropriate approach for many emergency SCN, such as the hSCN.

- **Chapter 5**: A new modelling framework and an efficient solution approach for the optimization of CAR T-cell therapies supply chain are developed. A novel patient-centric supply chain structure is proposed, as the administration of CAR T-cell therapies is performed in local treatment facilities located close to patients' sites.

- **Chapter 6**: This chapter encapsulates the key research findings of the present dissertation and discusses potential future research directions.

Chapter 2

Optimization of Location-Inventory-Routing Problems with Distribution Outsourcing

2.1 Introduction

Several companies understood the importance of the strategic relationships and started adopting logistics outsourcing as a key strategic component, in order to increase their competitiveness (Hjaila et al., 2016; Turkay et al., 2004). Cost reduction, decreased service times and improved customer service are considered as the main advantages of logistics activities outsourcing in the literature (Basligil et al., 2011; Zhu et al., 2017). Because of the crucial effect of the decisions integration and activities outsourcing on the performance of the supply chain, the combined study of these components seems to be highly promising.

This work introduces the Location-Inventory-Routing Problem (LIRP) with Distribution Outsourcing (LIRPDO) decisions. The underlying problem variant represents a more realistic situation, in which a company needs to outsource its distribution operation, as it cannot afford vehicles acquisition or a customer-specific fleet of vehicles is required. Then, more decisions should be made, such as the selection of the proper vehicles providers and the most efficient allocation of the company's opened depots to the selected providers. The proposed problem is NP-hard, which means that realistic large-sized problem instances cannot be solved by exact methods. Therefore, a Sequential General Variable Neighborhood Search (GVNS) combined with an Inventory Rescheduling Procedure (InvRP) for solving large instances of LIRPDO is proposed.

2.2 Problem statement

The LIRPDO is defined as a three echelon supply chain network with multiple potential vehicles' providers, multiple potential depots and a number of geographically dispersed customers. Each customer has a deterministic period-variable demand of one type commodity. It is also assumed that all potential vehicles' providers own the same type capacitated vehicles, but each of them has a different fixed-contract cost. A customer can be allocated to exactly one opened depot, and each opened depot can be served by exactly one vehicles' provider over the planning horizon. A vehicle is be sent from the location of its provider to the selected depot, in order to load the necessary quantity of product and then will travel through the customers allocated to its route. Finally, the vehicle will return to the location of its owner. Therefore, the routes are formed as provider-depot-customer(s)-provider. The objective in this problem is to minimize the total cost including of location, inventory, routing and outsourcing service costs.

2.3 Mathematical formulation

The proposed MIP extends the mathematical model proposed by Zhang et al. (2014) by considering distribution outsourcing decisions. To address those decisions, a new set of binary decision variables $PDA_{b,j}$ and a cost component for selecting vehicle providers, $\sum_{b \in B} \sum_{j \in J} PDA_{bj} * fp_b$, are considered. Also, new constraints 2.5-2.9 have been introduced in the original model.

For the sake of the reader clarity all model sets, parameters and variables contained are summarized in Tables 2.1, 2.2, and 2.3, respectively.

TABLE 2.1: Sets of the mathematical model

Indices	**Explanation**
V	set of nodes
J	set of candidate depots
I	set of customers
K	set of vehicles
H	set of discrete and finite planning horizon
B	set of vehicles' providers

TABLE 2.2: Parameters of the mathematical model

Parameter	Explanation
f_j	fixed opening cost of depot j
fp_b	fixed-contract cost of selecting provider b
C_j	storage capacity of depot j
h_i	unit inventory holding cost of customer i
Q_k	loading capacity of vehicle k
d_{it}	period variable demand of customer i
c_{ij}	travelling cost of locations pair (i, j)
VA_k	the ownership of vehicle k

TABLE 2.3: Decision variables of the mathematical model

Variable	Explanation
y_j	1 if j is opened; 0 otherwise
z_{ij}	1 if customer i is assigned to depot j; 0 otherwise
x_{ijkt}	1 if node j is visited after i in period t by vehicle k
q_{ikt}	product quantity delivered to customer i in period t by vehicle k
w_{itp}	quantity delivered to customer i in period p to satisfy its demand in period t
$PDA_{b,j}$	1 if depot j is served by provider b; 0 otherwise

$$\min \sum_{j \in J} f_j y_j + \sum_{i \in I} h_i \sum_{t \in H} \left(\frac{1}{2} d_{it} + \sum_{p \in H, p<t} w_{itp} \, (t-p) + \sum_{p \in H, p>t} w_{itp} \, (t-p+|H|) \right) + \sum_{i \in V} \sum_{j \in V} \sum_{t \in H} \sum_{k \in K} c_{ij} x_{ijkt} + \sum_{b \in B} \sum_{j \in J} PDA_{bj} * fp_b \tag{2.1}$$

Subject to

$$\sum_{j \in V} x_{ijkt} - \sum_{j \in V} x_{jikt} = 0 \qquad \forall i \in V, \forall k \in K, \, \forall t \in H \tag{2.2}$$

$$\sum_{j \in V} \sum_{k \in K} x_{ijkt} \leq 1 \quad \forall t \in H, \, \forall i \in I \tag{2.3}$$

$$\sum_{j \in V} \sum_{k \in K} x_{jikt} \leq 1 \quad \forall t \in H, \, \forall i \in I \tag{2.4}$$

$$\sum_{i \in I} x_{jikt} \geq x_{bjkt} \quad \forall j \in J, \forall b \in B, \forall k \in K, \forall t \in H \tag{2.5}$$

$$x_{bjkt} \leq PDA_{bj} * VA_{kb} \quad \forall b \in B, \forall j \in J, \forall k \in K, \forall t \in H \tag{2.6}$$

$$x_{bb_1kt} = 0 \quad \forall b, b_1 \in B, \forall k \in K, \forall t \in H \tag{2.7}$$

$$x_{bikt} = 0 \quad \forall b \in B, \forall i \in I, \forall k \in K, \forall t \in H \tag{2.8}$$

$$x_{jbkt} = 0 \quad \forall j \in J, \forall b \in B, \forall k \in K, \forall t \in H \tag{2.9}$$

$$x_{ijkt} = 0 \quad \forall i, j \in J,\ \forall k \in K,\ \forall t \in H,\ i \neq j \tag{2.10}$$

$$\sum_{i \in I} \sum_{b \in B} x_{ibkt} \leq 1 \quad \forall k \in K, \forall t \in H \tag{2.11}$$

$$\sum_{i \in I} q_{ikt} \leq Q_k \quad \forall k \in K,\ \forall t \in H \tag{2.12}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijkt} \leq |S| - 1 \quad \forall k \in K,\ \forall t \in H,\ \forall S \subseteq I \tag{2.13}$$

$$x_{jikt} \leq z_{ij} \quad \forall j \in J, \forall i \in I, \forall k \in K, \forall t \in H \tag{2.14}$$

$$\sum_{j \in J} z_{ij} = 1 \quad \forall i \in I \tag{2.15}$$

$$z_{ij} \leq y_j \quad \forall i \in I,\ \forall j \in J \tag{2.16}$$

$$\sum_{i \in I} \left(z_{ij} \sum_{t \in H} d_{it} \right) \leq C_j \quad \forall j \in J \tag{2.17}$$

$$\sum_{u \in I} x_{ujkt} + \sum_{u \in V \setminus \{i\}} x_{iukt} \leq 1 + z_{ij} \quad \forall i \in I,\ \forall j \in J,\ \forall \mathrm{k} \in \mathrm{K}, \forall t \in H \tag{2.18}$$

$$\sum_{i \in I} \sum_{k \in K} \sum_{t \in H} x_{jikt} \geq y_j \quad \forall j \in J \tag{2.19}$$

$$\sum_{i \in I} x_{jikt} \leq y_j \quad \forall j \in J,\ \forall k \in K,\ \forall t \in H \tag{2.20}$$

$$\sum_{p \in H} w_{itp} = d_{it} \quad \forall i \in I,\ \forall \mathrm{t} \in \mathrm{H} \tag{2.21}$$

$$\sum_{t \in H} w_{itp} = \sum_{k \in K} q_{ikp} \quad \forall i \in I,\ \forall p \in H \tag{2.22}$$

$$q_{ikt} \leq M \sum_{j \in V} x_{ijkt} \quad \forall i \in I,\ \forall t \in H,\ \forall k \in K \tag{2.23}$$

$$\sum_{j \in V} x_{ijkt} \leq M q_{ikt} \quad \forall i \in I,\ \forall t \in H,\ \forall \mathrm{k} \in \mathrm{K} \tag{2.24}$$

$$\mathrm{x}_{ijkt} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J,\ \forall t \in H,\ \forall k \in K \tag{2.25}$$

$$y_j \in \{0,1\} \quad \forall j \in J \tag{2.26}$$

$$z_{ij} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J \tag{2.27}$$

$$q_{ikt} \leq \min \left\{ Q_k, \sum_{p \in H} d_{ip} \right\} \quad \forall i \in I,\ \forall j \in J\ ,\forall k \in K \tag{2.28}$$

$$w_{itp} \leq d_{ip} \quad \forall i \in I,\ \forall t, p \in H \tag{2.29}$$

The above MIP model is an extension of the work of Zhang et al. (2014) and considers distribution outsourcing additionally. The objective function minimizes the total cost consisting of facilities opening cost, holding costs per unit of product kept at customers, routing costs and outsourcing costs. However, a short description of them is also provided in this section. Constraints 2.2 guarantee the equilibrium between the interior and exterior vehicles' flow in each node. Constraints 2.3 and 2.4 guarantee that each customer is visited by exactly one vehicle per period. Constraints 2.5 ensure that if a vehicle is sent from a provider to a depot, it should also be sent from that depot to a customer in a selected time period. Constraints 2.6, ensure that a vehicle will be moved from a provider to, an allocated to him depot with a vehicle owned by him. Constraints 2.7-2.10 forbid a vehicle to be moved from provider to provider, from provider to customer, from depot to provider and from depot to depot, respectively. Constraints 2.11 prevent a vehicle from performing more than one route per period. Constraints 2.12 impose that the capacity of each vehicle will not be exceeded. The subtour elimination requirements are given in constraints 2.13. Constraints 2.14 guarantee that a vehicle will be travelled from a depot to a customer only if that customer is allocated to the depot. Constraints 2.15 and 2.16 ensure that a customer must be allocated to exact one depot over the time horizon. Constraints 2.18 respect the capacity of each depot. Constraints 2.19 prevent the linking of a customer to a depot, if the customer is not allocated to that depot. A vehicle can be moved from a depot to a customer, only if that depot is opened as imposed by constraints 2.20 and 2.21. The total amount of deliveries must be equal to the demand of each customer as it is stated in constraints 2.22. Constraints 2.23 guarantee that, the total amount of scheduled deliveries for a customer must be equal to the overall actual deliveries to that customer. If a customer receives a replenishment on a specific time period by a specific vehicle, he should be visited by that vehicle as imposed by constraints 2.24 and 2.25.

2.4 Solution approach

2.4.1 Initialization phase

In order to find a feasible initial solution, a two-phase constructive heuristic has been implemented. Location and allocation decisions are made in the first phase while, inventory-routing decisions are determined in the second phase.

Location-allocation strategy

To determine the location and allocation decisions, a ratio-based depots' selection procedure combined with a nearest customer allocation strategy have been developed. In the depots' selection method, the ratio $\frac{fixed\_opening\_cost}{Capacity}$ is initially computed for each candidate depot and then, the depot with the minimum ratio is chosen. In the case that two or more depots have the same ratio, one of them is selected arbitrarily (commonly the first found). Then, for each opened depot the nearest customers' allocation strategy is applied. More precisely, the nearest customer to the opened depot is chosen. If the total demand of this customer is less or equal to the remaining capacity of the depot then, the selected customer is allocated to the depot. This first phase of the constructive heuristic is executed until the allocation of all customers.

Also, each opened depot is allocated to a vehicles' provider based on a minimum cost criterion (fixed-contract cost plus the routing cost depicted as the distance between the provider and the depot).

Inventory-routing construction

For each time period and each depot, a number of vehicles is selected in order to guarantee demand satisfaction of customers allocated to the current depot. For each selected vehicle an assignment of customers is done based on the limited capacity of the vehicle. In order to build the route of each vehicle, the Random Insertion move is applied (Glover et al., 2001). According to the inventory decisions, the quantity scheduled to be sent to each customer in each time period equals to its corresponding demand in that period.

2.4.2 Improvement Phase

Neighborhood structures

Six neighborhood structures are considered for guiding the search during the improvement phase as follows:

Inter-route Relocate (N_1): This local search operator removes customer i from his current route R_i and re-inserts him in a new route R_b, after customer b, in each period. A prerequisite for applying this move is, both customers i and b to be visited by vehicles in the same periods. Routes R_i and R_b, could be allocated either to the same depot or to different depots over the time horizon. If the move violates the capacity of the vehicle in route R_b, a replenishment shifting move is applied. Four possible cases are met for this neighborhood:

- Case 1: R_i and R_b are assigned to the same depot and no violations occur on vehicles capacities.
- Case 2: R_i and R_b are assigned to the same depot and vehicles capacity violations occur.
- Case 3: R_i and R_b are assigned to different depots and no violations occur on vehicles capacities.
- Case 4: R_i and R_b are assigned to different depots and vehicles capacity violations occur.

In case 1 only routing decisions are taken. In the second case, both routing and inventory decisions are made, while in the third case routing and allocation decisions are improved. Finally, in case 4 routing, inventory and allocation decisions are simultaneously addressed. Figures 2.1 and 2.2 provide an illustrative example of applying the inter-route relocation of customer $C3$ after customer $C1$ (customers are allocated to the same depot) in a three periods instance.

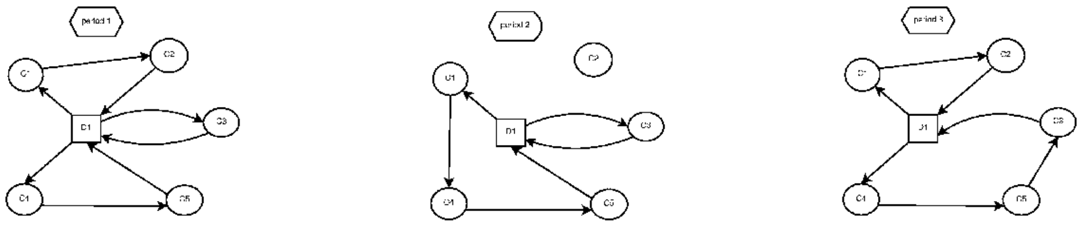

FIGURE 2.1: Routes from the same depot in each time period before the application of the inter-route relocate move.

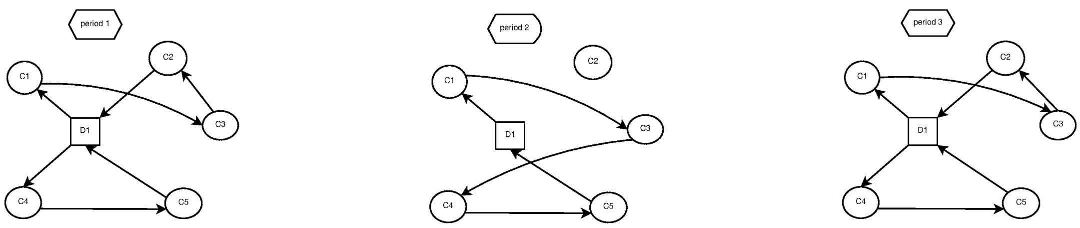

FIGURE 2.2: Routes from the same depot in each time period after the application of the inter-route relocate move.

An illustration of the inter-route relocate move applied on customers allocated to different depots, is shown in Figures 2.3 and 2.4. More specifically, customer $C2$ is removed from his current position and is inserted in the position after customer $C4$.

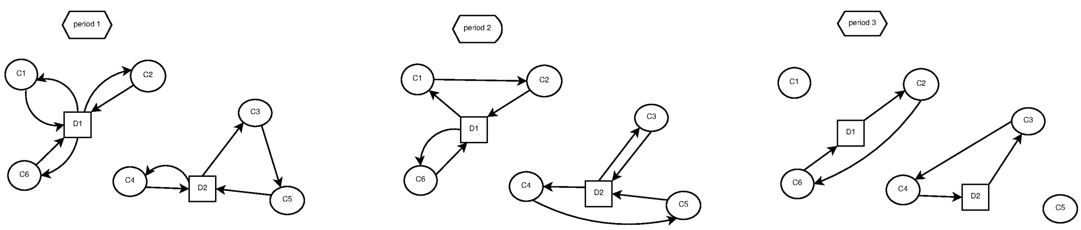

FIGURE 2.3: Routes from different depots in each time period before the application of the inter-route relocate move.

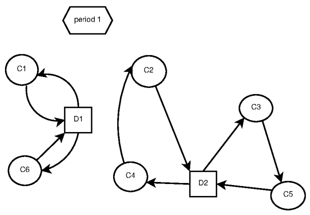

FIGURE 2.4: Routes from different depots in each time period after the application of the inter-route relocate move.

Inter-route Exchange (N_2): This neighborhood consists of swapping the positions of two customers (i and b) from different routes (R_i and R_b) over the time horizon. Routes R_i and R_b could be allocated to the same depot or to different depots. In the first case the move may not be applied to all time periods, while in the second case the swapping will be considered as applicable only if it is valid for all time periods. Three special cases could be met by applying this move:

- Case 1: No vehicles' capacity violations occurred.
- Case 2: The demand of customer i violates the capacity of the vehicle servicing customer b in one or more time periods.
- Case 3: The demand of customer b causes violations of the capacity of vehicle servicing customer i in one or more time periods.

In the above case 1 only routing decisions are made, while in cases 2 and 3 both routing and inventory decisions are tackled (inventory: forward/backward shifting to the nearest time periods). If customers are allocated in different depots, changes on allocation decisions are then applied. Figures 2.5 and 2.6 illustrate the application of the move in a three period instance. In the first period, customers C2 and C3 are swapped, while in the periods two and three, the pairs of exchanged customers are ($C2$, $C3$) and ($C3$, $C4$), respectively.

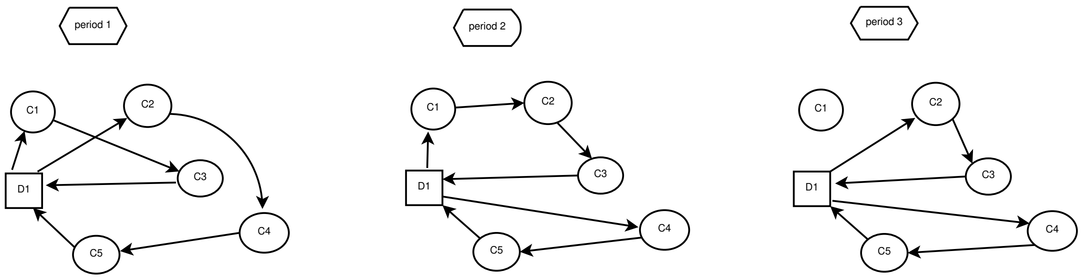

FIGURE 2.5: Routes from the same depot in each time period before the application of the inter-route exchange move.

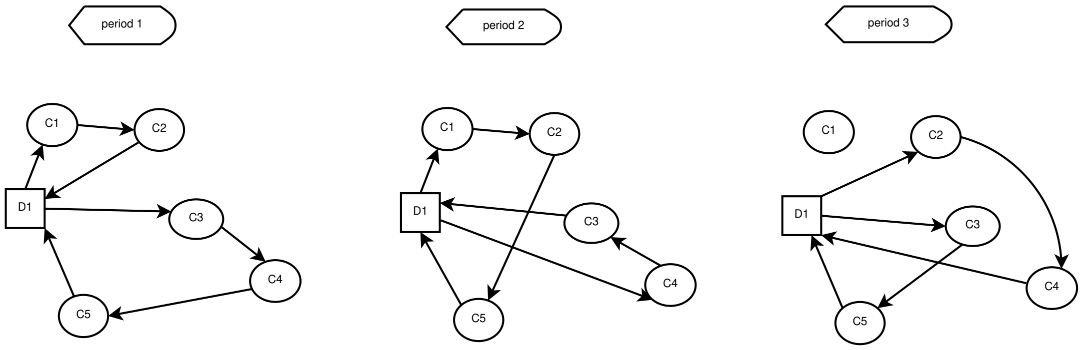

FIGURE 2.6: Routes from the same depot in each time period after the application of the inter-route exchange move.

An example of the inter-route exchange move between customers C3 and C4, allocated to different depots, is illustrated in Figures 2.7 and 2.8.

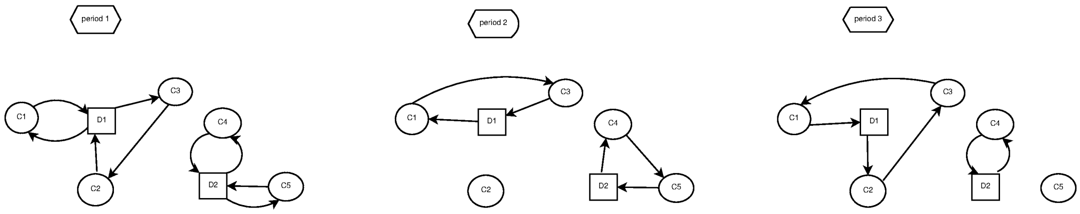

FIGURE 2.7: Routes from different depots in each time period before the application of the inter-route exchange move.

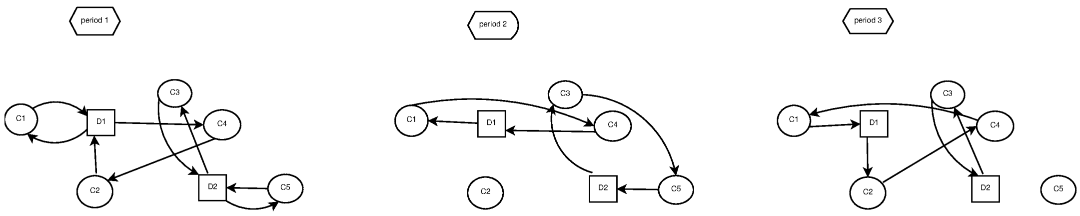

FIGURE 2.8: Routes from different depots in each time period after the application of the inter-route exchange move.

Exchange Opened-Closed Depots (N_3): This neighborhood consists of exchanging a closed depot i with a currently opened one j. The exchanging cost is calculated for each closed depot, with all opened depots. Then, the opened depot with the minimum exchanging cost is marked as closed and the validation of the move is examined. In the case of a valid move, a reordering of the routes allocated on depot j is calculated, based on the minimum insertion cost criterion of depot i. If the overall cost (location and routing costs) is decreased then, the move is marked as accepted and it is applied. The move is summarized in the following example in Figures 2.9 and

2.10. As it can be seen, in the route of customers $C3, C4$ and $C5$, a routing re-ordering has also been applied.

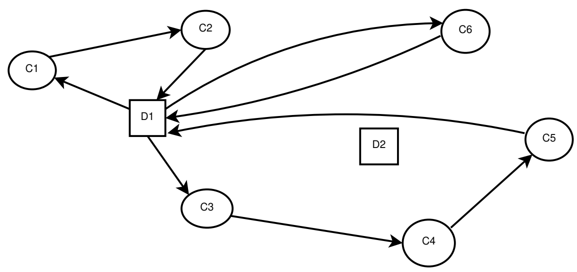

FIGURE 2.9: Routes allocation before the opened-closed exchange move.

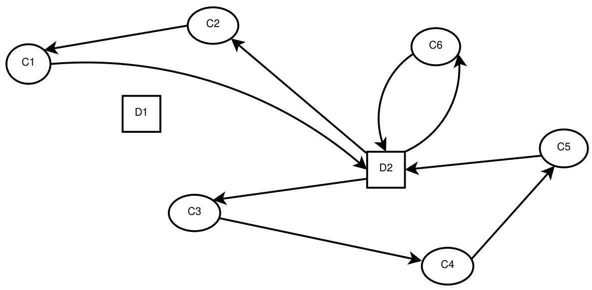

FIGURE 2.10: Routes allocation after the opened-closed exchange move.

Intra-route Relocate (N_4): The intra-route relocate operator removes a customer from its current position in its route and re-inserts it in a different position. This move handles only routing decisions. In Figures 2.11 and 2.12 the relocation of customer $C2$ after the customer $C4$ is illustrated.

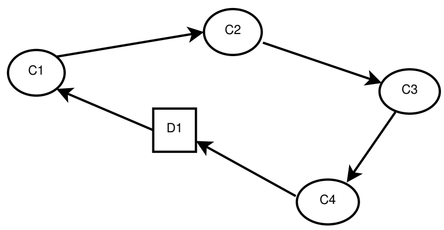

FIGURE 2.11: Routes from the same depot in each time period before the application of the intra-route relocate move.

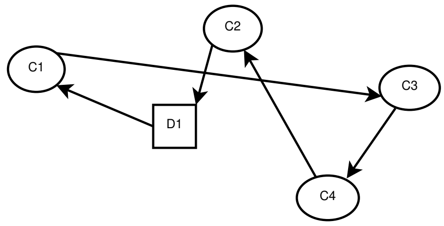

FIGURE 2.12: Routes from the same depot in each time period after the application of the intra-route relocate move.

2-2 Replenishment Exchange (N_5): This local search operator randomly selects two time periods t_1 and t_2 and then finds the two most distant customers i and b, both serviced in those two periods. Then, it computes the cost changes of removing i and b from their routes in periods t_1 and t_2 respectively and shifting their receiving deliveries from t_1 to t_2 for customer i and from t_2 to t_1 for b. This move is applied only in case where an improvement is produced and no vehicles' capacities are violated. Figures 2.13 and 2.14 provide an illustrative example of this move, applied on customers $C1$ and $C4$, that are allocated in the same route.

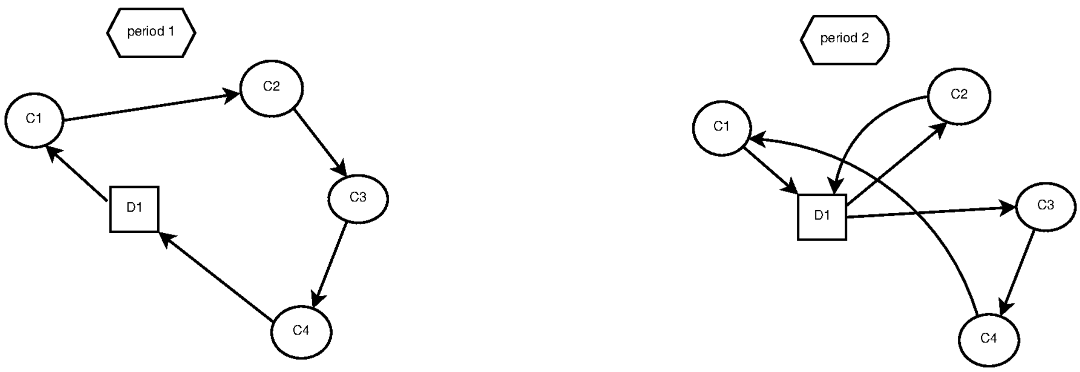

FIGURE 2.13: Routes in the two selected time periods before the application of the 2-2 replenishment exchange move.

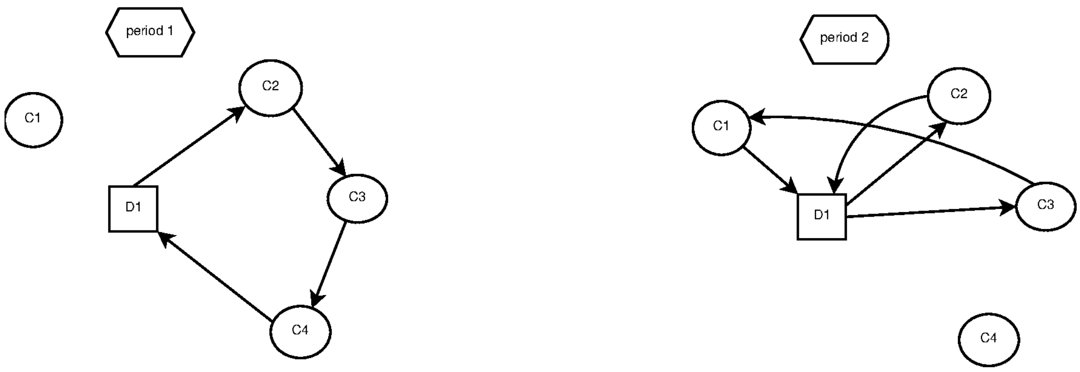

FIGURE 2.14: Routes in the two selected time periods after the application of the 2-2 replenishment exchange move.

The 2-2 Replenishment Exchange can also be applied on customers allocated to different routes. An example is presented in Figures 2.15 and 2.16, in which the move is applied between customers $C1$ and $C5$.

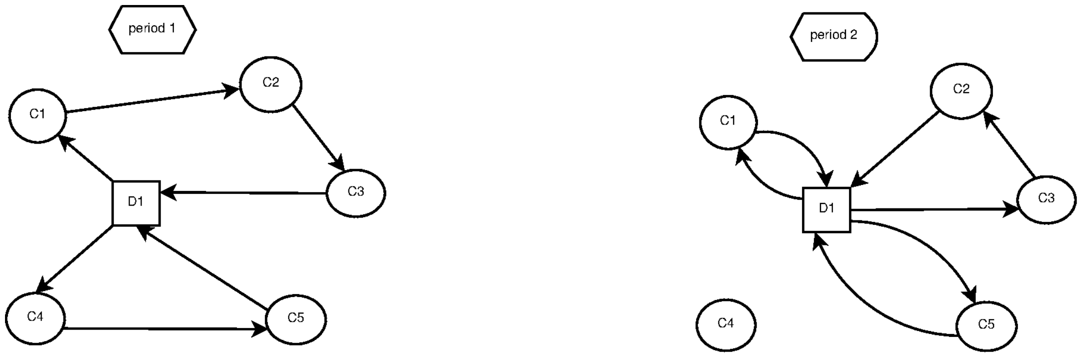

FIGURE 2.15: Routes in the two selected time periods before the application of the 2-2 replenishment exchange move.

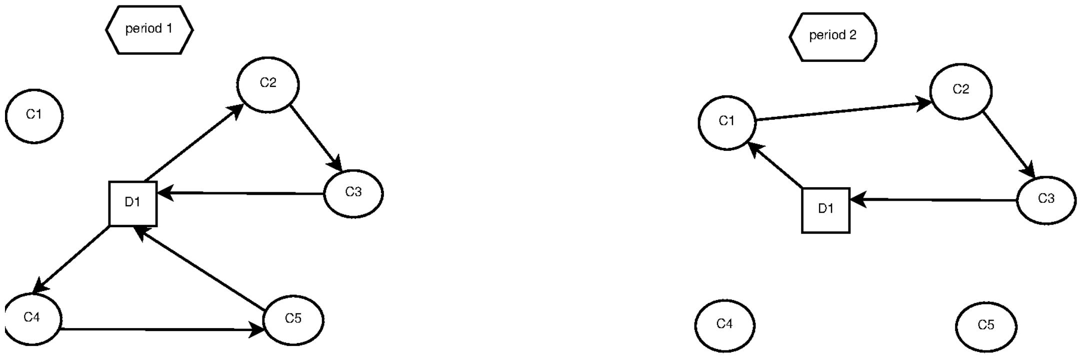

FIGURE 2.16: Routes in the two selected time periods after the application of the 2-2 replenishment exchange move.

Change Provider (N_6): This local search operator examines for each opened depot if an improvement may be achieved by allocating it to an other vehicles' provider. In the following illustrated example (Figure 2.17), the depot $D1$ which is allocated to provider $P1$, it will be allocated to provider $P2$. The order of customers in the routes remains the same.

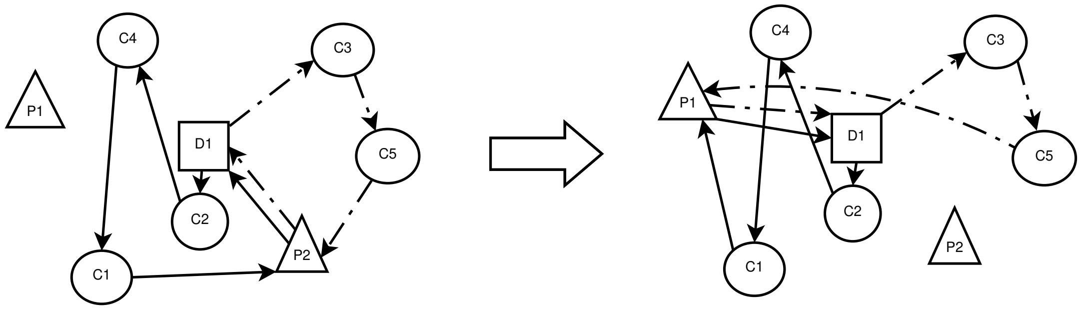

FIGURE 2.17: An example of the Change Provider operator

In order to avoid vehicles' capacity violations by applying the Inter-route Relocate and Inter-route Exchange moves, shifting of surplus product quantity may be needed to be also applied. An example of the application of the shifting procedure is given in Figure 2.18. As it can be seen, the surplus quantity in the second time period is equal to 15 for a selected customer. This customer is also serviced in first and third time periods and the available free space in the corresponding vehicles in these periods are 18 and five respectively. So, five units of product are shifted forward to the third time period and 10 units are shifted backward in the first time period.

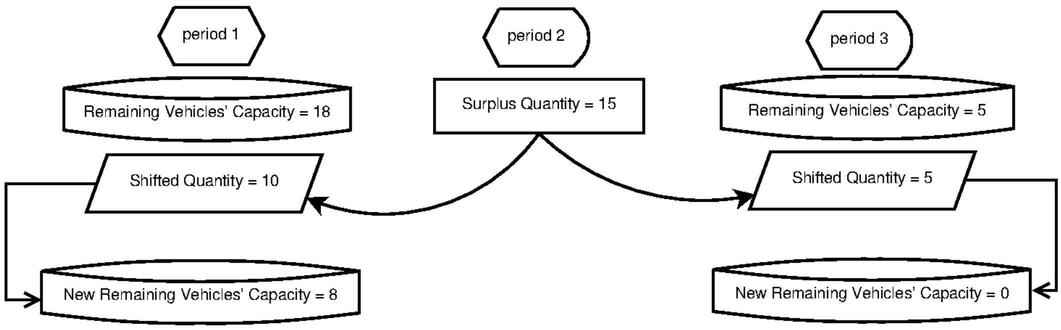

FIGURE 2.18: Example of the quantity shifting procedure.

Shaking procedure

A shaking procedure is developed in order to escape from local optimum solutions (Hansen et al., 2017). Thus, in each shaking phase a number of random jumps are applied in a randomly selected neighborhood from a predefined set of neighborhoods. The pseudo-code of this diversification method is presented in Algorithm 4, with the incumbent solution S and the maximum number of random jumps $k_{max} = 12$ (where k_{max} was experimentally set) as input. The new solution S' is obtained by applying k (where $1 < k < k_{max}$) times one randomly selected neighborhood (from the total l_{max} = 4 neighborhoods) and it is then returned as output.

Algorithm 4 Shaking Procedure

```
procedure SHAKE(S, k, l_max)
    l = random_integer(1, l_max)
    for i ← 1, k do
        select case(l)
        case(1)
        S' ← Inter_Relocate(S)
        case(2)
        S' ← Exchange_OpenedClosed_Depots(S)
        case(3)
        S' ← Intra_Relocate(S)
        case(4)
        S' ← Inter_Exchange(S)
        end select
    end for
    Return S'
end procedure
```

General Variable Neighborhood Search (GVNS)

As it has already been mentioned, the GVNS variant is an extension of the BVNS. Its main difference is the usage of a VND scheme as an improvement strategy. In this chapter, GVNS-based solution methods using either the cVND or the pVND as the intensification phase and both the first and best improvement (FI and BI respectively) search strategy are examined. The proposed GVNS algorithm is summarized in the following pseudo-code.

Algorithm 5 General VNS

```
procedure GVNS(S, k_max, max_time, l_max)
    while time ≤ max_time do
        for k ← 1, k_max do
            S* = Shake(S,k, l_max)
            S' = pVND(S*)
            if f(S') < f(S) then
                S ← S'
            end if
        end for
    end while
    return S
end procedure
```

Inventory Rescheduling Procedure

The Inventory Rescheduling Procedure (InvRP) functions as a post optimization part of the proposed solution approach. In each iteration, the most distant customer is selected and the total periods in which this customer is visited by vehicles, in order to satisfy his demand, are kept. Then, an alternative replenishment scheme is examined, trying to reduce the periods needed to visit the selected customer and as a result to reduce the routing costs. The method is terminated either when all customers have been checked or a time stopping criterion is met. An explanation of the variables and the parameters presented in the pseudo-code of Inventory Rescheduling Procedure is firstly given and then, the pseudo-code is provided in Algorithm 6.

- $NPeriods$: Keeps the total number of time periods.
- $NumOfNeededPeriods$: keeps the minimum number of periods needed to satisfy the total demand of a selected customer.
- $DemOfI$: Keeps the total demand of a selected customer over the planning horizon.
- $AvailableVehicles$: a binary 2D array ($N_{Vehicles} * N_{Periods}$) which denotes if a vehicle can visit a selected customer in a period (value equals 1, or not (value equals 0).
- $Veh2ServeI$: Stores the vehicle scheduled to visit a selected customer in a specific period.

- *Period2ServeI* : Logical array which marks the selected periods in the new replenishment plan as "True".

Algorithm 6 Inventory Rescheduling Procedure

```
procedure InvRP(S)
    while a stopping criterion is not met do
        Find the most distant customer i of all opened depots
        Mark customer i as "checked"
        for t ← 1, NPeriods do
            Find all the available vehicles for visiting i in t
            Store those vehicles in AvailableVehicles
        end for
        Compute the total periods in which i is currently serviced,
NPeriodsServedI
        Compute the total demand of customer i, DemOfI
        NumOfNeededPeriods = 0
        while DemOfI > 0 do
            Find an unselected vehicle k ∈ AvailableVehicles with maxCapacity
            Keep the vehicle in Veh2ServeI and the period in Period2ServeI
            NumOfNeededPeriods = NumOfNeededPeriods + 1
            Recalculate DemOfI based on the partial new replenishment schedule
        end while
        if NumOfNeededPeriods < NPeriodsServedI then
            for t ← 1, NPeriods do
                if Period2ServeI(t) then
                    Calculate changes on Inventory_Cost and Routing_Cost
                    Reschedule vehicle routes for i
                end if
            end for
            if Improvement then
                Renew Inventory and Routing Costs and Inventory_Levels
                Apply the routes' rescheduling
            end if
        end if
    end while
    return S
end procedure
```

GVNS-InvRP

The pseudo-code of the GVNS-InvRP follows in Algorithm 7.

Algorithm 7 GVNS-InvRP

```
procedure GVNS-InvRP
    S ← TwoPhaseConstructionHeuristic
    while time < max_time do
        S' ← GVNS(S)
    end while
    S ← InventoryReschedulingProcedure(S')
    return S
end procedure
```

Figure 2.19 illustrates a flowchart of the proposed solution approach.

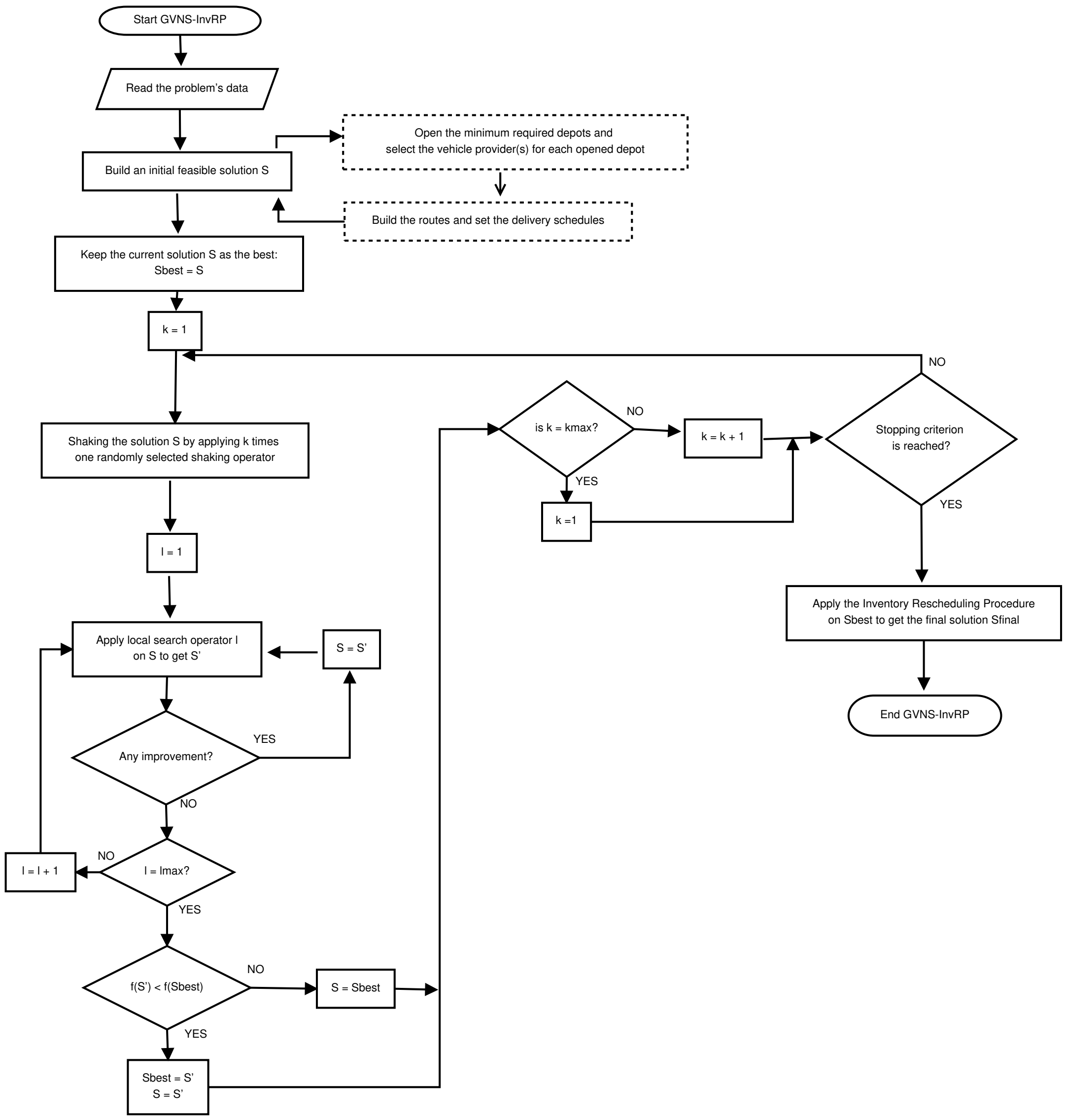

FIGURE 2.19: The flowchart of the proposed solution approach.

Initially, a feasible solution is obtained using the two-phase construction heuristic. This is the initial solution of the proposed solution method. GVNS is iteratively executed for 60s and alternates its shaking procedure, the pipe-VND procedure and the solution update step. In each shaking iteration, one of the shaking operators (see subsection 2.4.2) is randomly selected and applied k times (where the parameter k starts with the value one and increased by one in each iteration until $k = kmax$. If k equals k_{max} and the stopping criterion is not met, then the parameter k is set to one). Then, the pVND is applied. In this improvement step, all the local search operators are sequentially applied with the predefined order (see subsection **??**). The search with each local search operator is continued until no more improvements are obtained. After the completion of the pipe-VND, a solution update is performed, by checking if a better solution is available. Subsequently to the GVNS, the Inventory Rescheduling procedure is applied for each customer, starting from the most distant one. The goal of this post-optimization method is to reduce total cost by rescheduling the replenishment plan mainly of the most distant customers, in order to avoid frequent deliveries to them.

Furthermore, an auxiliary subroutine has been developed in order to ensure the feasibility of each solution. This subroutine examines whether the new solution satisfies the constraints of the model and checks the validations of cost renewals.

2.5 Computational analysis and results

In this section, a computational analysis is presented in order to evaluate the performance of the proposed solution method. In subsection 4.5.1, the necessary technical details (e.g., computing environment) are provided. Subsection 2.5.2 provides the results achieved by solving the LIRPDO while the subsection 2.5.3 summarizes the results obtained by the proposed algorithm on 20 LIRP benchmarks from the work of Zhang et al. (2014) and compared with those achieved by the proposed methods presented in the same work. In subsection 2.5.4 the results achieved by the GVNS-InvRP on 30 randomly generated large-scale instances are presented.

2.5.1 Computing environment & parameter settings

The methods presented in this work were implemented in Fortran and ran on a desktop PC running Windows 7 Professional 64-bit with an Intel Core i7-4771 CPU at 3.5 GHz and 16 GB RAM. The compilation of codes was done using Intel Fortran compiler 18.0 with optimization option /O3. Also, the maximum execution time limit was set (max_time = 60s) for the GVNS approach. The LIRPDO instances were modeled using GAMS (GAMS 24.9.1) (Brooke et al., 1998) and solved using CPLEX 12.7.1.0 solver with specified time limit (2h). CPLEX ran in the same computing environment with Intel Fortran compiler.

2.5.2 Computational results on LIRPDO instances

This is the first study introducing the LIRPDO, thus there are no previously published test instances in order to compare the efficiency of the proposed solution method. Consequently, 20 new instances were randomly generated following the instructions described in subsection 5.3.1 in the work of Zhang et al. (2014). The instances' names are shaped as X-Y-Z-L, where X is the number of potential depots, Y the number of potential vehicles' providers, Z the number of customers and L the number of time periods. These problem instances are available at: http://pse.cheng.auth.gr/index.php/publications/benchmarks.

Figures 2.20 and 2.21 illustrates the performance of the proposed solution approaches using either the cVND and the pVND as the main improvement phase and following both the first and best improvement search strategies. It is obvious that the overall performance can be improved by adopting an adaptive mixed search strategy. More specifically, the best improvement search strategy will be applied for instances with up to 90 customers and the first improvement search strategy for the cases with more than 90 customers.

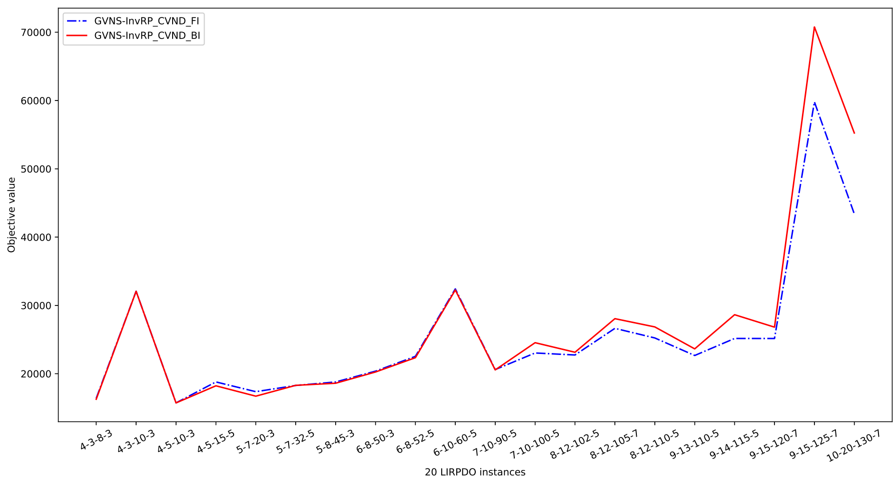

FIGURE 2.20: Performance of GVNS-InvRP with CVND using FI and BI.

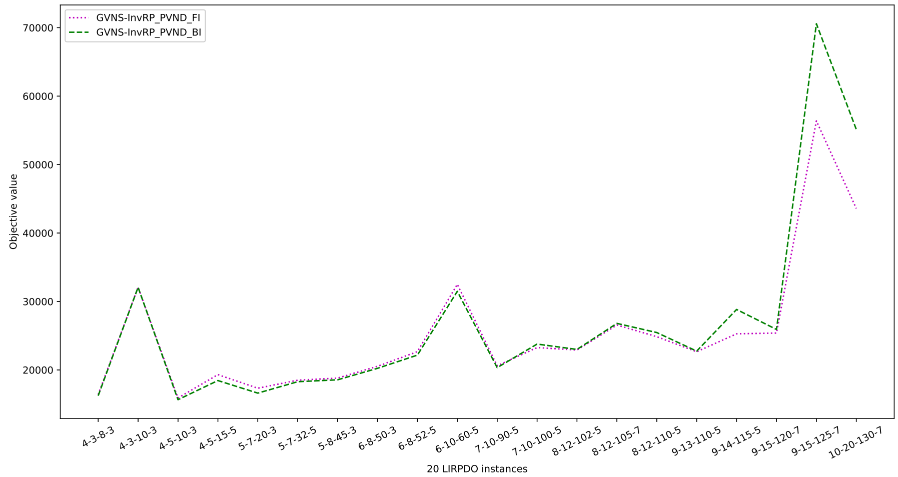

FIGURE 2.21: Performance of GVNS-InvRP with PVND using FI and BI.

Table 2.4 provides the results obtained by CPLEX, GVNS-InvRP with cVND as improvement phase and GVNS-InvRP with pVND. The results of GVNS-InvRP with cVND and GVNS-InvRP with pVND were achieved by the adaptive mixed search strategy. More specifically, in the first column the names of the instances are provided. The second column presents the results achieved by CPLEX, while the third, fourth, and sixth column provide the results achieved by the construction heuristic, the GVNS-InvRP with cVND, and the GVNS-InvRP with pVND, respectively. The fifth and the seventh column show the solution quality gaps of the two proposed methods with the CPLEX. The results of GVNS-based schemes are the average values of five runs per instance.

TABLE 2.4: Computational results on 20 LIRPDO

Instance	CPLEX (a)	CH (b)	$GVNS - InvRP_{cVND}$ (c)	gap (a-c) %	$GVNS - InvRP_{pVND}$ (d)	gap (a-d) %
4-3-8-3	16,253.89	16,738.29	16,262.33	-0.05	16,235.09	0.12
4-3-10-3	31,509.82	32,125.66	32,073.5	-1.79	32,074.06	-1.79
4-5-10-3	15,727.25	19,409.16	15,724.75	0.02	15,650.88	0.49
4-5-15-5	35,379.9	20,592.56	18,230.54	48.47	18,447.7	47.86
5-7-20-3	-	18,738.69	16,713.07	-	16,611.04	-
5-7-32-5	-	20,475.58	18,299.22	-	18,282.23	-
5-8-45-3	-	21,996.29	18,607.6	-	18,556.2	-
6-8-50-3	-	22,910.67	20,253.66	-	20,235.43	-
6-8-52-5	-	26,446.71	22,348.18	-	22,164.47	-
6-10-60-5	-	42,149.89	32,261.79	-	31,480.11	-
7-10-90-5	-	23,385.28	20,559.74	-	20,336.99	-
7-10-100-5	-	28,507.57	23,036.91	-	23,267.29	-
8-12-102-5	-	25,672.16	22,742.8	-	22,908.09	-
8-12-105-7	-	34,655.32	26,642.79	-	26,560.92	-
8-12-110-7	-	29,469.44	25,236.26	-	24,858.34	-
9-13-110-5	-	24,167.27	22,668.91	-	22,658.02	-
9-14-115-5	-	31,685.59	25,162.47	-	25,271.55	-
9-15-120-7	-	31,560.26	25,156.09	-	25,385.52	-
9-15-125-7	-	71,088.4	59,797.43	-	56,382.8	-
10-20-130-7	-	55,651.04	43,396.27	-	43,584.51	-

Also, in Table 2.5 the number of the opened depots and selected providers in the final solution of each solution method are provided.

TABLE 2.5: The number of opened depots and selected providers on 20 LIRPDO instances.

Instance	**CPLEX**		$GVNS - InvRP_{cVND}$		$GVNS - InvRP_{pVND}$	
	Depots	Providers	Depots	Providers	Depots	Providers
4-3-8-3	2	1	2	1	2	1
4-3-10-3	3	2	3	2	3	2
4-5-10-3	2	1	2	1	2	1
4-5-15-5	3	3	2	1	2	1
5-7-20-3	-	-	2	1	2	1
5-7-32-5	-	-	2	1	2	1
5-8-45-3	-	-	2	1	2	1
6-8-50-3	-	-	2	1	2	1
6-8-52-5	-	-	2	1	2	1
6-10-60-5	-	-	2	1	2	1
7-10-90-5	-	-	2	1	2	1
7-10-100-5	-	-	2	1	2	1
8-12-102-5	-	-	2	1	2	1
8-12-105-7	-	-	2	1	2	1
8-12-110-7	-	-	2	1	2	1
9-13-110-5	-	-	2	1	2	1
9-14-115-5	-	-	2	1	2	1
9-15-120-7	-	-	2	1	2	1
9-15-125-7	-	-	2	2	2	2
10-20-130-7	-	-	2	1	2	1

The CPLEX solver was able to provide an integer solution only for the four small-sized instances (4-3-8-3 to 4-5-15-5). For the next six medium-sized instances CPLEX cannot produce any feasible solution within a 2h time limit, while for the last 10 large-sized problem instances an out of memory error occurred during the execution. Both GVNS-based schemes were able to provide even for the small-sized instances high quality solutions in no more than 60 seconds. More specifically, for the case of the three small-sized instances $4-3-8-3$, $4-3-10-3$ and $4-5-10-3$ the solutions obtained by CPLEX solver in 2h are almost equal to those achieved by the proposed methods in one minute. However, for the case of the instance $4-5-15-5$, both $GVNS - InvRP_{cVND}$ and $GVNS - InvRP_{pVND}$ produce 48.47% and 47.86% better solutions than CPLEX, respectively.

Table 2.6 reports the best found values of the 20 LIRPDO instances.

TABLE 2.6: Best values found on 20 LIRPDO instances

Instance	Best	Instance	Best
4-3-8-3	16,208.14	7-10-90-5	20,310.28
4-3-10-3	32,058.2	7-10-100-5	22,843.81
4-5-10-3	15,501.9	8-12-102-5	22,683
4-5-15-5	18,146.71	8-12-105-7	26,232.41
5-7-20-3	16,604.6	8-12-110-7	24,823.45
5-7-32-5	18,011	9-13-110-5	22,371.2
5-8-45-3	18,537.22	9-14-115-5	24,958.39
6-8-50-3	20,168.32	9-15-120-7	25,005.08
6-8-52-5	22,000.75	9-15-125-7	54,365.64
6-10-60-5	31,410.41	10-20-130-7	42,398.68

2.5.3 Computational results on LIRP instances (Zhang et al., 2014)

The proposed methods with minor modification (remove the Change Provider local search operator and disable the provider selection in construction heuristic) can also solve LIRP instances, following the approach of Zhang et al. (2014). In this section a comparative analysis between the proposed GVNS-InvRP method, the SA-Hyb-ILRP, and the Sequential heuristic presented by Zhang et al. (2014) is provided. They divided the 20 LIRP instances into small-sized and large-sized instances. The 20 LIRP instances were classified according to their size, as presented in the literature (Mjirda et al., 2014) as follows:

- 10 small sized instances (with less than 20 customers).
- 6 medium sized instances (with customers between 20 and 90).
- 4 large sized instances (with more than 90 customers).

Figure 2.22 illustrates the performance of the two GVNS-based schemes using the adaptive mixed search strategy in all 20 LIRP instances. The results of the two methods are close enough, except the cases of some instances in which the GVNS-InvRP with the pVND as improvement method generates better solutions. Therefore, the GVNS-InvRP using pVND is selected to be compared with the solution approaches presented by Zhang et al. (2014).

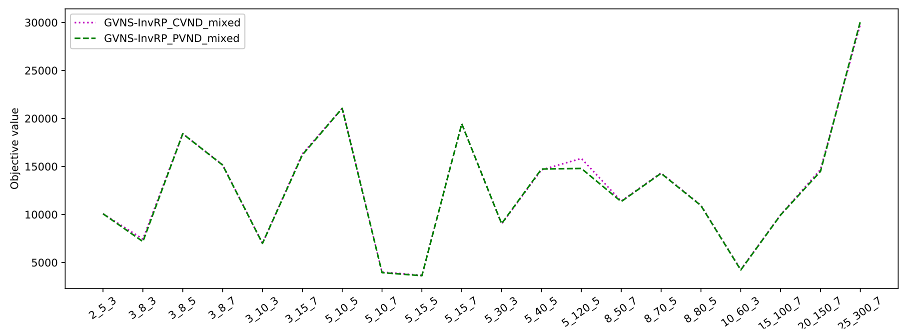

FIGURE 2.22: Performance of GVNS-InvRP with cVND and pVND using the adaptive mixed search strategy.

Table 2.7 shows the results of all three methods. More specifically, in the first column the names of instances are given. The name of each instance formed as $D - C - P$, where D is the number of the potential locations of depots, C is the number of customers and P represents the time periods. In columns 2 and 3 the results of SA-Hyb-ILRP and Sequential heuristic (Zhang et al., 2014) are provided, while the fourth column presents the average values of each instance (average of 5 runs) achieved by GVNS-InvRP method. The last two columns report the solution quality gap between the proposed method and the methods of Zhang et al. (2014).

The gaps are calculated as follows:

$$\text{Gap}_{SA\_Hyb\_ILRP-Seq\_GVNS\_InvRP} = \frac{(S_{SA\_Hyb\_ILRP} - S_{Seq\_GVNS\_InvRP})}{S_{SA\_Hyb\_ILRP}} * 100$$

and

$$\text{Gap}_{Sequentialheuristic-Seq\_GVNS\_InvRP} = \frac{(S_{Sequentialheuristic} - S_{Seq\_GVNS\_-InvRP})}{S_{Sequentialheuristic}} * 100 \,.$$

TABLE 2.7: Computational results on 20 LIRP benchmarks (Zhang et al., 2014)

Instance	SA-Hyb-ILRP (a)		Sequential heuristic (b)		GVNS-InvRP (c)		gap (a-c) %	gap (b-c) %
	Objective	Time	Objective	Time	Objective	Time		
2-5-3	9,958.98	15.96	10,363.16	1.14	10,072.38	60	-1.14	2.81
3-8-3	6,774.87	46.8	6,799.58	5.1	7,176.86	60	-5.93	-5.55
3-8-5	17,654.66	74.21	19,458.83	6.52	18,407.34	60	-4.26	5.4
3-8-7	14,252.47	125.14	14,372.96	5.25	15,144.31	60	-6.26	-5.37
3-10-3	6,530.9	260.32	7,101.85	12.32	6,986.72	60	-6.98	1.62
3-15-7	15,220.19	485.68	17,980.15	18.65	16,199.62	60	-6.44	9.90
5-10-5	19,936.59	587.63	20,070.7	20.12	21,055.71	60	-5.61	-4.91
5-10-7	3,296.23	495.6	3,709.88	14.23	3,941.18	60	-19.57	-6.23
5-15-5	3,143.41	523.65	4,157.6	17.86	3,622.48	60	-15.24	12.87
5-15-7	18,531.83	547.21	18,820.99	19.85	19,444.16	60	-4.92	-3.31
5-30-3	8,343.27	587.6	8,402.36	18.52	9,043.30	90	-8.39	-7.63
5-40-5	13,507.89	698.52	13,919.62	26.5	14,731.14	90	-9.06	-5.83
5-120-5	28,938.4	1042.68	37,906.5	80.35	14,793.61	90	48.88	60.97
8-50-7	10,127.58	714.3	19,341.65	28.65	11,340.69	90	-11.98	41.37
8-70-5	12,391.97	498.63	12,794.09	16.5	14,284.63	90	-15.27	-11.65
8-80-5	9,520.99	785.6	11,030.4	29.85	10,928.24	90	-14.78	0.93
10-60-3	3,837.94	695.25	4,148.37	20.2	4,216.21	90	-9.86	-1.64
15-100-7	27,761.56	1236.21	37,728.64	30.98	9,947.47	90	64.17	73.63
20-150-7	46,148.96	1562.3	55,912.54	58.47	14,485.69	90	68.61	74.09
25-300-7	87,186.54	2365.87	88,003.22	205.85	30,094.08	90	65.48	65.8
Average	18,153.26	667.46	20,601.15	31.84	12,795.79	75	5.07	14.86

As it is shown in Table 2.7 the solutions obtained by the GVNS-InvRP on the 20 large-sized instances (Zhang et al., 2014) are 5.07% better than the SA-Hyb-ILRP and 14.86% better than the Sequential heuristic, while the corresponding improvements for the four large-scale instances are 61.9% and 68.63%, respectively. The proposed method is much faster than the SA-Hyb-ILRP as it produces the solutions of all 20 LIRP instances in an average time of 75 s, while the SA-Hyb-ILRP needs 667.46 s. Focusing on the four large-scale instances, the GVNS-InvRP solves them with an average time of 90 s, while the SA-Hyb-ILRP needs an average of 1551.77 s. The Sequential Heuristic functions faster than the GVNS-InvRP for all the 20 LIRP instances, but for the four large-scale instances its average computational time is increased to 93.91 s.

It can be noticed that, the SA-Hyb-ILRP performs 7.64% better than the GVNS-InvRP on small-sized instances and 11.56% on six medium-sized instances. This preeminence of SA-Hyb-ILRP may be attributed to its hybridization with exact methods. Sequential heuristic performs almost equivalently to the GVNS-InvRP for the case of small-scaled instances, with the proposed method to produce 0.72% better solutions and 2.59% better solutions for medium-sized problem instances. This inefficiency of the proposed solution method may be attributed to the use of a predefined order of the local search operators in the pVND, which can potentially limit the exploration performance of the proposed solution method for small- and medium-sized problem instances. However, the main strength of the GVNS-InvRP is its ability of opening the minimum required number of depots for satisfying the total demand of all customers.

TABLE 2.8: Number of opened depots in four large-sized LIRP instances (Zhang et al., 2014)

Instance	CFLP	GVNS-InvRP	SA-Hyb-ILRP
5-120-5	2	2	4
15-100-7	2	2	6
20-150-7	2	2	7
25-300-7	2	2	14

Table 2.8 presents the number of the opened depots by each solution method for the four large-sized instances. The second column provides the number of opened depots in the optimal solution of the Capacitated Fixed-charge Location problem presented by Zhang et al. (2014), and columns 3 and 4 provide the number of opened depots in the final solution of GVNS-InvRP and SA-Hyb-ILRP, respectively. The first column contains the instances names. As it can be observed, GVNS-InvRP focus on opening the minimum number of the needed depots. The randomly opening of depots in Depot-Exchange post optimization operator in SA-Hyb-ILRP might be attributed for opening more depots and as a consequence for increasing the overall cost.

2.5.4 Computational results on randomly generated large-scale instances

30 new large-scale instances (currently the largest available in the literature) were generated following the instructions described by Zhang et al. (2014) in subsection 5.3.1. The smallest instance consists of 28 depot potential locations, 320 customers, and 7 time periods while the biggest one consists of 120 depot potential locations, 680 customers, and 12 time periods. These problem instances are available http://pse.cheng.auth.gr/index.php/publications/benchmarks.

In order to evaluate the performance of GVNS-InvRP on these large-scale instances, a comparison between GVNS-InvRP and CPLEX is attempted, but an out-of-memory error is occurred. Table 2.9 reports the average and the best solutions achieved by GVNS-InvRP with either cyclic and pipe VND as improvement method. Each solution reported in Table 2.9 is the average value of five runs.

TABLE 2.9: Computational results on 30 large scale LIRP instances (average & best solutions)

Instance	$GVNS - InvRP_{cVND}$	$GVNS - InvRP_{pVND}$	**Gap %**	**BestKnownValue**
28-320-7	19,335.81	19,431.41	-0.49	19,011.52
30-350-7	18,499.62	17,901.95	3.23	17,882.12
30-375-7	32,852.77	31,379.75	4.48	30,015.27
32-380-7	26,085.52	24,134.91	7.48	23,456.88
35-400-7	20,804.13	21,079.49	-1.32	20,658.84
37-415-7	26,222.85	26,210.01	0.05	25,746.4
40-420-7	21,438.87	21,472.92	-0.16	21,301
42-450-7	29,078.81	29,021.97	0.2	28,740.94
45-480-7	26,407.55	26,398.17	0.04	26,189.98
47-490-7	23,237.25	23,264.26	-0.12	23,015.13
50-490-9	41,778.62	41,851.58	-0.17	41,229
52-495-9	65,948.28	66,398.66	-0.68	65,335.34
55-500-9	31,431.10	31,817.83	-1.23	31,379.49
62-510-9	79,544.16	79,263.62	0.35	78,725.77
65-520-9	32,688.15	32,593.36	0.29	32,072.06
67-540-9	42,637.17	43,064.84	-1	41,954.39
70-550-9	48,195.86	48,273.41	-0.16	48,024
74-560-9	33,359.90	33,081.88	0.83	32,907.79
78-570-9	35,041.01	35,799.88	-2.17	34,862.95
80-580-9	61,245.09	62,322.33	-1.76	60,756.2
85-590-12	68,559.28	69,257.85	-1.02	68,367.91
90-595-12	104,381.54	104,045.38	0.32	103,671.9
92-600-12	78,181.17	78,074.18	0.14	77,596.09
95-610-12	78,976.95	80,957.60	-2.51	76,871.55
98-620-12	65,268.30	64,902.25	0.56	64,808.86
100-650-12	51,805.04	51,429.94	0.72	51,337.58
105-655-12	132,348.48	132,425.78	-0.06	131,798.1
110-660-12	54,551.25	54,447.37	0.19	54,005.45
115-670-12	53,860.97	53,653.04	0.39	52,836.14
120-680-12	61,965.16	61,839.74	0.2	60,905.86
Average	48,857.69	48,859.84	0.22	48,182.15

TABLE 2.10: Heuristic vs metaheuristic performance on 30 large scale LIRP instances

Instance	Two-phase heuristic	$GVNS - InvRP_{pVND}$	Gap %
28-320-7	23,688.94	19,431.41	17.97
30-350-7	20,583.44	17,901.95	13.03
30-375-7	45,948.68	31,379.75	31.71
32-380-7	36,242.85	24,134.91	33.41
35-400-7	27,343.06	21,079.49	22.91
37-415-7	38,818.12	26,210.01	32.48
40-420-7	27,067.15	21,472.92	20.67
42-450-7	38,391.9	29,021.97	24.41
45-480-7	34,203.68	26,398.17	22.82
47-490-7	27,656.52	23,264.26	15.88
50-490-9	58,521.38	41,851.58	28.48
52-495-9	103,278.6	66,398.66	35.71
55-500-9	37,453.72	31,817.83	15.05
62-510-9	138,957.9	79,263.62	42.96
65-520-9	38,588.45	32,593.36	15.54
67-540-9	52,051.25	43,064.84	17.26
70-550-9	61,861.2	48,273.41	21.96
74-560-9	37,137.18	33,081.88	10.92
78-570-9	41,172.02	35,799.88	13.05
80-580-9	88,570.99	62,322.33	29.64
85-590-12	81,680.78	69,257.85	15.21
90-595-12	136,311	104,045.38	23.67
92-600-12	94,972.56	78,074.18	17.79
95-610-12	90,975.61	80,957.6	11.01
98-620-12	75,700.98	64,902.25	14.26
100-650-12	55,773.44	51,429.94	7.79
105-655-12	162,519	132,425.78	18.52
110-660-12	58,072.93	54,447.37	6.24
115-670-12	57,275.8	53,653.04	6.33
120-680-12	67,627.68	61,839.74	8.56
Average	61,948.23	48,859.84	19.84

Many companies face complex supply chain optimization problems, such as the LIRP and the LIRPDO and they try to deal with them using simple heuristics, as the two-phase construction heuristic described in subsection 2.4.1. However, results reported in Table 2.10 illustrate clearly that, the use of pure metaheuristic approaches can help companies to improve their cost efficiency. For example, in the case of the large-sized LIRP instances, the $GVNS - InvRPpVND$ has resulted in approximately 20% better solutions (in average) than a simple heuristic.

2.6 Concluding remarks

This chapter considers the optimization of a new complex supply chain problem, the LIRPDO. This problem integrates strategic, tactical, and operational level decisions in order to explore simultaneously their synergistic benefits. Due to its computational complexity, a new metaheuristic solution approach was developed, based on the framework of GVNS. An extensive computational analysis on several large-scale problem instances illustrates the efficiency of the proposed approach in terms of solution quality, especially in large-scale problem instances with potential industrial relevance. Two are the main strengths of the proposed solution approach. The first is its ability to open the minimum required number of depots for satisfying the customers demand (see Table 2.8) . The second one, is the adaptive search strategy, which significantly enhances the performance of the improvement phase in the GVNS component of the proposed solution method. However, the proposed approach does not perform efficiently in small- and medium-sized problem instances.

Chapter 3

Optimization of Pollution-Location-Inventory-Routing Problems

3.1 Introduction

LIRP is a complex NP-hard combinatorial optimization problem, which simultaneously tackles strategic (location/allocation), tactical (inventory levels and replenishment rates) and operational (routing schedules) decisions (Javid & Azad, 2010; Rayat et al., 2017b; Zhang et al., 2014). Its main goal is to determine an optimal schedule for achieving economic benefits, such as total cost minimization (Jabir et al., 2017). However, due to the fact that the supply chain activities emit pollutants, like carbon dioxide (CO_2), the environmental impact of logistics should also be considered (Cheng et al., 2017; Koç et al., 2014). This chapter presents a Pollution-LIRP (PLIRP), which considers both economic and environmental impacts of the main logistic activities, such as facilities location, inventory control and vehicle routing.

3.2 Problem statement

The PLIRP is defined as a two-echelon SCN.
Given:

- a set of time periods,
- a set of potential capacitated depots,
- a set of geographically distributed customers,
- a set of homogeneous capacitated fleet of vehicles,

- a single type of product,
- a period-variable demand of each customer

Determine:

- the number and location of depots to be established,
- the allocation of customers to the opened depots,
- the inventory levels at each customer,
- the replenishment quantities and rates for each customer,
- the routes of vehicles,
- the selection of a speed level for traveling each link of the scheduled network.

In order to: minimize an objective function representing the total cost.

The key model assumption are as follows:

- each customer is serviced by one depot,
- each customer is serviced by at most one vehicle in each time period,
- a vehicle departs from and returns to the same depot after servicing one or more customer(s),
- the total delivered quantity to each customer over the time horizon must be equal to its total demand,

3.3 Mathematical formulation

The problem is formulated as a mixed integer programming model, by integrating an LIRP model (Zhang et al., 2014) with a Green Inventory-Routing problem (G-IRP) model (Cheng et al., 2017). Its sets, parameters and variables are provided in Tables 3.1, 3.2, 3.3, and 3.4.

TABLE 3.1: Sets of the mathematical model

Indices	Explanation
V	set of nodes
J	set of candidate depots
I	set of customers
K	set of vehicles
H	set of discrete and finite planning horizon
R	set of speed levels

TABLE 3.2: Vehicles' parameters

Parameter	Explanation	Value
ϵ	fuel-to-air mass ratio	1[1]
g	gravitational constant (m/s^2)	9.81[1]
ρ	air density (kg/m^3)	1.2041[1]
CR	coefficient of rolling resistance	0.01[1]
η	efficiency parameter for diesel engines	0.45[1]
f_c	unit fuel cost $(Euros/L)$	1.3
f_e	unit CO_2 emission cost $(Euros/kg)$	0.2793[1]
f_d	driver wage $(Euros/s)$	0.0025[1]
σ	CO_2 emitted by unit fuel consumption (kg/L)	2.669[1]
$HVDF$	heating value of a typical diesel fuel (kj/g)	44[1]
ψ	conversion factor (g/s to L/s)	737[1]
θ	road angle	0[1]
τ	acceleration (m/s^2)	0[1]
CW_k	curb weight (kg)	3500[2]
EFF_k	engine friction factor $(kj/rev/L)$	0.25[1]
ES_k	engine speed (rev/s)	39[1]
ED_k	engine displacement (L)	2.77[1]
CAD_k	coefficient of aerodynamics drag	0.6[1]
FSA_k	frontal surface area (m^2)	9[1]
$VDTE_k$	vehicle drive train efficiency	0.4[1]

1: (Cheng et al., 2017) 2: (Koç et al., 2014)

The value of the parameter f_c is calculated as the average of the petrol prices in 40 European countries taken from the site www.globalpetrolprices.com in 26th of February in 2018. The value of parameters f_e and f_d are converted into Euro currency (26th of February, 2018).

TABLE 3.3: Rest PLIRP model parameters

Notation	Explanation
f_j	fixed opening cost of depot j
C_j	storage capacity of depot j
h_i	unit inventory holding cost of customer i
Q_k	loading capacity of vehicle k
d_{it}	period-variable demand of customer i
c_{ij}	distance of locations pair (i, j)
s_r	the value of the speed level r

TABLE 3.4: PLIRP model variables

Notation	Explanation
y_j	1 if j is opened; 0 otherwise
z_{ij}	1 if customer i is assigned to depot j; 0 otherwise
x_{ijkt}	1 if node j is visited after i in period t by vehicle k
q_{ikt}	product quantity delivered to customer i in period t by vehicle k
w_{itp}	quantity delivered to customer i in period p to satisfy its demand in period t
a_{vikt}	load weight by travelling from node v to the customer i with vehicle k in period t
$zz_{v_1v_2ktr}$	1 if vehicle k travels from node $v1$ to $v2$ in period t with speed level r

The objective of the problem represents the minimization of total cost, including the following cost components:

- Location Cost: $\sum_{j \in J} f_j y_j$, which represents the cost of opening the needed number of depots.
- Inventory Cost: $\sum_{i \in I} h_i \sum_{t \in H} \left(\frac{1}{2} d_{it} + \sum_{p \in H, p < t} w_{itp} (t - p) + \sum_{p \in H, p > t} w_{itp} (t - p + |H|) \right)$. It consists of three cost components. The first component represents the average inventory holding cost. The remaining terms impose penalty costs for any early or late replenishment.
- Routing Cost: $\sum_{i \in V} \sum_{j \in V} \sum_{t \in H} \sum_{k \in K} c_{ij} x_{ijkt}$. It represents general routing costs, such as vehicles' maintenance and/or insurance costs.
- Fuel Consumption Cost: $\sum_{i \in V} \sum_{j \in V} \sum_{k \in K} \sum_{t \in H} \left\{ \lambda \left(f_c + (f_e \sigma) \right) \left(\sum_{r \in R} \frac{\left(zz_{ijktr}\ EFF_k\ ES_k\ ED_k\ c_{ij} \right)}{s_r} + \left(\alpha \gamma_k \left(CW_k\ x_{ijkt} + a_{ijkt} \right)\ c_{ij} \right) + \left(\beta_k\ \gamma_k \sum_{r \in R} \left(s_r\ zz_{ijktr} \right)^2 \right) \right) \right\}$. The Comprehensive Modal Emission Model (CMEM) is adopted (Barth et al.,

2005). Thus, the fuel consumption is affected by vehicle specific characteristics, such as the weight of the load, the vehicle's speed and obviously the traveling distance. More specifically, the following formulas are utilized.

- $\lambda = \frac{\epsilon}{HVDF * \psi}$
- $\gamma_k = \frac{1}{1000 \; VDTE \; \eta}$
- $\alpha = \tau + g \; CR \; \sin\theta + g \; CR \; \cos\theta$
- $\beta_k = 0.5 \; CAD \; \rho \; FSA_k$

The first component is the fuel consumption based on the vehicles' engine function, while the second cost term represents the cost of consumed fuel because of the total vehicles' weight (curb weights plus load weights). The third one represents the fuel consumption cost related to the vehicles' speed levels.

- Driver Wages Cost: $\sum_{i \in V} \sum_{j \in V} \sum_{k \in K} \sum_{t \in H} \sum_{r \in R} f_d \frac{(zz_{ijktr} \; c_{ij})}{s_r}$, which represents the cost of the drivers wages.

Thus, the total cost (TC) is calculated as: $TC = Location\ Cost + inventory\ Cost + Routing\ Cost + Fuel\ Consumption\ Cost + Driver\ Wages\ Cost$.
The mathematical formulation, of the problem under consideration, is as follows:

$$\min \quad TC \tag{3.1}$$

Subject to

$$\sum_{r \in R} zz_{ijktr} = 1 \quad \forall i, j \in V, \forall k \in K, \forall t \in H \tag{3.2}$$

$$\sum_{i \in V} a_{ijkt} - \sum_{i \in V} a_{jikt} = q_{jkt} PW \quad \forall j \in I, \forall k \in K, \forall t \in H \tag{3.3}$$

$$\sum_{j \in V} x_{ijkt} - \sum_{j \in V} x_{jikt} = 0 \qquad \forall i \in V, \forall k \in K, \ \forall t \in H \tag{3.4}$$

$$\sum_{j \in V} \sum_{k \in K} x_{ijkt} \leq 1 \quad \forall t \in H, \ \forall i \in I \tag{3.5}$$

$$\sum_{j\in V}\sum_{k\in K} x_{jikt} \leq 1 \quad \forall t \in H, \forall i \in I \tag{3.6}$$

$$\sum_{i\in I}\sum_{j\in J} x_{ijkt} \leq 1 \quad \forall k \in K, \forall t \in H \tag{3.7}$$

$$x_{ijkt} = 0 \quad \forall i, j \in J, \forall k \in K, \forall t \in H, i \neq j \tag{3.8}$$

$$\sum_{i\in I} q_{ikt} \leq Q_k \quad \forall k \in K, \forall t \in H \tag{3.9}$$

$$\sum_{j\in J} z_{ij} = 1 \quad \forall i \in I \tag{3.10}$$

$$z_{ij} \leq y_j \quad \forall i \in I, \forall j \in J \tag{3.11}$$

$$\sum_{i\in I}\left(z_{ij}\sum_{t\in H} d_{it}\right) \leq C_j \quad \forall j \in J \tag{3.12}$$

$$\sum_{u\in I} x_{ujkt} + \sum_{u\in V\setminus\{i\}} x_{iukt} \leq 1 + z_{ij} \quad \forall i \in I, \forall j \in J, \forall \mathrm{k} \in \mathrm{K}, \forall t \in H \tag{3.13}$$

$$\sum_{i\in I}\sum_{k\in K}\sum_{t\in H} x_{jikt} \geq y_j \quad \forall j \in J \tag{3.14}$$

$$\sum_{i\in I} x_{jikt} \leq y_j \quad \forall j \in J, \forall k \in K, \forall t \in H \tag{3.15}$$

$$\sum_{p\in H} w_{itp} = d_{it} \quad \forall i \in I, \forall \mathrm{t} \in \mathrm{H} \tag{3.16}$$

$$\sum_{t\in H} w_{itp} = \sum_{k\in K} q_{ikp} \quad \forall i \in I, \forall p \in H \tag{3.17}$$

$$q_{ikt} \leq M \sum_{j\in V} x_{ijkt} \quad \forall i \in I, \forall t \in H, \forall k \in K \tag{3.18}$$

$$\sum_{j\in V} x_{ijkt} \leq M q_{ikt} \quad \forall i \in I, \forall t \in H, \forall \mathrm{k} \in \mathrm{K} \tag{3.19}$$

$$\mathrm{x}_{ijkt} \in \{0,1\} \quad \forall i \in I, \forall j \in J, \forall t \in H, \forall k \in K \tag{3.20}$$

$$y_j \in \{0,1\} \quad \forall j \in J \tag{3.21}$$

$$z_{ij} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J \tag{3.22}$$

$$q_{ikt} \leq \min\left\{Q_k, \sum_{p \in H} d_{ip}\right\} \quad \forall i \in I,\ \forall j \in J\ ,\forall k \in K \tag{3.23}$$

$$w_{itp} \leq d_{ip} \quad \forall i \in I,\ \forall t, p \in H \tag{3.24}$$

Constraints 4.4 impose that a vehicle travels between nodes with a specific speed level in each time period. Constraints 4.5 act as subtour elimination constraints, as they declare that the difference between the total weight of the incoming flow of product to a selected customer and the total weight of the outcoming product flow of that customer equals the product weight delivered to that customer in the selected time period with the selected vehicle. The equilibrium between the interior and exterior flow of vehicles is guaranteed by Constraints 4.6. Constraints 4.7 and 4.8 ensure that exactly one vehicle visits each customer at each time period. Constraints 4.9 guarantee that a vehicle performs mostly one route at each time period. Constraints 4.10 forbid the movement of a vehicle between depots. Constraints 4.11 ensure that, the total amount of products sent by a vehicle at a specific period does not exceed the capacity of that vehicle. Constraints 4.12 guarantee that a vehicle will be travelled from a depot to a customer only if that customer is allocated to the depot. Constraints 4.13 impose that a customer is assigned to a depot only if that depot is selected to be opened. Constraints 4.14 respect the capacity of depots. A customer is connected to a depot, only if that customer is assigned to that depot, according to Constraints 4.15. A vehicle departures from a depot only if that depot is opened according to Constraints 4.16 and 4.17. The delivered amount of product to each customer at each time period satisfies the demand of that customer, as guaranteed by Constraints 4.18. Constraints 3.17 ensure the equilibrium between scheduled and actual deliveries. A customer is visited at a specific period, only if a replenishment is scheduled for that period, according to Constraints 4.19. The rest constraints of the model declare the nature of the decision variables.

3.4 Solution approach

3.4.1 Construction heuristic

The scope of the constructive phase is the generation of a feasible initial solution. In this work, a three-phased construction heuristic has been developed. In the first phase, location and allocation decisions are made. Inventory-routing decisions are determined in the second phase, while the speed levels for traveling through the network nodes are selected in the last phase.

Initially, for taking the location and allocation decisions, a ratio-based selection criterion is applied for opening the required depots while a nearest customer allocation strategy has been employed for the assignment step. For each one of the candidate depots, the ratio $\frac{fixed\_opening\_cost}{Capacity}$ is computed and the depot with the minimum ratio is chosen. If two or more depots have the same ratio, one of them is selected arbitrarily (commonly the first found). According to the customers' allocation process, the nearest, to the opened depot, customer is chosen and in the case that its total demand does not violate the remaining capacity of the depot, the customer is assigned to that depot. This initial step of construction heuristic is completed when all customers have been allocated to the opened depots.

According to the inventory-routing decisions, for each time period and each selected vehicle, the Random Insertion method is applied, in order to construct its route for visiting all the assigned, to that vehicle, customers once (Glover et al., 2001). The delivered quantities are set equal to the corresponding demand for each customer at each time period. Obviously, in this initial phase if a customer does not require any quantity of the product in a selected period, he will not be included in any route over that period. In the last phase, the selection of the speed levels for traveling through the nodes of the structured network are performed.

3.4.2 Neighborhood structures

Six local search operators are used in the improvement phase of each proposed solution method. These neighborhood structures are the following:

Inter-route Relocate: In this neighborhood structure a selected customer is removed from its route and moved in the next position of an other selected customer, who is assigned to a different route. Those two customers can be allocated to the same or different depots. Both of the selected customers must be visited by vehicles in the same time periods, in order this move to be applicable. A replenishment shifting may be applied if a vehicle capacity violation occurs. The following cases can be met by applying this neighborhood:

- Case 1: The two selected customers are allocated to the same depot and no vehicle capacity violations occur.
- Case 2: The selected customers are assigned to the same depot and vehicles capacity violations occur.
- Case 3: The two selected customers are assigned to different depots and no vehicle capacity violations occur.
- Case 4: The two selected customers are allocated to different depots and vehicles capacity violations occur.

In the first case only routing decisions are made. In the second case, both routing and inventory decisions are taken, while routing and allocation decisions are considered in the third case of this move. In the last case, routing, inventory and allocation decisions are simultaneously made.

Inter-route Exchange: This neighborhood swaps two customers from different routes in the time horizon. The exchanged customers can be allocated either to the same depot or different depots. If the customers are allocated to the same depot, the swapping may not be applied in each time period. However, in the second case the exchanging must be valid for all time periods, in order to be applicable. The special cases of this move are summarized as follows:

- Case 1: Vehicle capacity violation does not occur.
- Case 2: The demand of a customer exceeds the capacity of the vehicle servicing the other customer in one or more time periods.

The first case makes only routing decisions, while in the second case routing and inventory decisions are taken. Changes on allocation decisions will take place only if the swapped customers are allocated to different depots.

Exchange Opened-Closed Depots: This neighborhood exchanges a closed depot with an opened depot. For a selected closed depot, the cost change for swapping it with each one of all opened depots, is calculated. Then, the pair of the opened-closed depots with the minimum exchanging cost is selected. Subsequently, it is examined if the scheduled swapping does not violate any capacity constraint. After the validation checking step, the newly opened depot is inserted in the routes assigned to the currently opened depot via a minimum insertion cost procedure. This insertion process may involve customers re-ordering in the routes. Obviously, the move is applied only if the overall cost decreases, which consists of location and routing costs.

Intra-route Relocate: In this move a selected customer is removed from its current position in its route and moved in a different position in the same route.

2-2 Replenishment Exchange: In this neighborhood structure, two time periods t_1 and t_2 are randomly selected and the two most distant customers i and b, both serviced in those two periods, are identified. The replenishment of customer i in period t_1 is moved to the period t_2, and the replenishment of customer b is moved from period t_2 to period t_1. Consequently, there is no need to visit customers i and b in periods t_1 and t_2 respectively. If the total cost decreases and there are no violations on the vehicles capacities, the move is applied.

In order to avoid potential violation of vehicles capacities, while applying the Inter-route Relocate and the Inter-route Exchange moves, a shifting of surplus product quantity may be applied.

Speed Selection Procedure (SSP): Examines which speed level has the highest fuel cost decrease for each depot-customer and customer-customer pair in the current solution.

3.4.3 Shaking procedure

For escaping local optimum solutions, a shaking procedure with three local search operators is proposed, including the following structures:

- Inter-route Exchange.
- Exchange Opened-Closed Depots.
- Intra-Route Relocate.

In each iteration of this diversification method, one of the proposed local search operators is randomly selected and a predefined number of random jumps are applied. Its pseudo-code is given in Algorithm 8.

Algorithm 8 Shaking Procedure

```
procedure SHAKE(S, k, ShakingN)
    l = random_integer(1, |ShakingN|)
    for i ← 1, k do
        select case(l)
        case(1)
        S' ← Inter − route_Exchange(S)
        case(2)
        S' ← Exchange_OpenedClosed_Depots(S)
        case(3)
        S' ← Intra_Relocate(S)
        end select
    end for
    Return S'
end procedure
```

The shaking procedure receives an incumbent solution S, the maximum number of iterations k_{max} executed in the perturbation phase and the set of shaking neighborhood structures *ShakingN* as input. A new solution S' is obtained by applying k (where $1 < k < k_{max}$) times one randomly selected neighborhood of the above local search operators.

3.4.4 VNS algorithms

In this chapter, a BVNS, two GVNS (GVNS with cVND and GVNS with pVND) solution methods and their corresponding adaptive variants, have been developed. The proposed solution algorithms are provided in the following pseudo-codes.

Algorithm 9 Basic VNS

```
procedure BVNS(k_max, max_time, l_max, ShakingN)
    S ← Construction_Heuristic
    l ← 1
    while time ≤ max_time do
        for each neighborhood structure l do
            for k ← 1, k_max do
                S′ ← Shake(S, k, ShakingN)
                S″ ← Local_Search(S′, l)
                if f(S″) < f(S) then
                    S ← S″
                end if
                l ← l + 1
                if l > l_max then
                    l ← 1
                end if
            end for
        end for
    end while
    Return S
end procedure
```

Algorithm 10 $GVNS_{pVND}$

```
procedure GVNS(S, k_max, max_time, ShakingN)
    S ← Construction_Heuristic
    while time ≤ max_time do
        for k ← 1, k_max do
            S∗ = Shake(S, k, ShakingN)
            S′ = pVND(S∗, l_max)
            if f(S′) < f(S) then
                S ← S′
            end if
        end for
    end while
    return S
end procedure
```

The pseudo-code of the $GVNS_{cVND}$ algorithm is exactly the same with the pseudo-code of the $GVNS_{pVND}$ algorithm with the only difference that it uses the cVND instead of pVND. The adaptive variants of these methods uses an adaptive re-ordering mechanism of the local search operators. This adaptive mechanism uses past experience, such the number of improvements achieved by each operator and proceeds a different order. More specifically, the array "Improvements_Counter" stores in its positions the improvements achieved by each operator and then a descending sorting is applied on this array. The pseudo-code of this mechanism is summarized in Algorithm 11.

Algorithm 11 Adaptive_Order

procedure ADAPTIVE_ORDER($N\_Order, Improvements\_Counter$)
    if no improvement is found in any neighborhood **then**
        Keep the same order
    end if
    if an improvement is found **then**
        $New\_N\_Order \leftarrow Descending\_Order(N\_Order, Improvements\_Counter)$
    end if
    $N\_Order \leftarrow New\_N\_Order$
    return N_Order
end procedure

An example of the adaptive schemes is provided in the following pseudo-code of the Adaptive $GVNS_{pVND}$ algorithm.

Algorithm 12 $AGVNS_{pVND}$

procedure $AGVNS_{pVND}(S, k_{max}, l_{max}, max\_time, N\_Order, Improvements\_Counter, ShakingN)$
    while $time \leq max\_time$ **do**
        $S^* = Shake(S, k, ShakingN)$
        $N\_Order \leftarrow Adaptive\_Order(N\_Order, Improvements\_Counter)$
        $S' = pVND(S^*, l_{max})$
        if $f(S') < f(S)$ **then**
            $S \leftarrow S'$
        end if
    end while
    return S
end procedure

3.5 Computational analysis and results

3.5.1 Computing environment & parameter settings

The proposed methods have been implemented in Fortran (Intel Fortran compiler 18.0 with optimization option /O3) and ran on a desktop PC running Windows 7 Professional 64-bit with an Intel Core i7-4771 CPU at 3.5 GHz and 16 GB RAM. The execution time limit for the proposed algorithms was set at 60s. The PLIRP was also modeled in GAMS (GAMS 24.9.1) (Brooke et al.,1998) and its problem instances were solved using CPLEX 12.7.1.0 solver with the time limit of 2h for the small-sized instances and 5h for medium and large-sized instances. It should be mentioned that CPLEX ran in the same computing environment with Intel Fortran compiler.

3.5.2 Computational results on PLIRP instances

In this chapter 30 new PLIRP instances have been created by following the format of instances proposed in a previous work (Zhang et al., 2014). They are reported in the form X-Y-Z, where X represents the number of potential depots, Y the number of customers and Z is the number of time periods. These instances are available in: http://pse.cheng.auth.gr/index.php/publications/benchmarks. Table 3.5 provides the average and the best found total cost values of ten iterations for each one of the proposed solution methods for different values of the k_{max} parameter.

TABLE 3.5: Shaking strength analysis on the performance of proposed methods

Method	$k_{max} = 5$		$k_{max} = 10$		$k_{max} = 12$	
	Avg. TC	Best TC	Avg. TC	Best TC	Avg. TC	Best TC
BVNS	41,542.67	40,643.76	41,630.07	40,397.15	41,547.22	40,483.04
ABVNS	37,684.82	36,062.44	37,896.45	35,926.57	38,092.11	36,339.28
$GVNS_{pVND}$	35,304.38	34,455.14	**35,055.97**	**34,295.98**	35,403.77	34,559.92
$AGVNS_{pVND}$	35,735.40	34,426.16	36,031.34	34,625.69	36,012.60	34,674.18
$GVNS_{cVND}$	37,261.36	36,487.80	37,395.34	36,570.37	37,281.85	36,348.84
$AGVNS_{cVND}$	37,687.23	36,361.82	37,715.74	36,416.86	37,570.39	36,375.08
Method	$k_{max} = 15$		$k_{max} = 18$		$k_{max} = 20$	
	Avg. TC	Best TC	Avg. TC	Best TC	Avg. TC	Best TC
BVNS	41,549.48	40,407.81	41,495.84	40,681.73	41,724.02	40,557.38
ABVNS	38,227.76	36,312.05	38,184.58	36,627.57	38,504.86	37,158.58
$GVNS_{pVND}$	35,473.07	34,574.27	35,310.72	34,520.85	35,684.59	34,572.27
$AGVNS_{pVND}$	35,783.10	34,608.32	35,814.60	34,565.73	35,735.4	34,563.19
$GVNS_{cVND}$	37,373.07	36,510.59	37,283.29	36,549.10	37,431.34	36,474.75
$AGVNS_{cVND}$	37,522.14	36,331.10	37,784.85	36,233.78	37,717.24	36,290.09

According to the reported solutions, the parameter value $k_{max} = 5$ produces in average the best values for the proposed BVNS, ABVNS and $GVNS_{cVND}$ algorithms. The $AGVNS_{cVND}$ algorithm performs better by using a shaking strength of $k_{max} = 15$, while the $AGVNS_{pVND}$ algorithm produces better solution using the parameter value $k_{max} = 20$. The results achieved using the $GVNS_{pVND}$ with the parameter value $k_{max} = 10$ were the best found solutions in average compared with either the same scheme but with different k_{max} values or the other schemes (18.5% from BVNS, 7.5% from ABVNS, and 6.3% from $GVNS_{cVND}$, 7% from $AGVNS_{cVND}$ and 1.94% from $AGVNS_{pVND}$).

From a problem size perspective, the $AGVNS_{pVND}$ algorithm produces better solutions for small-sized instances (using $k_{max} = 20$) and medium-sized instances (using $k_{max} = 5$) than other approaches, while the $GVNS_{pVND}$ algorithm using the shaking strength $k_{max} = 10$ is more efficient than other methods for the solution of large problem cases.

From this analysis it is noticed that classic GVNS-based methods provide better solutions than their corresponding adaptive variants. An explanation of this conspicuous observation is that using the adaptive re-ordering mechanism leads to further time consumption. Thus, the number of iterations of the improvement phase is significantly decreased for the case of large-sized instances.

The $GVNS_{pVND}$ uses the Speed Selection Procedure after each local search operator. Table 3.6 illustrates potential difference between this GVNS scheme and $GVNS_{pVND}$ which applies the SSP once after the completion of a $pVND$ iteration. The initial scheme is called $GVNS_{pVND}\_1$ and the second one, $GVNS_{pVND}\_2$.

TABLE 3.6: Results achieved by the two $GVNS_{pVND}$ schemes on the 30 PLIRP instances

Instance	$GVNS_{pVND}\_1\_Avg$	$GVNS_{pVND}\_1\_Best$	$GVNS_{pVND}\_2\_Avg$	$GVNS_{pVND}\_2\_Best$
4-9-3	25,336.12	25,131.71	25,142.51	**24,932.36**
4-10-3	20,776.21	**20,649.12**	20,780.24	20,709.14
4-10-5	17,503.66	17,481.13	17,491.73	**17,445.33**
4-12-5	27,127.71	27,069.97	27,133.89	**26,958.58**
4-15-3	16,174.14	16,116.6	16,182.92	**16,070.26**
5-12-3	25,571.71	25,514.4	25,619.41	**25,437.45**
5-15-3	16,430.51	**16,389.54**	16,501.02	16,418.06
5-15-5	20,186.32	**20,058.49**	20,348.43	20,164.08
5-18-3	22,202.75	22,109.4	22,216.44	**22,041.49**
5-20-3	20,023.01	**19,845.57**	20,078.84	19,989.58
6-40-5	24,561.19	**24,116.99**	24,575.04	24,341.58
7-52-5	21,218.05	**20,950.23**	21,270.73	21,159
7-55-7	26,922.81	26,574.72	26,677.95	**26,287.57**
8-60-5	31,514.82	31,076.11	31,428.91	**30,732.43**
8-65-7	48,265.4	47,764.15	48,251.36	**47,599.15**
9-70-5	30,936.66	**30,665.22**	31,419.68	30,691.73
9-75-7	29,260.28	**28,588.11**	30,856.81	29,836.59
9-85-5	28,994.05	**26,784.22**	30,735.91	26,815.18
9-88-7	32,165.91	**31,768.12**	32,015.68	31,844.08
10-90-7	27,532.91	**26,879.52**	27,924.18	27,274.21
15-100-7	15,320.21	15,080.38	15,211.79	**14,990.04**
15-100-10	39,134.05	38,666.61	39,425.27	**38,661.29**
15-120-10	38,945.91	**37,473.08**	41,930.15	40,152.04
20-150-10	41,981.88	40,608.95	41,535.43	**40,381.66**
20-180-12	74,109.8	**73069.77**	75,688.98	74,869.72
25-200-12	75,429.81	72,587.55	74,513.52	**68,146.04**
30-250-10	50,017.53	**47,756.68**	51,379.41	48,755.04
30-270-10	58,450.29	56,708.96	59,376.01	**56,637.84**
35-300-10	73,763.83	71,014.24	71,539.39	**69,361.56**
35-310-12	71,821.7	70,379.8	72,096.53	**68,963.52**
Average	*35,055.97*	34,295.98	35,311.61	**34,255.55**

Despite the fact that, the $GVNS_{pVND}\_2$ scheme produces more best values than the $GVNS_{pVND}\_1$, the $GVNS_{pVND}\_1$ is slightly better in terms of average solution quality. The previous results demonstrate that a further improved solution method can be proposed by adopting a hybrid scheme. More specifically, for the solution of small- and medium-sized instances the $AGVNS_{pVND}$ method with a shaking strength of $k_{max} = 20$ will be applied, while the $GVNS_{pVND}$ using $k_{max} = 10$ will be selected for larger problems. The overall process of the solution method is illustrated in Algorithm 13.

Algorithm 13 $Hybrid\_GVNS_{pVND}$

```
procedure HYBRID ALGORITHM(max_time, l_max)
    S ← Construction_Heuristic
    if small-sized instance then
        k_max ← 20
        S' ← AGVNS_pVND(S, k_max, max_time, l_max)
    else if medium-sized instance then
        k_max ← 5
        S' ← AGVNS_pVND(S, k_max, max_time, l_max)
    else
        k_max ← 10
        S' ← GVNS_pVND(S, k_max, max_time, l_max)
    end if
    return S'
```

The second column of Table 3.7 reports the results achieved by GAMS/CPLEX. As it can be noticed, the CPLEX solver can solve only nine out of ten small-sized instances within a time limit of ($2h$). No solution, using CPLEX, was found for the rest of the problem instances. In the third column, the average objective values achieved by the $Hybrid\_GVNS_{pVND}$, while the fourth column contains the best results achieved by the $Hybrid\_GVNS_{pVND}$ algorithm. The last two columns provide the solution gaps between CPLEX and $Hybrid\_GVNS_{pVND}$. According to the reported results, the $Hybrid\_GVNS_{pVND}$ provides 2.4% better results than CPLEX (approximately 3% focused on the best found solutions of the $Hybrid\_GVNS_{pVND}$ algorithm). Based on the fact that, the CPLEX solver cannot provide any feasible solution for medium- and large-sized instances even within a time limit of $5h$ and its high computational time for finding the reported solutions in the case of the small-sized instances it can be concluded that, the proposed $Hybrid\_GVNS_{pVND}$ algorithm is an efficient method for solving large-scale PLIRP instances.

TABLE 3.7: Comparative analysis between CPLEX and $Hybrid\_GVNS_{pVND}$ on the small- and medium-sized instances

Instance	CPLEX (a)	$Hybrid\_GVNS_{pVND}\_Avg$ (b)	$Hybrid\_GVNS_{pVND}\_Best$ (c)	$Gap(b-a)(\%)$	$Gap(c-a)(\%)$
4-9-3	25,182.74	25,065.69	24,775.13	0.46	1.62
4-10-3	19,908.16	20,787.31	20,619.48	-4.42	-3.52
4-10-5	17,786.65	17,502.38	17,457.5	1.6	1.85
4-12-5	26,741.55	27,053.31	26,958.67	-1.17	-0.81
4-15-3	15,370.4	16,125.36	15,968.86	-4.91	-3.89
5-12-3	25,353.81	25,517.97	25,451.99	-0.65	-0.39
5-15-3	18,670.19	16,422.19	16,406.91	12.04	12.12
5-15-5	N/A	18,224.79	18,084.47	-	-
5-18-3	26,353.47	22,078	21,982.96	16.22	16.58
5-20-3	N/A	20,124.22	19,878.09	-	-
6-40-5	N/A	24,872.74	24,090.17	-	-
7-52-5	N/A	21,093.97	20,810.7	-	-
7-55-7	N/A	26,702.27	26,470.52	-	-
8-60-5	N/A	31,436.84	31,249.98	-	-
8-65-7	N/A	47,497.09	46,207.73	-	-
9-70-5	N/A	30,978.96	30,310.59	-	-
9-75-7	N/A	29,266.82	28,533.75	-	-
9-85-5	N/A	27,883.9	27,014.67	-	-
9-88-7	N/A	31,860.02	31,750.8	-	-
10-90-7	N/A	27,129.59	26,708.53	-	-

Table 3.8 reports the best known solutions of the 30 PLIRP instances.

TABLE 3.8: Best known solutions for the 30 PLIRP instances

Instance	BKS	Instance	BKS
4-9-3	24,775.13	9-70-5	30,310.59
4-10-3	20,619.48	9-75-7	28,533.75
4-10-5	17,457.5	9-85-5	27,014.67
4-12-5	26,958.67	9-88-7	31,750.8
4-15-3	15,968.86	10-90-7	26,708.53
5-12-3	25,451.99	15-100-7	15,080.38
5-15-3	16,406.91	15-100-10	38,666.61
5-15-5	20,012.77	15-120-10	37,473.08
5-18-3	21,982.96	20-150-10	40,608.95
5-20-3	19,878.09	20-180-12	73,069.77
6-40-5	24,090.17	25-200-12	72,587.55
7-52-5	20,810.7	30-250-10	47,756.68
7-55-7	26,470.52	30-270-10	56,708.96
8-60-5	31,249.98	35-300-10	71,014.24
8-65-7	46,207.73	35-310-12	70,379.8

Table 3.9 summarizes the number of the vehicles used for each problem instance using the $Hybrid\_GVNS_{pVND}$ algorithm. According to the number of opened depots, all the proposed methods open exactly two depots in each problem instance. Based on the total demand of customers, two depots are the minimum required for fulfilling customers demands per instance. A replenishment policy highly impacts the produced solutions. Furthermore, a flexible replenishment policy enables the building of cost efficient routing patterns (Zachariadis et al., 2009; Zhang et al., 2014).

TABLE 3.9: Number of used vehicles per instance using $Hybrid\_GVNS_{pVND}$

Instance	#Vehicles	Instance	#Vehicles	Instance	#Vehicles
4-9-3	4	6-40-5	7	15-100-7	4
4-10-3	5	7-52-5	4	15-100-10	4
4-10-5	3	7-55-7	3	15-120-10	4
4-12-5	3	8-60-5	13	20-150-10	5
4-15-3	3	8-65-7	28	20-180-12	9
5-12-3	6	9-70-5	8	25-200-12	8
5-15-3	4	9-75-7	3	30-250-10	4
5-15-5	10	9-85-5	5	30-270-10	5
5-18-3	8	9-88-7	4	35-300-10	11
5-20-3	5	10-90-7	3	35-310-12	4

Figure 3.1 illustrates the effect of flexible replenishment policy on the routing and the inventory costs. More specifically, the values of routing and inventory costs reported in the successful iterations of the $Hybrid\_GVNS_{pVND}$ are depicted. Routing costs can be decreased by allowing flexible reorder points and order quantities for the customers due to the reduction of deliveries or the efficient clustering of customers into routes. For instance, the deliveries to a distant customer can be reduced by replenishing it with more product quantities in less time periods. Also, this shifting of product quantities results on more available space in the vehicles, so more customers can be serviced by the same vehicle. This can lead to cost efficient routing circuits. On the other hand, the deferred deliveries leads to an increase in the inventory cost.

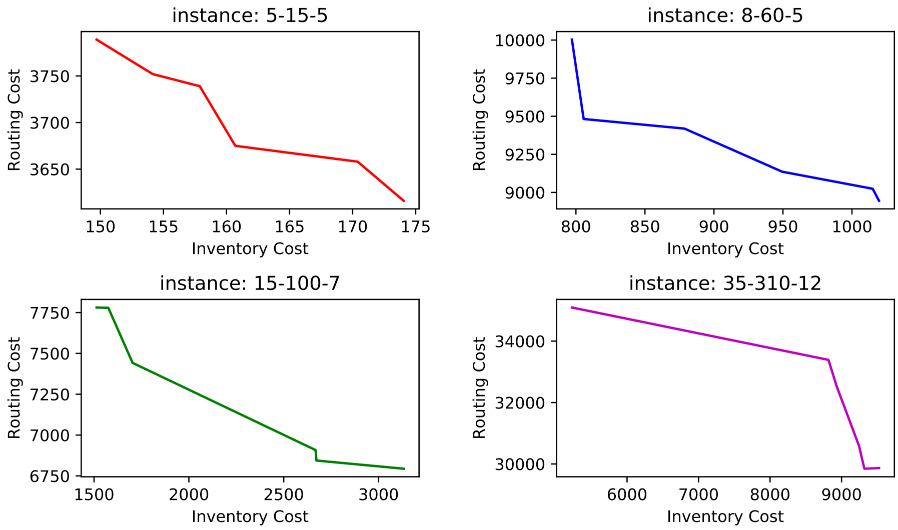

FIGURE 3.1: Effect of flexible replenishment policy on the relationship between routing and inventory costs.

It should be mentioned that, the solutions obtained by the $Hybrid\_GVNS_{pVND}$ algorithm do not always follow the previous relation. In some cases, the algorithm splits the routes in order to reduce the routing cost without increasing the inventory cost. However, the splitting strategy leads to the usage of more vehicles.

3.5.3 Sensitivity analysis

To further consider the significance of the flexible replenishment policy, the effect of holding costs on the total cost is studied through a parametric analysis. Holding costs are crucial for the performance of logistics design and operation. In the literature, several works studied the effect of holding cost variations on the overall performance of logistic systems (Alfares & Ghaithan, 2016; Hu et al., 2018). In this work, two testing scenarios are considered. In the first one, a holding cost increase by 10% is examined, while in the second one the holding cost is increased by 15%. The $Hybrid\_GVNS_{pVND}$ algorithm, which marked as the most efficient between the proposed solution methods, was used for solving the problem instances in these two scenarios. Table 3.10 provides both the average and best found results per scenario.

TABLE 3.10: The average and best found results in sensitivity analysis scenarios

Instance	*OV_Avg*_10%	*OV_Best*_10%	*OV_Avg*_15%	*OV_Best*_15%
4-9-3	25,181.77	25,041.14	25,188.46	25,031.23
4-10-3	20,863.64	20,781.04	20,848.53	20,793.92
4-10-5	17,520.59	17,432.05	17,525.54	17,475.08
4-12-5	27,082.38	26,984.32	27,107.96	27,029.25
4-15-3	16,126.86	16,044.12	16,119.94	15,980.26
5-12-3	25,598.02	25,485.53	25,519.32	25,485.46
5-15-3	16,435.71	16,399.19	16,435.47	16,419.09
5-15-5	20,285.74	19,992.44	20,357.95	20,184.11
5-18-3	22,096.39	22,014.08	22,096.62	21,975.62
5-20-3	20,033.03	19,932.21	20,021.84	19,940.18
6-40-5	25,032.92	24,918.66	24,720.14	24,404.99
7-52-5	21,321.34	21,167.21	21,383.7	21,154.03
7-55-7	27,103.04	26,566.88	26,828.2	26,664.36
8-60-5	31,558.35	30,769.91	31,495.14	31,157.83
8-65-7	48,084.16	47,074.4	48,907.25	48,480.28
9-70-5	31,429	30,582.42	31,426.12	30,949.3
9-75-7	29,717.7	29,010.54	30,748.95	29,495.22
9-85-5	29,476.81	27,049.91	29,153.98	27,418.47
9-88-7	32,205.27	31,682.29	32,245.75	32,156.69
10-90-7	28,073.3	27,728.77	28,111.57	27,222.74
15-100-7	15,949.38	15,378.73	15,627.67	15,129.28
15-100-10	40,619.23	39,295.37	39,715.32	38,962.52
15-120-10	40,926.89	39,351.76	42,157.16	40,537.22
20-150-10	42,974.34	41,330.91	45,801.18	41,101.49
20-180-12	75,943.7	73,496.76	75,164.3	73,597.62
25-200-12	77,760.61	73,258.41	78,961.45	73,650.02
30-250-10	53,034.67	50,785.36	52,581.2	50,516.56
30-270-10	60,394.23	56,942.57	59,424.19	57,147.17
35-300-10	77,476.23	73,685.28	77,152.02	71,780.28
35-310-12	72,246.86	69,485.7	74,525.83	69,823.05
Average	35,751.74	34,655.6	35,911.76	34,722.11

The results indicate that, the second scenario produces 0.45% worse solutions in average comparing to the first one. Moreover, the initial average solutions of the instances are 1.98% and 2.44% better than those achieved in the first and the second scenario respectively, which means that the objective value seems to be sensitive on the changes of the holding costs. However, the adoption of a flexible replenishment policy keeps the increase of the cost in relatively low levels.

Also, there are some instances in which better solutions were produced by increasing the holding costs. This is justified as, an increase in the holding costs, forces the usage of more vehicles (max two more than the reported vehicles in Table 3.9) in order to form better routing patterns, thus leading to further routing cost reduction.

3.6 Concluding remarks

This chapter presents a new green SCN optimization problem, which considers both economic and environmental concerns. GVNS- and Adaptive GVNS-based solution approaches have been developed for the efficient solution of medium- and large-sized instances. A new set of 30 random generated instances is used in extended numerical analyses. A computational k_{max} parameter analysis has been made. This analysis indicates that, the GVNS scheme which uses the pVND ($GVNS_{pVND}$) method in its improvement process is proved to be the most efficient method. In addition, the effect of executing the Speed Selection Procedure either after each local search operator or in the end of each VND iteration is tested. The GVNS scheme which uses the Speed Selection Procedure after each local search operator proved as the most efficient solution method. However, from a problem size perspective, a hybrid solution approach, which uses the Adaptive GVNS ($AGVNS_{pVND}$) algorithm with different k_{max} values for the solution of small- and medium-sized instances and the $GVNS_{pVND}$ for solving large problem cases. This hybrid solution scheme is compared with the CPLEX solver in ten small-sized instances. The proposed solution method produces approximately 3% better solutions than CPLEX. Finally, a sensitivity analysis is performed to study the effect of the variations of holding costs on the objective value. The results illustrate that, any increase on the holding costs affects the objective value. However, the use of the flexible replenishment policy keeps the increase of total cost in relatively low levels. Some exceptions are noticed by using more vehicles.

Chapter 4

Optimization of Fleet Size & mix Pollution-Location-Inventory-Routing Problems with Just in Time and Capacity Choices

4.1 Introduction

The importance of selecting an appropriate replenishment policy in the green IRP problem has been clearly highlighted in previous studies. The Just-in-Time (JiT) replenishment policy is a popular inventory management strategy based on the lean management philosophy and the increased customer satisfaction. Recent studies have shown that this policy positively affects the sustainable performance of a company (Kong et al., 2018; Wang & Ye, 2018). This is due to the elimination of storage activities and consequently the relative waste. Moreover, the significance of facilities-related decisions on a company's sustainable performance is indisputable. That being said, the capacity planning of facilities is also critical for achieving sustainability due to its strategic nature (Aldis, 2017).

This work addresses a new variant of the LIRP, the Fleet-size and Mix Pollution-LIRP with JiT replenishment policy and capacity planning (FSM-PLIRP). This new NP-hard problem considers further strategic level decisions, such as the capacity planning and fleet composition. The JiT replenishment policy is the only appropriate in some emergency supply chain networks, such as the medical supply chains. However, it reduces the flexibility on route scheduling and makes even harder the effort of building efficient routes. For the efficient solution of the underlying problem, the development of problem-specific solution methods is crucial. The capability of an algorithm to use past experience in order to improve its performance is a key feature of intelligent optimization. Therefore, in this work we develop General Variable Neighborhood Search (GVNS) metaheuristic algorithms within adaptive shaking mechanisms in an effort to improve their performance and solve a supply chain problem of significant industrial interest. The proposed modeling framework and solution approaches can provide the basis for the development of an expert system that can assist decision makers to derive rigorous and fast decisions related to the operation and design of such supply chains.

4.2 Problem statement

The FSMPLIRP is defined as a complete graph $G = (V, E)$, where V denotes the set of nodes including both the set of customers $I = \{1, ..., N_{Customers}\}$ and the set of potential depots $J = \{1, ..., N_{Depots}\}$ and $E = \{(v, v_1) : v, v_1 \in V, v \neq v_1\}$ is the set of edges. Each customer has a period-dependent demand for a single-type of product and it is served by a heterogeneous fleet of vehicles. A vehicle has a fixed usage cost and a specific capacity level. A mixed integer programming (MIP) model is proposed to describe this problem. The model is an extension of the previously proposed PLIRP formulation and it tackles a more complex variant of the LIRP by considering facility capacity planning, fleet composition and JiT replenishment policy. The use of JiT replenishment policy means that the delivered and the demanded quantity of product must be equal for each customer in each time period. Thus, the formation of efficient routes gets even harder than the case of PLIRP.

4.3 Mathematical formulation

For clarity reason the model sets, parameters and variables are provided in Tables 4.1, 4.2, 4.3 and 4.4.

TABLE 4.1: Sets of the mathematical model

Indices	Explanation
V	set of nodes
J	set of candidate depots
I	set of customers
K	set of vehicles
H	set of discrete and finite planning horizon
R	set of speed levels
L	set of capacity levels

TABLE 4.2: Vehicles' parameters.

Parameter	Explanation	Value (Cheng et al., 2017; Koç et al., 2014)
ϵ	fuel-to-air mass ratio	1
g	gravitational constant (m/s^2)	9.81
ρ	air density (kg/m^3)	1.2041
CR	coefficient of rolling resistance	0.01
η	efficiency parameter for diesel engines	0.45
f_c	unit fuel cost $(Euros/L)$	0.7382
f_e	unit CO_2 emission cost $(Euros/kg)$	0.2793
f_d	driver wage $(Euros/s)$	0.0025
σ	CO_2 emitted by unit fuel consumption (kg/L)	2.669
$HVDF$	heating value of a typical diesel fuel (kj/g)	44
ψ	conversion factor (g/s to L/s)	737
θ	road angle	0
τ	acceleration (m/s^2)	0
CW_k	curb weight (kg)	3500 or 5500
EFF_k	engine friction factor $(kj/rev/L)$	0.25 or 0.2
ES_k	engine speed (rev/s)	39 or 33
ED_k	engine displacement (L)	2.77 or 5
CAD_k	coefficient of aerodynamics drag	0.6
FSA_k	frontal surface area (m^2)	9
$VDTE_k$	vehicle drive train efficiency	0.4 or 0.45
Q_k	loading capacity of vehicle k	instance-dependent
VFC_k	usage cost of vehicle k	1200 or 1400

The value of parameters f_e and f_d are converted into Euro currency (26th of February, 2018). The usage cost for light-duty vehicles taken as 1200 Euros and for the case of medium-duty vehicles is 1400 Euros. Generally, if a cell includes two values, the first refers to light-duty trucks and the second to medium-duty trucks.

TABLE 4.3: Non-vehicle related FSMPLIRP model parameters.

Notation	Explanation
f_{jl}	fixed opening cost of depot j with capacity level l
C_{jl}	storage capacity of depot j with capacity level l
h_i	unit inventory holding cost of customer i
d_{it}	period-variable demand of customer i
c_{ij}	travelling cost of locations pair (i, j)
s_r	the value of the speed level r

TABLE 4.4: GLIRP model variables.

Notation	Explanation
y_{jl}	1 if depot j with capacity level l is opened; 0 otherwise
z_{ij}	1 if customer i is assigned to depot j; 0 otherwise
vs_{kt}	1 if vehicle k is selected in period t; 0 otherwise
x_{ijkt}	1 if node j is visited after i in period t by vehicle k
q_{ikt}	product quantity delivered to customer i in period t by vehicle k
a_{vikt}	load weight by travelling from node v to the customer i with vehicle k in period t
$zz_{v_1 v_2 ktr}$	1 if vehicle k travels from node $v1$ to $v2$ in period t with speed level r

Also in this chapter, the comprehensive fuel consumption model is adopted (Barth et al., 2005).

$$\min \sum_{j \in J} f_{jl} y_{jl} + \sum_{i \in I} h_i \sum_{t \in H} \frac{1}{2} d_{it} \;\; + \sum_{i \in V} \sum_{j \in V} \sum_{t \in H} \sum_{k \in K} c_{ij} x_{ijkt}$$

$$+ \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} \sum_{t \in H} \left\{ \lambda \left(f_c + (f_e \sigma) \right) \left(\sum_{r \in R} \frac{\left(zz_{ijktr}\ EFF_k\ ES_k\ ED_k\ c_{ij} \right)}{s_r} \right. \right.$$

$$\left. \left. + \left(\alpha \gamma_k \left(CW_k\ x_{ijkt} + a_{ijkt} \right)\ c_{ij} \right) + \left(\beta_k\ \gamma_k \sum_{r \in R} \left(s_r\ zz_{ijktr} \right)^2 \right) \right) \right\}$$

$$+ \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} \sum_{t \in T} \sum_{r \in R} f_d \frac{\left(zz_{ijktr}\ c_{ij} \right)}{s_r} + \sum_{k \in K} \sum_{t \in H} vs_{kt} VFC_k (4.1)$$

Subject to

$$vs_{kt} \leq \sum_{v \in V} \sum_{v_1 \in V} x_{vv_1kt}, \quad \forall k \in K, \forall t \in H, v \neq v_1 \tag{4.2}$$

$$x_{vv_1kt} \leq vs_{kt} \quad \forall v, v_1 \in V, v \neq v_1, \forall k \in K, \forall t \in H \tag{4.3}$$

$$\sum_{r \in R} zz_{ijktr} = 1 \quad \forall i, j \in V, \forall k \in K, \forall t \in H \tag{4.4}$$

$$\sum_{i\in V} a_{ijkt} - \sum_{i\in V} a_{jikt} = q_{jkt}PW \quad \forall j \in I, \forall k \in K, \forall t \in H \tag{4.5}$$

$$\sum_{j\in V} x_{ijkt} - \sum_{j\in V} x_{jikt} = 0 \qquad \forall i \in V, \forall k \in K, \ \forall t \in H \tag{4.6}$$

$$\sum_{j\in V}\sum_{k\in K} x_{ijkt} \leq 1 \quad \forall t \in H, \ \forall i \in I \tag{4.7}$$

$$\sum_{j\in V}\sum_{k\in K} x_{jikt} \leq 1 \quad \forall t \in H, \ \forall i \in I \tag{4.8}$$

$$\sum_{i\in I}\sum_{j\in J} x_{ijkt} \leq 1 \quad \forall k \in K, \ \forall t \in H \tag{4.9}$$

$$x_{ijkt} = 0 \quad \forall i, j \in J, \ \forall k \in K, \ \forall t \in H, \ i \neq j \tag{4.10}$$

$$\sum_{i\in I} q_{ikt} \leq Q_k \quad \forall k \in K, \ \forall t \in H \tag{4.11}$$

$$\sum_{j\in J} z_{ij} = 1 \quad \forall i \in I \tag{4.12}$$

$$z_{ij} \leq y_{jl} \quad \forall i \in I, \ \forall j \in J \tag{4.13}$$

$$\sum_{i\in I}\left(z_{ij} \sum_{t\in H} d_{it} \right) \leq C_{jl} \quad \forall j \in J, \forall l \in L \tag{4.14}$$

$$\sum_{u\in I} x_{ujkt} + \sum_{u\in V\setminus\{i\}} x_{iukt} \leq 1 + z_{ij} \quad \forall i \in I, \ \forall j \in J, \ \forall \mathrm{k} \in \mathrm{K}, \forall t \in H \tag{4.15}$$

$$\sum_{i\in I}\sum_{k\in K}\sum_{t\in H} x_{jikt} \geq y_{jl} \quad \forall j \in J, \forall l \in L \tag{4.16}$$

$$\sum_{i\in I} x_{jikt} \leq y_{jl} \quad \forall j \in J, \forall l \in L, \forall k \in K, \ \forall t \in H \tag{4.17}$$

$$\sum_{k\in K} q_{ikt} = d_{it}, \quad \forall i \in I, \forall t \in H \tag{4.18}$$

$$q_{ikt} \leq M \sum_{j\in V} x_{ijkt} \quad \forall i \in I, \ \forall t \in H, \ \forall k \in K \tag{4.19}$$

$$\sum_{j \in V} x_{ijkt} \leq Mq_{ikt} \quad \forall i \in I,\ \forall t \in H,\ \forall \mathrm{k} \in \mathrm{K} \tag{4.20}$$

$$\mathrm{x}_{ijkt} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J,\ \forall t \in H,\ \forall k \in K \tag{4.21}$$

$$y_{jl} \in \{0,1\} \quad \forall j \in J, \forall l \in L \tag{4.22}$$

$$z_{ij} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J \tag{4.23}$$

$$q_{ikt} \leq \min\{Q_k, d_{it}\} \quad \forall i \in I,\ \forall k \in K\ , \forall t \in H \tag{4.24}$$

The objective criterion of this model is the minimization of the total cost which consists of the facilities' opening costs, the average inventory holding costs, general routing costs, fuel and CO_2 emissions costs, driver wages and vehicle usage costs. Constraints 4.2 ensure that a vehicle is selected in a period only if a route has been scheduled for it in that period. Constraints 4.3 guarantee that a vehicle will move through a pair of nodes in a period, only if it is selected in that period. Constraints 4.4 impose the selection of a specific speed level for traveling through two nodes in each time period. Constraints 4.5 satisfy the product flow balance and simultaneously act as subtour elimination constraints. Constraints 4.6 guarantee the equilibrium between the interior and exterior flow of vehicles. A customer will be serviced by one vehicle at most in each time period, as it is imposed by Constraints 4.7 and 4.8. Constraints 4.9 force a vehicle to not perform more than one route per time period. Constraints 4.10 ensure that a vehicle will not move through two depot locations. The product quantity delivered with a vehicle must not exceed its capacity, as it is imposed by Constraints 4.11. According to Constraints 4.12 a vehicle will move from a depot to a customer, only if that customer is assigned to the depot. A customer can be assigned only to an open depot based on Constraints 4.13. Constraints 4.14 ensure the observance of depots' capacities. Constraints 4.15 impose a customer to connect with a depot, only if that customer is allocated to that depot. A vehicle departures from a depot only if that depot is opened according to Constraints 4.16 and 4.17. **??** guarantee that the demand of each customer will be satisfied in each period. A customer is visited at a specific period, only if a replenishment is scheduled for that period, according to Constraints 4.19. The last four set of constraints declare the nature of the decision variables.

4.4 Solution approach

4.4.1 Initial solution

To build an initial feasible solution, a three-phase construction method is proposed. Location, capacity planning and allocation decisions are made in its first phase. In an effort to find the minimum required number of depots, a ratio-based depots' selection method is applied, as presented in the previous chapter. If more than one depots are needed for servicing the given customers, then a nearest customer allocation procedure is used for each opened depot. In the next phase, the deliveries are set equal to their corresponding demands for each customer in each time period and the routes are built by applying a modified Nearest Neighbor heuristic (Flood, 1956). Finally, the speed levels for traveling through the links of the designed network are randomly set.

4.4.2 Local search operators & pVND

This section describes eight local search operators which are designed to explore the solutions of the corresponding neighborhood structures. These operators are the following:

Inter-route Relocate (N_1). This operator selects two customers assigned to different routes. Then, it removes the first selected customer from its current position and relocates it to the next position of the second selected customer. The initial routes of the two selected customers can either be assigned to the same depot or different depots.

Opened-Closed Depots Exchange (N_2). In this operator for each closed depot the maximum capacity level is selected and examined if that depot can replace one of the currently opened depots. It is mainly examined if the capacity of the closed depot is enough to deal with the total demand of customers allocated to the, potentially to be exchanged, opened depot.

Intra-route Relocate (N_3). This operator selects two customers allocated to the same depot and moves the first selected customer from its current position to the next position of the second selected customer.

Inter-route Exchange (N_4). This operator swaps two selected customers which they are assigned to different routes. Similarly to the Inter-route Relocate, the routes can be allocated to the same depot or not.

Intra-route 2-Opt (N_5). It selects two pairs of successive customers, assigned to the same route, (i, j) and (k, l). Next, it breaks them and reconnects them differently, such as (i, k) and (j, l).

One Medium-Two Light Vehicles Exchange (N_6). This operator selects two currently used light-duty vehicles and examines if the serviced, by those vehicles, customers can be serviced by one unselected medium-duty vehicle.

Select Depot Capacity Level (N_7). In this operator the most cost-efficient capacity level is selected for each opened depot with respect to the total demand of its customers.

Medium-To-Light Vehicles Exchange (N_8). This operator selects a used medium-duty vehicle and examines if the total demand of its customers can be serviced by a light-duty vehicle, in order to perform an exchange between those two vehicles.

These local search operators are included in two pVND methods. The first method contains operators $N_1 - N_5$, while the second one contains operators $N_1 - N_6$. The pVND is selected due to its efficiency in solving hard optimization problems, as it is highlighted in previous chapters. An adaptive search strategy is also adopted (Best improvement is applied on small- and medium-sized problem instances, while first improvement is used for the case of large-sized instances). The pseudocodes of the proposed pVND schemes are summarized in Algorithms 14 and 15.

Algorithm 14 pVND 1

```
procedure pVND_1(S, l_max)
    l = 1
    while l ≤ l_max do
        select case(l)
        case(1)
        S' ← N_1(S)
        case(2)
        S' ← N_2(S)
        case(3)
        S' ← N_3(S)
        case(4)
        S' ← N_4(S)
        case(5)
        S' ← N_5(S)
        end select
        if f(S') < f(S) then
            S ← S'
        else
            l = l + 1
        end if
    end while
    Return S
end procedure=0
```

Algorithm 15 pVND 2

```
procedure PVND_2(S, l_max)
    l = 1
    while l ≤ l_max do
        select case(l)
        case(1)
        S' ← N_1(S)
        case(2)
        S' ← N_2(S)
        case(3)
        S' ← N_3(S)
        case(4)
        S' ← N_4(S)
        case(5)
        S' ← N_5(S)
        case(6)
        S' ← N_6(S)
        end select
        if f(S') < f(S) then
            S ← S'
        else
            l = l + 1
        end if
    end while
    Return S
end procedure
```

Operators N_7 and N_8 are applied within the pVND methods as an integrated improvement phase.

4.4.3 Shaking procedures

Diversification methods are critical components of metaheuristic algorithms (Xu & Cai, 2018). They are strategies for escaping from local optimum solutions by using properly modified local search operators. Here, five shaking operators are designed:

- **Inter-route Exchange Shaking (S_1).** It works as the local search operator N_4 with the difference that the two customers are selected randomly.

- **Opened-Closed Depots Exchange (S_2).** This shaking operator functions similar to N_2. The main difference is that the closed depot is selected randomly.
- **Intra-route Relocate (S_3).** In this operator two customers are randomly selected in each time period. Then, this shaking operator performs like as N_3.
- **Select Depot Capacity Level Shaking (S_4).** This operator selects randomly an opened depot and changes the capacity level of that depot, with respect to the total demand of the customers serviced by it.
- **Light2Medium Vehicles Exchange Shaking (S_5).** Initially, a time period is randomly selected and then a selected light-duty vehicle is exchanged with a medium-duty vehicle.

The above operators are embedded in two shaking procedures (the first does not include the S_5). Their pseudocodes are provided in Algorithms 16 and 17.

Algorithm 16 Shaking procedure 1

procedure SHAKE_1(S, l)
    select case(l)
    case(1)
    $S' \leftarrow S_1(S)$
    case(2)
    $S' \leftarrow S_2(S)$
    case(3)
    $S' \leftarrow S_3(S)$
    case(4)
    $S' \leftarrow S_4(S)$
    end select
    Return S'
end procedure

Algorithm 17 Shaking procedure 2

```
procedure SHAKE_2(S, l)
    select case(l)
    case(1)
    S' ← S_1(S)
    case(2)
    S' ← S_2(S)
    case(3)
    S' ← S_3(S)
    case(4)
    S' ← S_4(S)
    case(5)
    S' ← S_5(S)
    end select
        Return S'
end procedure
```

The most commonly used diversification method within VNS is the intensified shaking, which randomly selects a shaking operator and applies it k times, where k denotes the intense of diversification and it is $1 \leq k \leq k_{max}$, with k_{max} being the shaking strength. Additional to the intensified shaking, this work proposes two adaptive shaking procedures. Initially, the five shaking operators are ordered in a set. According to that initial order, two adaptive shaking procedures are formed. In the first procedure the initial order of operators is based on their computational complexity, while in the second one their ordering is performed randomly. However, both of them are executed similarly. More specifically, in each GVNS iteration and for a specific k value, the shaking operators are executed sequentially (shaking operator - pVND - solution renewal check). A five positions array is used to count the improvements, achieved by using each shaking operator. Each position is matched with one shaking operator and in case of finding a new best solution, the value in this position is increased by one. In the next iteration of GVNS, the sequence of shaking operators is re-ordered according to the number of improvements recorded in the previous iteration. If no improvements or the same number of improvements are achieved during an iteration, the initial order is adopted for the next iteration. Essentially, the core difference between the adaptive shaking schemes and the intensified shaking lies in the manner the shaking operators are handled.

Focused on the adaptive shaking strategies, a reduced scheme is also examined. In particular, in each GVNS iteration, different shaking operators are applied for different k values. For instance, for $k = 1$, the first shaking operator is applied, for $k = 2$, the next operator and so on. If all operators are applied and variable k has not reached the k_{max} value, the diversification process will continue from the first operator. In each next GVNS iteration, the re-ordering step is applied such as in the previously discussed adaptive shaking strategies.

4.4.4 GVNS schemes

The use of different components leads to different GVNS schemes. Moreover, the structure of numerical analyses may impose the formation of further GVNS schemes. From a problem solution perspective, two cases of GVNS schemes are met:

- Case_1: GVNS schemes for solving the homogeneous case of the problem.
- Case_2: GVNS schemes for solving the heterogeneous case of the problem.

From a shaking strategy perspective, three cases of GVNS schemes are investigated:

- Case_1: GVNS schemes which use the intensified shaking.
- Case_2: GVNS schemes which use the adaptive shaking method with complexity-based initialization.
- Case_3: GVNS schemes that they use the adaptive shaking method with random initial order.

Therefore, the following main GVNS are defined:

- $GVNS\_1$: This heuristic is proposed for solving the homogeneous case of the problem and uses the intensified shaking as its diversification strategy.
- $GVNS\_2$: This GVNS scheme solves the same problem case as the $GVNS\_1$, but it uses the adaptive shaking with complexity-based initialization.
- $GVNS\_3$: An other heuristic for solving the homogeneous case of the problem which uses the adaptive shaking with random initial order.
- $GVNS\_4$: This GVNS scheme is proposed for solving the heterogeneous case of the problem. The intensified shaking is used.

- $GVNS\_5$: This heuristic solves the heterogeneous case of the problem and uses the adaptive shaking with complexity-based initialization.
- $GVNS\_6$: This GVNS heuristic solves the heterogeneous case of the problem and the adaptive shaking with random-based initialization is used.

The pseudocodes of the first three GVNS schemes are provided in Algorithms 19, 20 and 21. However, before the presentation of these pseudocodes, the re-ordering mechanism of shaking operators is provided in Algorithm 18. The *ShakingOrder* is the ordered set of shaking operators, *InitialOrder* keeps the initial order of the shaking operators and *ShakingOperatorsChecked* is a logical array which indicates if a shaking operator is selected during the re-ordering phase.

Algorithm 18 Re-ordering mechanism

```
procedure ADAPTIVE_ORDER(ShakingOrder, InitialOrder)
    if no improvement is found in any neighborhood then
        ShakingOrder = InitialOrder
    end if
    if an improvement is found then
        for i ← 1,5 do
            l = Operator with maximum number of improvements
            ShakingOperatorChecked(l) = .true.
            ShakingOrder(i) = l
        end for
    end if
    return ShakingOrder
end procedure
```

Algorithm 19 GVNS 1

procedure GVNS_1($S, k_{max}, max\_time, l_{max}$)
    while $time \leq max\_time$ **do**
        for $k \leftarrow 1, k_{max}$ **do**
            $S^* = Shake\_1(S, l)$
            $S' = pVND\_1(S^*, l_{max})$
            $S^* = N_7(S')$
            if $f(S^*) < f(S)$ **then**
                $S \leftarrow S^*$
            end if
        end for
    end while
    return S
end procedure

Algorithm 20 GVNS 2

procedure GVNS_2($S, k_{max}, max\_time, l_{max}$)
    while $time \leq max\_time$ **do**
        $ShakingOrder = Adaptive\_Order(ShakingOrder, InitialOrder)$
        for $k \leftarrow 1, k_{max}$ **do**
            for $i \leftarrow 1, 5$ **do**
                $l = ShakingOrder(i)$
                $S^* = Shake\_1(S, l)$
                $S' = pVND\_1(S^*, l_{max})$
                $S^* = N_7(S')$
                if $f(S^*) < f(S)$ **then**
                    $S \leftarrow S^*$
                end if
            end for
        end for
    end while
    return S
end procedure

Algorithm 21 GVNS 3

```
procedure GVNS_3(S, k_max, max_time, l_max)
    for i ← 1,5 do
        InitialOrder(i) = i
    end for
    ShakingOrder = Shuffle(InitialOrder)
    while time ≤ max_time do
        ShakingOrder = Adaptive_Order(ShakingOrder, InitialOrder)
        for k ← 1, k_max do
            for i ← 1,5 do
                l = ShakingOrder(i)
                S* = Shake_1(S, l)
                S' = pVND_1(S*, l_max)
                S* = N_7(S')
                if f(S*) < f(S) then
                    S ← S*
                end if
            end for
        end for
    end while
    return S
end procedure
```

The pseudocodes of $GVNS\_4$, $GVNS\_5$ and $GVNS\_6$ are omitted, as they are similar to the previously provided GVNS schemes. Their differences are the use of $pVND\_2$ and operator N_8, which is executed exactly after operator N_7. More specifically, to solve the heterogeneous problem case efficiently, further local search and shaking operators are required. These operators perform proper changes in order to improve the fleet composition.

Due to the fact that the reduced adaptive shaking strategy is a special case of the adaptive shaking strategy, each GVNS scheme, which uses this shaking approach, is defined as $GVNS\_X_R$, where $GVNS\_X$ is the corresponding GVNS scheme with no-reduced adaptive shaking. For instance, the reduced variant of $GVNS\_3$ is the $GVNS\_3_R$ and its pseudocode is provided in Algorithm 22 .

Algorithm 22 GVNS 3 with reduced adaptive shaking

```
procedure GVNS_3_R(S, k_max, max_time, l_max)
    for i ← 1,5 do
        InitialOrder(i) = i
    end for
    ShakingOrder = Shuffle(InitialOrder)
    while time ≤ max_time do
        ShakingOrder = Adaptive_Order(ShakingOrder, InitialOrder)
        i = 1
        for k ← 1, k_max do
            l = ShakingOrder(i)
            S* = Shake_1(S, l)
            S' = pVND_1(S*, l_max)
            S* = N_7(S')
            if f(S*) < f(S) then
                S ← S*
            end if
            i = i + 1
            if i > 5 then
                i = 1
            end if
        end for
    end while
    return S
end procedure
```

The results of computational experiments on the heterogeneous case of the problem, show a potential benefit with increasing fleet diversity (for further details see Subsection 4.5.3). Those GVNS schemes use the *Shake_2* instead of *Shake_1*.

Finally, it should be mentioned that several auxiliary methods have been developed to guarantee the feasibility of the obtained solutions. For instance, a method which examines the existence of sub-routes in a selected route.

4.5 Computational analysis and results

4.5.1 Computing environment

The MIP formulation of the studied problem was implemented in GAMS (GAMS 24.9.1) (Brooke et al., 1998) and its instances were solved by CPLEX 12.7.1.0 solver. The time limit for solving small-sized instances was set at two hours, while for the medium- and large-sized instances the time limit was increased to up to three hours. The proposed algorithms were coded in Fortran and they were executed by Intel Fortran compiler 18.0 using the optimization option /O3. Both CPLEX and Intel Fortran compiler ran on a laptop PC running Windows 10 Home 64-bit with an Intel Core i7-6700 CPU at 2.6 GHz and 16 GB RAM. The execution time limit for the designed heuristic algorithms was set at $60s$.

4.5.2 Problem instances

Due to the fact that the FSMPLIRP is introduced in this work, there is no available test instances in the literature. Thus, 30 new problem instances were randomly generated, using the instructions given by Zhang et al. (2014). The vehicle fixed cost of light-duty vehicles is randomly generated with a Normal distribution with parameters $\mu = 1000$ and $\sigma = 500$, while the cost of medium-duty vehicles is calculated as $floor((light\_cost + (light\_cost * (20\% + rand(0.5, 5)))))$. Each problem instance has a name formed as X-Y-Z, where X denotes the number of potential depots, Y the number of customers and Z the number of time periods. The set of generated problem instances are available in http://pse.cheng.auth.gr/index.php/publications/benchmarks/.

4.5.3 Parameter setting & computational results

Before the presentation of the experimental study, an overview of the proposed solution method is provided in Figure 4.1.

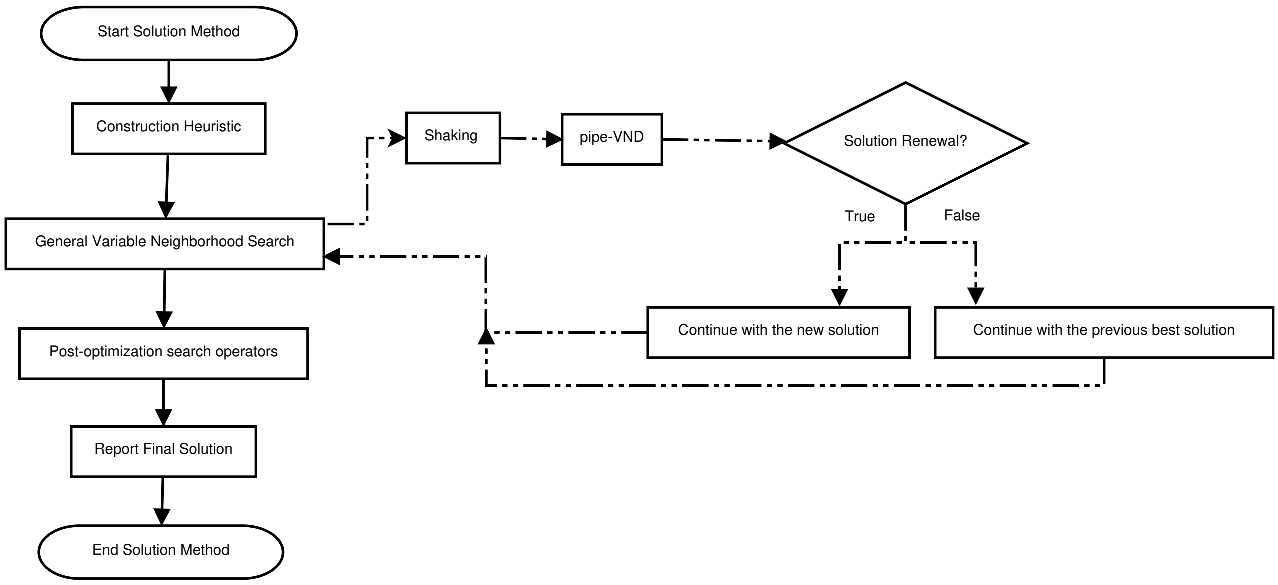

FIGURE 4.1: Flowchart of the proposed solution method.

A critical parameter of a VNS-based heuristic algorithm is k_{max}. In this regard, a parameter estimation is performed in order to select the most efficient value of this parameter. The examined values of k_{max} are 10, 12, 15, 20 and 25. For this estimation process, the $GVNS\_1$ is used (light-duty vehicles case). Table 4.5 summarizes the total cost achieved for each problem instance and different values of k_{max}. It should be mentioned that in all presented results, the reported value of each instance is the average solution of 10 runs.

TABLE 4.5: k_{max} analysis on the $GVNS\_1$ performance on 30 GLIRP

Instance	$k_{max} = 10$	$k_{max} = 12$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = 25$
4-9-3	19,950.64	19,950.64	20,025.74	19,965.42	20,001.45
4-10-3	20,776.81	20,401.16	20,323.12	20,442.53	20,484.32
4-10-5	16,890	16,757.12	16,639.51	16,994.84	16,773.05
4-12-5	20,745.54	21,537.64	19,257.65	19,241.52	19,298.63
4-15-3	10,205.15	10,202.94	10,202.69	10,202.63	10,207.29
5-12-3	12,966.13	12,966	12,966.47	12,966.83	12,982.32
5-15-3	15,980.28	15,973.33	15,979.88	15,970.49	15,979.27
5-15-5	21,963.65	22,156.5	21,973	21,984.31	22,028.07
5-18-3	23,382.51	22,843.35	22,989.54	23,503.2	22,795.83
5-20-3	19,082.94	19,080.76	19,145.74	19,095.92	19,083.83
6-40-5	22,053.01	22,116.36	22,086.75	22,166.08	22,051.46
7-52-5	16,565.59	16,459.8	16,475.07	16,449.83	16,602.87
7-55-7	20,640.6	20,740.16	20,680.7	20,734.96	20,680.7
8-60-5	25,158.09	25,270.78	24,917.71	25,192.96	25,094.77
8-65-7	45,432.84	46,333.11	46,389.98	46,813.38	46,404.39
9-70-5	27,257.63	27,257.63	27,257.63	26,954.93	26,422.93
9-75-7	29,229.05	29,235.23	29,256.86	29,229.98	29,234.61
9-85-5	23,312.07	23,113.37	23,355.22	23,346.96	23,307.15
9-88-7	28,413.28	28,298.62	28,497.24	28,622.26	28,606.53
10-90-7	25,664.05	25,744.83	25,744.83	25,651.54	25,744.6
15-100-7	21,079.91	21,175.92	20,676.81	21,168.42	21,061.69
15-100-10	32,776.21	33,164.21	32,454.69	33,162.97	33,162.97
15-120-10	32,001.47	31,998.65	31,712.23	31,869.61	31,866.53
20-150-10	27,251.23	27,247.53	27,011.78	27,251.23	27,242.03
20-180-12	56,623.45	56,001.89	55,474.01	56,363.05	56,779.96
25-200-12	53,858.84	55,481.55	53,660.57	55,448.09	55,502.41
30-250-10	40,514.82	40,608.62	40,608.62	40,621.34	40,339.15
30-270-10	40,604.43	40,001.64	39,793.64	39,817.2	39,804.99
35-300-10	69,917.79	71,524.05	70,530.91	70,429.36	70,638.92
35-310-12	70,241.78	69,334.98	70,366.98	69,721.21	70,114.25
Average	29,684.66	29,765.95	**29,350.96**	29,022.74	29,676.57

In accordance with the average values of the previously reported results, it is obvious that $k_{max} = 15$ produces slightly better solutions than the other tested values. This minor improvement is mainly based on the results achieved on ten small-sized and ten large-sized instances. The selected strength of shaking presumably permits more iterations of the improvement phase than the more intense shaking options and better exploration than the limited k_{max} choices.

To fairly compare the intensified shaking with the two proposed adaptive shaking methods (actually their corresponding GVNS schemes), the same k_{max} value is also used in the adaptive cases. Table 4.6 provides the average and best results obtained by $GVNS\_1$, $GVNS\_2$ and $GVNS\_3$.

TABLE 4.6: Average and best values of $GVNS\_1$, $GVNS\_2$ and $GVNS\_3$

Instance	$GVNS_1\_Avg$	$GVNS_1\_Best$	$GVNS_2\_Avg$	$GVNS_2\_Best$	$GVNS_3\_Avg$	$GVNS_3\_Best$
4-9-3	20,025.74	19,893.35	19,965.42	19,893.35	19,965.42	19,893.35
4-10-3	20,323.12	20,211.22	20,388.18	20,386.88	20,477.19	20,236.63
4-10-5	16,639.51	16,639.47	16,654.57	16,639.49	16,668.95	16,639.52
4-12-5	19,257.65	19,218.74	19,330	19,218.7	19,999.71	19,232.61
4-15-3	10,202.69	10,197.83	10,204.27	10,199.25	10,206.14	10,197.82
5-12-3	12,966.47	12,965.52	12,977.22	12,965.53	12,966.05	12,965.53
5-15-3	15,979.88	15,968.03	15,978.63	15,967.97	15,982.42	15,968.01
5-15-5	22,040.32	21,811.93	21,973.37	21,829.92	22,097.31	22,061.04
5-18-3	22,989.54	22,034.8	22,393.16	22,044.37	22,769.88	22,048.34
5-20-3	19,145.74	19,072.08	19,097.41	18,970.59	19,109.76	18,969.02
6-40-5	22,086.75	21,869.92	22,033.75	21,955.77	22,113.73	21,864.69
7-52-5	16,475.07	16,346.4	16,492.06	16,338.96	16,523.26	16,357.35
7-55-7	20,680.7	20,483.23	20,289.07	20,133.6	20,617.13	20,220.49
8-60-5	25,209.82	24,851.46	24,917.71	24,366.79	25,008.58	24,745.05
8-65-7	46,389.98	45,296.65	46,216.56	45,553.83	46,749.3	46,256.98
9-70-5	27,257.63	25,545.31	25,532.28	25,277.36	25,450.91	25,224.12
9-75-7	29,256.86	29,142.09	29,272.45	29,137.69	29,205.54	29,107.46
9-85-5	23,355.22	23,022.43	22,858.58	22,608.26	23,240.07	22,985.04
9-88-7	28,497.24	28,392.9	28,615.11	28,451.68	28,676.73	28,392.88
10-90-7	25,744.83	25,484.67	25,438.81	25,245.69	25,437.12	25,021.07
15-100-7	20,676.81	18,625.63	20,581.55	20,285.07	20,507.03	20,234.9
15-100-10	32,742.03	31,188.25	32,454.69	31,394.29	32,586.89	31,942.82
15-120-10	31,712.23	30,893.46	32,617.02	32,180.6	32,171.38	31,680.79
20-150-10	27,011.78	26,103.88	26,916.86	26,619.65	26,723.04	26,606.19
20-180-12	55,894.64	55,090.93	55,474.01	55,074.75	56,836.62	56,310.66
25-200-12	53,660.57	52,278.51	52,938.72	52,275.73	52,322.95	51,564.92
30-250-10	40,608.62	39,350.63	40,342.18	39,633.71	40,846.64	39,432.27
30-270-10	39,793.64	37,218.66	38,271.97	37,788.59	37,477.92	36,481.77
35-300-10	70,530.91	67,347.98	69,935.89	69,155.16	69,935.89	69,155.16
35-310-12	70,366.98	67,722.48	69,916.42	69,088.91	69,916.42	69,088.91
Average	29,584.1	28,808.95	**29,335.93**	**29,022.74**	29,419.67	29,029.51

The above results illustrate that both GVNS schemes using adaptive shaking perform better than the GVNS scheme using the classic intensified shaking. More specifically, both of the adaptive shaking methods are more effective than the classic one. This effectiveness may depend on the reduced randomness in the selection of shaking operators.The adaptive shaking with a complexity-based initial order is a pure deterministic method, while the second one confines randomness in the initial order of its operators. In the classical shaking method, each shaking operator has the same probability to be selected. It has been observed that in some problem instances one or more shaking operators cannot lead to efficient search, they keep being selected iteratively, though. Moreover, the $GVNS\_2$ produces better quality solutions than the $GVNS\_3$. Further, the $GVNS\_2$ is compared with its corresponding reduced scheme, $GVNS\_2_R$. Their numerical results are reported in Table 4.7.

TABLE 4.7: $GVNS\_2$ vs $GVNS\_2_R$

Instance	*GVNS_2_Avg*	*GVNS_2_Best*	$GVNS\_2_R\_Avg$	$GVNS\_2_R\_Best$
4-9-3	19,965.42	19,893.35	20,001.45	19,893.35
4-10-3	20,388.18	20,386.88	20,371.25	20,306.64
4-10-5	16,654.57	16,639.49	16,670.78	16,639.5
4-12-5	19,330	19,218.7	19,408.15	19,363.85
4-15-3	10,204.27	10,199.25	10,202.1	10,197.66
5-12-3	12,977.22	12,965.53	12,965.53	12,965.52
5-15-3	15,978.63	15,967.97	15,975.89	15,966.53
5-15-5	21,973.37	21,829.92	21,973.37	21,829.92
5-18-3	22,393.16	22,044.37	22,912.03	22,393.16
5-20-3	19,097.41	18,970.59	19,060.16	19,013.63
6-40-5	22,033.75	21,955.77	22,054.8	21,930.66
7-52-5	16,492.06	16,338.96	16,431.74	16,213.8
7-55-7	20,289.07	20,133.6	20,263.26	20,188.2
8-60-5	24,917.71	24,366.79	24,917.71	24,366.79
8-65-7	46,216.56	45,553.83	46,550.04	45,434.11
9-70-5	25,532.28	25,277.36	25,260.02	25,095.64
9-75-7	29,272.45	29,137.69	29,229.07	29,051.29
9-85-5	22,858.58	22,608.26	22,977.73	22,777.82
9-88-7	28,615.11	28,451.68	28,594.76	28,410.18
10-90-7	25,438.81	25,245.69	25,599.85	25,336.84
15-100-7	20,581.55	20,285.07	20,670.1	20,333.84
15-100-10	32,454.69	31,394.29	32,454.69	31,394.29
15-120-10	32,617.02	32,180.6	32,303.52	31,684.38
20-150-10	26,916.86	26,619.65	26,928.55	26,681.24
20-180-12	55,474.01	55,074.75	55,474.01	55,074.75
25-200-12	52,938.72	52,275.73	52,965.83	52,275.73
30-250-10	40,342.18	39,633.71	40,497.02	39,821.39
30-270-10	38,271.97	37,788.59	38,190.2	37,634.62
35-300-10	69,935.89	69,155.16	70,789.39	69,969.02
35-310-12	69,916.42	69,088.91	71,584.55	68,786.73
Average	**29,335.93**	**29,022.74**	29,442.59	29,034.37

The results indicate that the $GVNS\_2$ is a more suitable scheme for solving the homogeneous case of the problem than its reduced version. Thus, the $GVNS\_2$ is compared with the results obtained by the CPLEX solver, in order to further evaluate its efficiency. This comparison is summarized in Table 4.8. "OM" indicates the out-of-memory error occurred by solving large-sized instances.

TABLE 4.8: Compare the results achieved by $GVNS\_2$ and CPLEX (using light-duty vehicles)

Instance	CPLEX (a)	$GVNS\_2\_Avg$ **(b)**	$GVNS\_2\_Best$ **(c)**	**Gap a-b %**	**Gap a-c %**
4-9-3	19,261.33	19,965.42	19,893.35	- 3.66	- 3.28
4-10-3	20,022.66	20,388.18	20,306.64	- 1.83	- 1.82
4-10-5	16,690.5	16,654.57	16,639.5	0.22	0.31
4-12-5	19,551.98	19,330	19,218.7	1.14	1.7
4-15-3	10,412.98	10,204.27	10,199.25	2	2.05
5-12-3	13,146.48	12,977.22	12,965.53	1.29	1.38
5-15-3	15,715.24	15,978.63	15,965.53	- 1.68	- 1.61
5-15-5	23,045.4	21,973.37	21,829.92	4.65	5.27
5-18-3	22,572.41	22,393.16	22,044.37	0.79	2.34
5-20-3	23,873.07	19,097.41	18,970.59	20	20.54
6-40-5	N/A	22,033.75	21,955.77	-	-
7-52-5	N/A	16,492.06	16,338.96	-	-
7-55-7	N/A	20,289.07	20,133.6	-	-
8-60-5	N/A	24,917.71	24,366.79	-	-
8-65-7	N/A	46,216.56	45,553.83	-	-
9-70-5	N/A	25,532.28	25,277.36	-	-
9-75-7	N/A	29,272.45	29,137.69	-	-
9-85-5	N/A	22,858.58	22,608.26	-	-
9-88-7	N/A	28,615.11	28,594.76	-	-
10-90-7	OM	25,438.81	25,245.69	-	-
15-100-7	OM	20,581.55	20,285.07	-	-
15-100-10	OM	32,454.69	31,394.29	-	-
15-120-10	OM	32,617.02	32,180.6	-	-
20-150-10	OM	26,916.86	26,619.65	-	-
20-180-12	OM	55,474.01	55,074.75	-	-
25-200-12	OM	52,938.72	52,275.73	-	-
30-250-10	OM	40,342.18	39,633.71	-	-
30-270-10	OM	38,271.97	37,788.59	-	-
35-300-10	OM	69,935.89	69,155.16	-	-
35-310-12	OM	69,916.42	69,088.91	-	-

The $GVNS\_2$ produces almost 3% better solutions than CPLEX in average for the case of the small-sized instances. Focused on the best found solutions of the $GVNS\_2$, this gap is increased approximately to 3.4%. As it can be noticed, the CPLEX solver cannot provide feasible solutions for the medium-sized instances under the specified time limit. Moreover, an out-of-memory error occurred during the solution of the medium-sized instance “10-90-7” and all large-sized instances. As the $GVNS\_2$ proved to be efficient in solving problem instances of the studied problem, it is also used to solve these instances under the usage of medium-duty trucks. The achieved results are compared with those produced by CPLEX solver and they are reported in Table 4.9.

TABLE 4.9: Compare the results achieved by $GVNS\_2$ and CPLEX (using medium-duty vehicles)

Instance	CPLEX (a)	$GVNS\_2\_Avg$ **(b)**	$GVNS\_2\_Best$ **(c)**	**Gap a-b %**	**Gap a-c %**
4-9-3	19,161.21	19,907.18	19,867.3	- 3.89	- 3.68
4-10-3	19,871.37	20,025.72	19,984.59	- 0.78	- 0.57
4-10-5	16,641.27	16,481.28	16,478.75	0.96	0.98
4-12-5	20,477.68	19,135.32	19,125.16	6.56	6.6
4-15-3	10,296.25	10,178.69	10,176.11	1.14	1.17
5-12-3	13,038.15	12,858.72	12,854.84	1.38	1.41
5-15-3	15,633.85	15,801.46	15,787.22	- 1.07	- 0.98
5-15-5	23,523.54	21,838.86	20,449.79	7.16	13.07
5-18-3	22,345.25	21,816.29	20,858.26	2.37	6.65
5-20-3	20,379.37	18,857	18,784.91	7.47	7.82
6-40-5	N/A	21,401.83	21,069.95	-	-
7-52-5	N/A	16,370.36	16,202.46	-	-
7-55-7	N/A	20,684.1	20,289.71	-	-
8-60-5	N/A	23,977.64	23,607.5	-	-
8-65-7	N/A	42,803.82	41,951.12	-	-
9-70-5	N/A	25,551.22	24,409.88	-	-
9-75-7	N/A	28,340.91	28,310.15	-	-
9-85-5	N/A	23,452.48	23,050.71	-	-
9-88-7	N/A	29,048.78	28,817.38	-	-
10-90-7	OM	25,876.82	25,483.62	-	-
15-100-7	OM	15,211.71	13,986.74	-	-
15-100-10	OM	32,413.15	32,058.96	-	-
15-120-10	OM	33,093.59	32,846.37	-	-
20-150-10	OM	27,227.04	26,836.73	-	-
20-180-12	OM	58,273.09	57,768.52	-	-
25-200-12	OM	53,710.14	52,649.32	-	-
30-250-10	OM	46,402.38	45,561.52	-	-
30-270-10	OM	39,775.43	39,323.89	-	-
35-300-10	OM	65,889.8	64,470.15	-	-
35-310-12	OM	73,180.99	72,344.19	-	-

The proposed GVNS algorithm performs approximately 2.5% better than CPLEX in solving the homogeneous case of the problem using medium-duty vehicles (approximately up to 4% focused on best found solutions of GVNS). Moreover, the commercial solver cannot provide feasible solutions even for some small-sized instances.

The impact of using different type of vehicles on fuel consumption, its cost and CO_2 emissions is illustrated in Figures 4.2, 4.3 and 4.4 respectively.

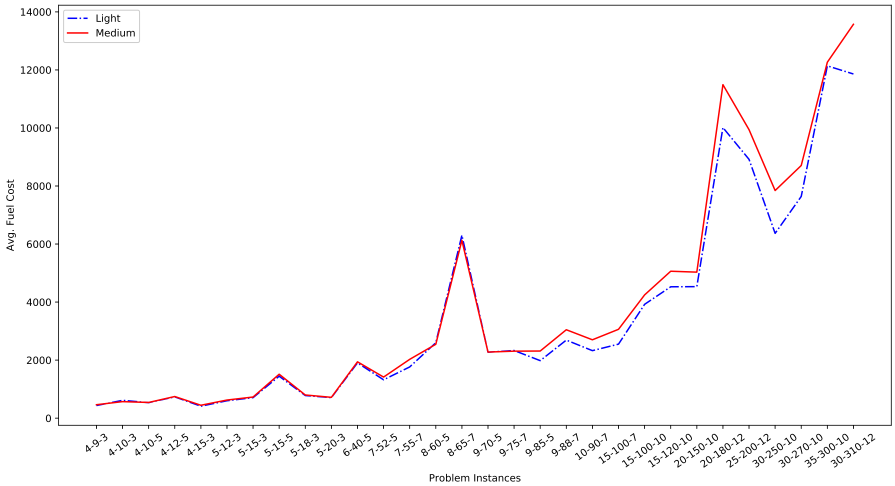

FIGURE 4.2: The average fuel consumption cost in cases of light- and medium-duty vehicles.

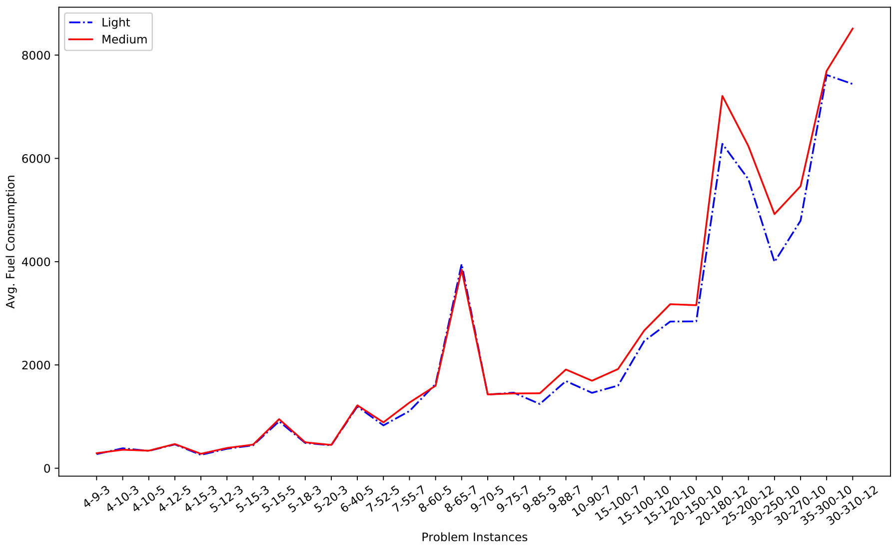

FIGURE 4.3: The average fuel consumption (L) in cases of light- and medium-duty vehicles.

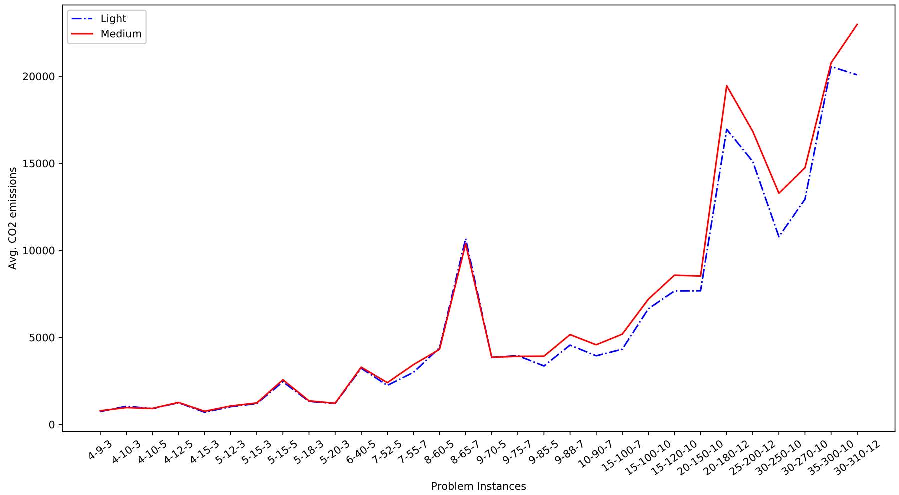

FIGURE 4.4: The average CO_2 emissions (kg) in cases of light- and medium-duty vehicles.

It is important to highlight that the use of medium-duty vehicles leads to significant increase on fuel consumption and CO_2 emissions. More specifically, the fuel consumption, the CO_2 levels and their corresponding costs are increased by approximately 10% (9.66%). Nonetheless, the solutions obtained by using a fleet of medium-duty vehicles are slightly better than those achieved using light-duty vehicles.

However, a mixed-fleet is commonly adopted in real-life applications. Thus, further examination is made in this direction. $GVNS\_4$, $GVNS\_5$ and $GVNS\_6$ are initially tested on the 30 random generated instances. Their results are provided in Table 4.10.

TABLE 4.10: Average and best values of $GVNS\_4$, $GVNS\_5$ and $GVNS\_6$

Instance	$GVNS_4\_Avg$	$GVNS_4\_Best$	$GVNS_5\_Avg$	$GVNS_5\_Best$	$GVNS_6\_Avg$	$GVNS_6\_Best$
4-9-3	30,520.18	29,647.29	30,765.42	30,693.35	30,829.18	30,799.61
4-10-3	35,995.16	34,513.94	34,548.23	34,513.94	36,137.61	34,513.95
4-10-5	28,832	28,832	28,736.49	28,714.56	28,719.87	28,714.62
4-12-5	37,984.05	37,211.55	37,084.53	36,260.56	39,506.74	37,204
4-15-3	17,403.02	17,397.8	17,404.49	17,397.68	17,406.42	17,397.69
5-12-3	27,441.68	27,423.38	27,440.32	27,422.85	27,427.79	27,422.85
5-15-3	30,375.2	30,368.01	30,373.36	30,366.56	30,374.89	30,368.03
5-15-5	69,067.65	67,497.91	68,913.66	66,977.66	69,133.17	68,662.37
5-18-3	46,657.86	46,390.79	46,656.1	46,390.98	46,917.01	46,406.27
5-20-3	33,527.63	33,475.16	33,526.97	33,470.58	33,485.1	33,402.98
6-40-5	61,421.62	59,524.58	62,042.52	61,639.27	61,680.45	61,591.17
7-52-5	40,447.48	40,391.73	40,481.73	40,448.62	40,492.68	40,428.09
7-55-7	45,860.46	45,718.6	45,621.25	45,421.78	45,926.44	45,722.02
8-60-5	94,385.55	93,461.52	93,143.03	92,208.35	93,827.05	92,065.25
8-65-7	253,406.5	251,405.2	253,101.7	251,508.6	252,217.1	249,853.3
9-70-5	73,244.6	68,938.83	72,703.7	68,663.82	72,474.47	69,907.24
9-75-7	58,312.67	58,174.97	58,347.14	58,260.16	58,187.53	57,971.01
9-85-5	47,269.27	46,890.23	47,061.94	46,992.61	47,293.92	47,010.44
9-88-7	62,355.02	62,181.18	62,376.73	62,196.86	62,315.84	62,242.41
10-90-7	45,197.7	43,846.87	43,894.43	43,761.46	43,860.43	43,517.92
15-100-7	50,805.08	48,844.84	50,482.09	48,101.82	48,846.77	48,371.12
15-100-10	71,896.95	69,419.84	71,738.12	70,931.28	71,147.65	70,017.36
15-120-10	78,304.45	76,205.23	76,163.42	74,377.12	78,649.79	77,165.66
20-150-10	87,011.1	86,023.12	86,066.52	83,902.64	86,030.11	83,007.77
20-180-12	187,639.4	186,855.5	187,120.5	186,712.4	186,053.3	184,981.3
25-200-12	165,293	163,618.9	164,964.5	161,578.4	164,715.2	162,377.6
30-250-10	76,810.38	75,669.81	76,497.02	75,821.38	77,208.77	76,601.8
30-270-10	88,248.25	85,529.95	86,804.2	86,043.47	86,473.67	85,925.92
35-300-10	196,312	194,688.7	196,371.9	194,698.6	194,955.3	194,174.2
35-310-12	133,082	131,259.6	133,164.1	132,237.7	133,164.1	132,237.7
Average	75836.93	74713.57	**75453.2**	**74590.5**	75515.28	74668.72

Similar to the homogeneous case of the problem, the GVNS scheme which uses the adaptive shaking mechanism with a complexity-based initial order is proved the most efficient method. Furthermore, it is interesting to examine the reduced case of $GVNS\_5$. Table 4.11 contains the average and the best found solutions of $GVNS\_5_R$ and their gap (%) from the corresponding $GVNS\_5$ solutions.

TABLE 4.11: The results achieved by $GVNS\_5_R$ and their gap from the results of $GVNS\_5$

Instance	$GVNS\_5_R\_Avg$	$GVNS\_5_R\_Best$	Gap Avg. Solutions %	Gap Best Solutions %
4-9-3	30,786.67	30,693.35	-0.07	0
4-10-3	34,487.43	34,381.39	0.18	0.38
4-10-5	28,776.89	28,731.59	-0.14	-0.06
4-12-5	37,093.05	36,277.87	-0.02	-0.05
4-15-3	17,405.14	17,399.35	0	-0.01
5-12-3	27,432.32	27,422.85	0.03	0
5-15-3	30,379.59	30,368.05	-0.02	0
5-15-5	69,218.02	67,989.64	-0.44	-1.51
5-18-3	46,775.61	46,390.61	-0.26	0
5-20-3	33,520.76	33,481.52	0.02	-0.03
6-40-5	61,423.02	60,347.19	1	2.1
7-52-5	40,679.31	40,458.96	-0.49	-0.03
7-55-7	45,511.59	45,452.61	0.24	-0.07
8-60-5	93,115.1	92,390.42	0.03	-0.2
8-65-7	251,383.6	249,532.1	0.68	0.79
9-70-5	72,597.73	69,526.91	0.15	-1.26
9-75-7	58,223.48	58,030.34	0.21	0.39
9-85-5	47,107.16	46,879.55	-0.1	0.24
9-88-7	62,397.38	62,219.98	-0.03	-0.04
10-90-7	43,938.91	43,811.77	-0.1	-0.11
15-100-7	45,505.46	41,004.02	9.86	14.76
15-100-10	72,098.84	71,548.5	-0.5	-0.87
15-120-10	76,792.02	76,115.48	-0.83	-2.34
20-150-10	86,059.3	84,094.49	0.01	-0.23
20-180-12	186,510	185,948.6	0.33	0.41
25-200-12	163,087.7	159,319.8	1.14	1.4
30-250-10	76,439.84	75,821.38	0.07	0
30-270-10	86,554.11	85,849.48	0.29	0.23
35-300-10	195,684.7	194,490.1	0.35	0.11
35-310-12	133,164.1	132,237.7	0	0

Despite the results clearly indicate that both $GVNS\_5$ and $GVNS\_5_R$ perform almost equivalently, it seems that the GVNS with the reduced adaptive shaking scheme can produce slightly better solutions than the initial scheme, especially on large problem instances. This may be occurred by the significant reduction of shaking iterations which enables the improvement phase to be executed more times. Furthermore, during the experiments with $GVNS\_5$ and $GVNS\_5_R$, it is noticed that the solutions with an increase in vehicle mixing, are found to be the best. In this direction, an alternative of the $GVNS\_5$ and $GVNS\_5_R$ schemes ($GVNS\_5^*$ and $GVNS\_5_R^*$ respectively), which use the $Shake_2$ instead of $Shake_1$, are tested. Due to the fact that a local search operator is more complex than a shaking operator, the shaking operator S_5 is selected to be used as the expedient on increasing the fleet diversity. However, in order to control this diversity, operator N_8 is also used. The numerical results of $GVNS\_5^*$ and $GVNS\_5^*_R$ are given in Table 4.12.

TABLE 4.12: The average and best found results of $GVNS\_5^*$ and $GVNS\_5^*_R$

Instance	$GVNS\_5^*\_Avg$ **(b)**	$GVNS\_5^*\_Best$ **(c)**	$GVNS\_5^*_R\_Avg$	$GVNS\_5^*_R\_Best$
4-9-3	30,556.21	29,647.29	30,828.79	30,723.82
4-10-3	34,302.87	33,435.3	34,759.74	34,383.76
4-10-5	29,062.39	28,716.11	29,091.07	28,713.12
4-12-5	33,141.53	31,853.81	33,695.88	31,857.18
4-15-3	17,402.1	17,397.66	17,405.18	17,402.33
5-12-3	27,633.45	27,423.38	27,390.11	27,136.26
5-15-3	29,351.3	28,505.29	29,388.97	28,782.4
5-15-5	67,118.98	65,760.69	68,626.54	65,987.51
5-18-3	44,665.75	43,821.99	45,368.95	44,717.53
5-20-3	33,515.91	33,409.03	33,578.46	33,375.17
6-40-5	62,363.02	60,861.09	61,515.82	59,792.91
7-52-5	40,515.12	40,408.63	40,571.34	40,364.73
7-55-7	45,617.8	45,369.76	45,614.4	45,435.75
8-60-5	93,676.57	92,339.34	93,438.79	92,419.65
8-65-7	252,522.6	251,273.5	253,098.5	251,052.1
9-70-5	74,397.69	71,337.26	72,309.77	68,820.45
9-75-7	58,347.14	58,260.16	58,187.02	57,950.38
9-85-5	47,023.98	46,917.3	47,088.62	46,879.55
9-88-7	62,373.95	62,196.86	62,389.43	62,233.86
10-90-7	44,038.39	43,900.62	43,981.29	43,738.73
15-100-7	49,374.27	48,124.49	49,237.94	48,022.45
15-100-10	71,331.44	70,797.74	71,560.47	71,225.41
15-120-10	76,893.95	74,216.33	77,064.65	75,899.13
20-150-10	85,949.9	83,769.98	86,057.27	83,902.64
20-180-12	187,129.2	186,536.6	187,238.5	186,772.4
25-200-12	162,581	159,529	165,767.3	164,258.2
30-250-10	76,497.02	75,821.38	76,439.84	75,821.38
30-270-10	86,813.8	85,990.55	86,723.59	86,022.63
35-300-10	195,334.7	194,490.1	196,106.5	194,698.6
35-310-12	132,730.1	131,412.7	132,730.1	131,412.7
Average	75,075.4	74,117.46	75,241.83	74,326.76

From the reported results in Table 4.12, it is observed that the strategy of increasing fleet diversity leads to further improvements, as the fleet mixing can potentially lead to better formation of routes and lower vehicles usage costs. Also, following this approach, the $GVNS\_5^*$ performs slightly better than its reduced variant both in terms of average and best found solutions.

To further evaluate, the performance of $GVNS\_5^*$, a comparison with CPLEX is attempted and results are provided in Table 4.13.

TABLE 4.13: Compare the results achieved by $GVNS\_5^*$ and CPLEX

Instance	CPLEX (a)	$GVNS\_5^*\_Avg$ **(b)**	$GVNS\_5^*\_Best$ **(c)**	**Gap a-b %**	**Gap a-c %**
4-9-3	30,303	30,556.21	29,647.29	- 0.84	2.16
4-10-3	33,457.4	34,302.84	33,435.3	- 2.53	0.07
4-10-5	32,475.16	29,062.39	28,716.11	10.51	11.58
4-12-5	38,363.81	33,141.53	31,853.81	13.61	16.97
4-15-3	18,356.11	17,402.1	17,397.66	5.2	5.22
5-12-3	28,593.7	27,633.45	27,423.38	3.36	4.09
5-15-3	N/A	29,351.3	28,505.29	-	-
5-15-5	N/A	67,118.98	65,760.69	-	-
5-18-3	N/A	44,665.75	43,821.99	-	-
5-20-3	N/A	33,515.91	33,409.03	-	-
6-40-5	N/A	62,363.02	60,861.09	-	-
7-52-5	N/A	40,515.12	40,408.63	-	-
7-55-7	N/A	45,617.8	45,369.76	-	-
8-60-5	N/A	93,676.57	92,339.34	-	-
8-65-7	N/A	252,522.6	251,273.5	-	-
9-70-5	N/A	74,397.69	71,337.26	-	-
9-75-7	N/A	58,347.14	58,260.16	-	-
9-85-5	N/A	47,023.98	46,917.3	-	-
9-88-7	N/A	62,373.95	62,196.86	-	-
10-90-7	OM	44,038.39	43,900.62	-	-
15-100-7	OM	49,374.27	48,124.49	-	-
15-100-10	OM	71,331.44	70,797.74	-	-
15-120-10	OM	76,893.95	74,216.33	-	-
20-150-10	OM	85,949.9	83,769.98	-	-
20-180-12	OM	187,129.2	186,536.6	-	-
25-200-12	OM	162,581	159,529	-	-
30-250-10	OM	76,497.02	75,821.38	-	-
30-270-10	OM	86,813.8	85,990.55	-	-
35-300-10	OM	195,334.7	194,490.1	-	-
35-310-12	OM	132,730.1	131,412.7	-	-

As shown in Table 4.13, CPLEX can produce feasible solutions only for the six out of ten small-sized instances. The $GVNS\_5^*$ performs approximately 5.21% better than CPLEX, and their difference is increased up to around 7.2% in the case of best found solutions by $GVNS\_5^*$. Considering the high complexity of the studied problem, the achieved quality difference and the significant difference on execution time limits, it can be highlighted that the proposed GVNS scheme is quite efficient for solving the FSMPLIRP. Despite CPLEX is a state-of-the-art optimization solver, setting a strict time limit for the solution of NP-hard problems leads to the production of solutions with high optimality gap (gap between the best integer and the relaxed LP solution). Thus, the solutions obtained by our proposed solution approach are better even for small-sized instances.

4.5.4 Medium-duty vehicles vs mixed-fleet

It has been shown that the use of medium-duty only vehicles performs much better than using only light-duty vehicles mainly in terms of fuel consumption and CO_2 emissions. Therefore, it is interesting to examine how a homogeneous fleet of medium-duty vehicles and a mixed-fleet affect the fuel consumption (L), the corresponding cost and the CO_2 emissions (kg). Figures 4.5, 4.6 and 4.7 illustrate the discussed impacts for various problem instances.

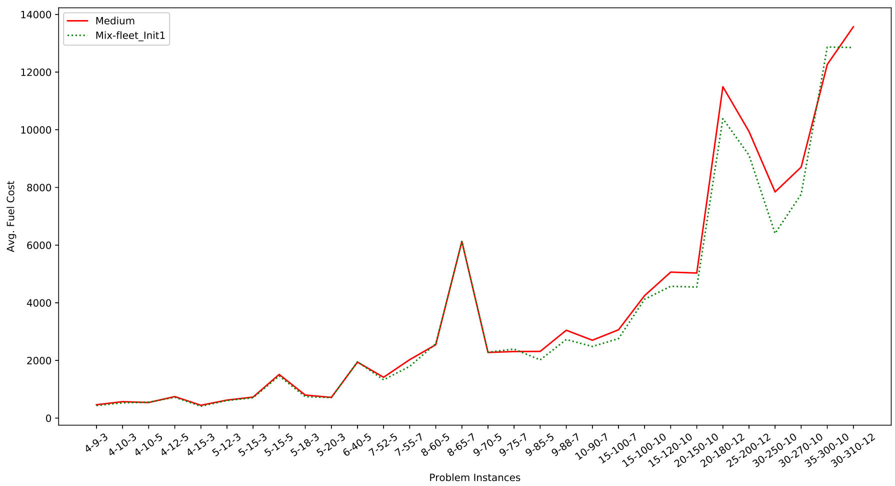

FIGURE 4.5: The average fuel consumption cost in cases of medium-duty vehicles and mixed-fleet.

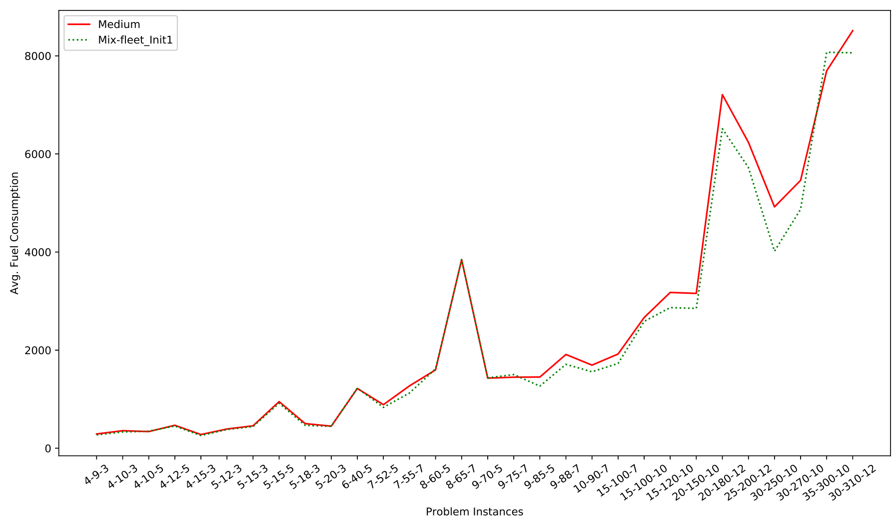

FIGURE 4.6: The average fuel consumption (L) in cases of medium-duty vehicles and mixed-fleet.

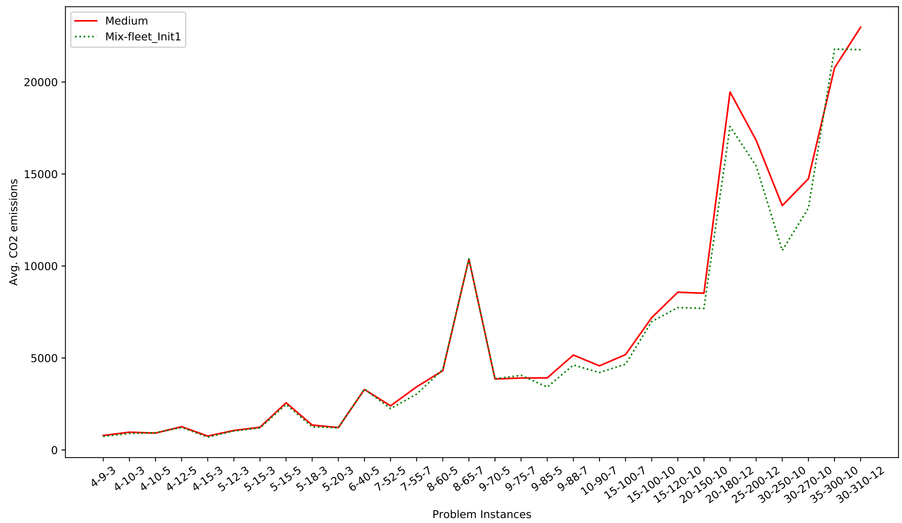

FIGURE 4.7: The average CO_2 emissions (kg) in cases of medium-duty vehicles and mixed-fleet.

The selection of a mixed-fleet significantly decreases the fuel consumption, the CO_2 emissions and their corresponding cost, especially for the case of large problem cases. However, an other critical decision parameter is the vehicle usage cost. Figure 4.8 illustrates the different vehicle usage cost levels for each fleet case.

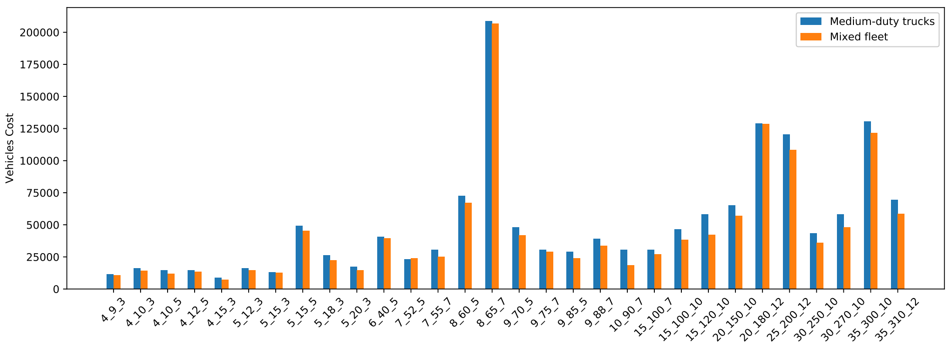

FIGURE 4.8: The vehicles usage costs in cases of medium-duty vehicles and mixed-fleet.

It is clear that the mixed-fleet is more cost effective than the case of using a homogeneous fleet of medium-duty vehicles (approximately 10%). Therefore, the use of a mixed-fleet is a sustainable strategic decision.

The impact of initialization. The use of different initialization rules has a potential effect on the solution of a GVNS heuristic (Hansen & Mladenović, 2014). Therefore, it is examined whether an alternative customers' allocation rule has a considerable effect on the final solution of the $GVNS\_5^*$ (which has been proved the best scheme for solving the FSMPLIRP) or not. More specifically, the alternative allocation is also a nearest allocation method, which is applied by considering all the opened depots. The new GVNS scheme is mentioned as $GVNS\_5^*_{Init2}$. Table 4.14 provides the results obtained by $GVNS\_5^*_{Init2}$ and the comparison of them with the solutions produced by CPLEX.

TABLE 4.14: Compare the results achieved by $GVNS\_5^*$ using different initialization methods

Instance	$GVNS\_5^*_{Init2}\_Avg$	$GVNS\_5^*_{Init2}\_Best$	Gap a %	Gap b %
4-9-3	29,992.57	29,257.67	1.84	1.31
4-10-3	34,727.45	34,366.66	-1.24	-2.79
4-10-5	28,928.39	29,822.83	0.46	-0.72
4-12-5	33,404.32	32,841.5	-0.79	-3.1
4-15-3	17,406.08	17,400.79	-0.02	-0.02
5-12-3	27,428.02	27,423.36	0.74	0
5-15-3	29,628.76	28,672.01	-0.95	-0.58
5-15-5	70,556.1	69,437.41	-5.12	-5.59
5-18-3	47,051.2	46,479.4	-5.34	-6.06
5-20-3	33,640.74	33,528.41	-0.37	-0.36
6-40-5	63,186.84	63,129.26	-1.32	-3.73
7-52-5	39,992.37	39,828.16	1.29	1.44
7-55-7	45,121.02	45,020.08	1.09	0.77
8-60-5	94,286.66	92,886.35	-0.65	-0.59
8-65-7	249,722.7	248,333.5	1.11	1.17
9-70-5	72,037.95	71,681.4	3.17	-0.48
9-75-7	51,536.19	51,314.25	11.67	11.92
9-85-5	46,006.43	45,953.77	2.16	2.05
9-88-7	64,279.82	63,211.97	-3.06	-1.63
10-90-7	44,038.39	43,900.62	0	0
15-100-7	36,894.5	32,459.81	25.28	32.55
15-100-10	68,180.36	67,422.41	4.42	4.77
15-120-10	82,139.7	81,899.34	-6.82	-10.35
20-150-10	85,949.9	83,769.98	0	0
20-180-12	187,034.1	186,467.2	-0.05	0.04
25-200-12	148,209	147,419.9	8.84	7.59
30-250-10	78,958.62	78,463.9	-3.22	-3.49
30-270-10	89,426.97	88,609.33	-3.01	-3.05
35-300-10	182,873.9	181,887.7	6.38	6.48
35-310-12	115,865	115,465	12.71	12.14

The obtained results accentuate the impact of using different initialization rules. Focused on the large-sized instances the quality gap between $GVNS\_5^*$ and $GVNS\_5^*_{Init2}$ is approximately 5%. It has been observed that, by applying this alternative initialization rule, better routes can be built. The potential efficient geographic segmentation of customers can be a reasonable justification of the reported improvements. It is also interesting to focus on the potential effect of different initialization methods on fuel- and emissions-based details. Figures 4.9, 4.11 and 4.12 illustrate the observed differences.

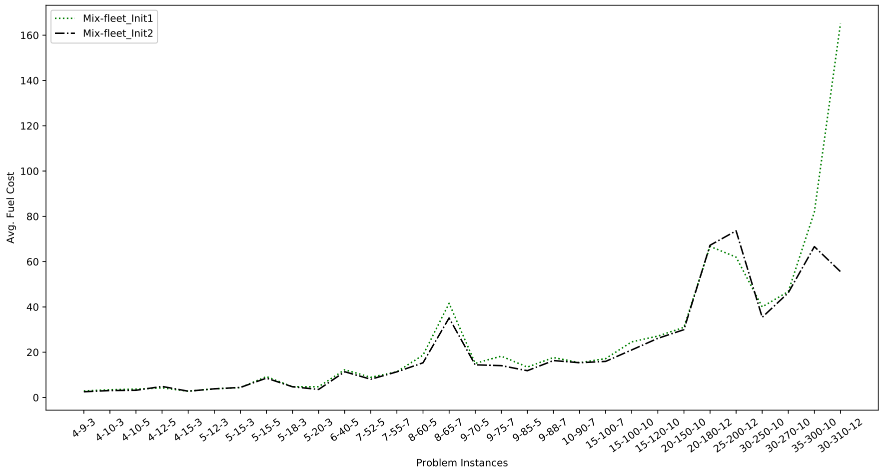

FIGURE 4.9: The average fuel consumption cost using different initialization rules.

Figure 4.9 cannot provide a clear view on fuel cost changes for the case of the ten small-sized instances. Thus, a more focused view on these instances is given in Figure 4.10.

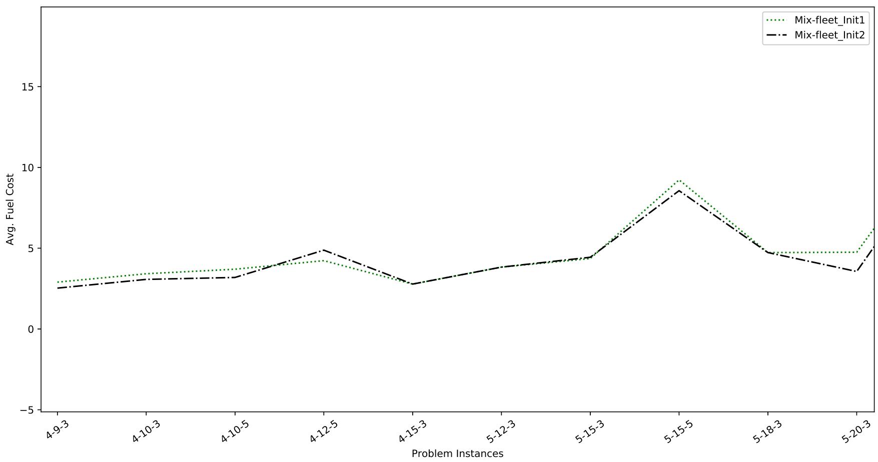

FIGURE 4.10: The average fuel consumption cost of ten small-sized instances using different initialization rules.

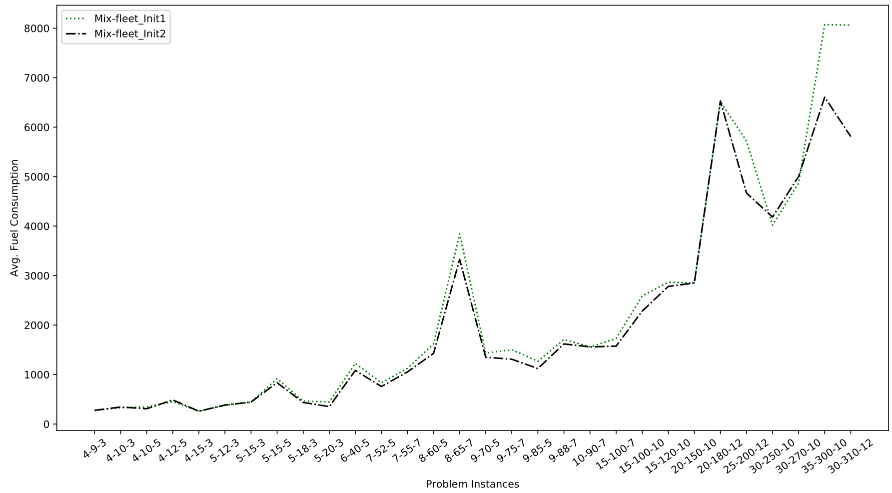

FIGURE 4.11: The average fuel consumption (L) using different initialization rules.

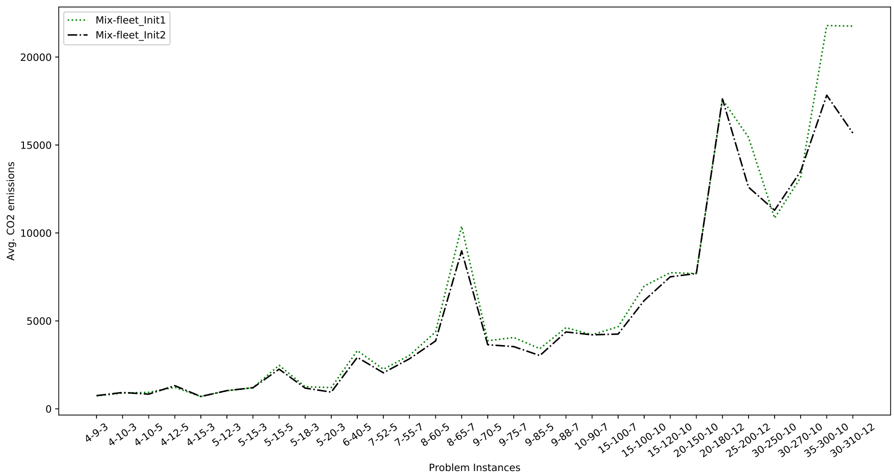

FIGURE 4.12: The average CO_2 emissions (kg) using different initialization rules.

It is noted that the $GVNS\_5^*_{Init2}$ mainly leads to more environmentally efficient solutions.

4.5.5 Opened depots and fleet composition

This section summarizes the number of opened depots, the capacity levels and the number (and type) of vehicles as they reported in the best found solutions for each problem instance. Thus, Table 4.15 provide the number of opened depots for each case of the studied problem and each solution method (CPLEX & GVNS).

TABLE 4.15: Number of opened depots per instance.

Instance	CPLEX_Light	$GVNS\_2$ (Light)	CPLEX_Medium	$GVNS\_2$ (Medium)	CPLEX_MixedFleet	$GVNS\_5^{*}_{Init2}$
4-9-3	2	2	2	2	2	2
4-10-3	2	2	2	2	2	2
4-10-5	2	2	2	2	2	2
4-12-5	3	2	2	2	2	2
4-15-3	1	1	1	1	2	1
5-12-3	1	1	1	1	1	1
5-15-3	1	1	1	1	-	1
5-15-5	2	2	2	2	-	2
5-18-3	2	2	2	2	-	2
5-20-3	3	2	-	2	-	2
6-40-5	-	2	-	2	-	2
7-52-5	-	2	-	2	-	2
7-55-7	-	2	-	2	-	2
8-60-5	-	2	-	2	-	2
8-65-7	-	2	-	2	-	2
9-70-5	-	2	-	2	-	2
9-75-7	-	2	-	2	-	2
9-85-5	-	2	-	2	-	2
9-88-7	-	2	-	2	-	2
10-90-7	-	2	-	2	-	2
15-100-7	-	2	-	2	-	2
15-100-10	-	2	-	2	-	2
15-120-10	-	2	-	2	-	2
20-150-10	-	2	-	2	-	2
20-180-12	-	1	-	1	-	1
25-200-12	-	2	-	2	-	2
30-250-10	-	2	-	2	-	2
30-270-10	-	2	-	2	-	2
35-300-10	-	2	-	2	-	2
35-310-12	-	2	-	2	-	2

Solution by the proposed GVNS-based heuristic algorithms lead to opening the minimum required number of depots. As shown in Table 4.15, the proposed GVNS algorithms managed to open equal or less depots than the CPLEX solver for the case of small-sized instances. In all problem cases the same depots are selected to be opened. The reason behind this fact, is that in all these cases the structure of locations are kept unmodified. A more detailed information about the opened depots and their planned capacity levels is given in Table 4.16.

TABLE 4.16: The opened depots and their capacity levels.

Instance	Decisions	Values		Instance	Decisions	Values	
4-9-3	depots	depot_2	depot_3	9-70-5	depots	depot_4	depot_5
	cap. level	level_2	level_3		cap. level	level_3	level_3
4-10-3	depots	depot_3	depot_4	9-75-7	depots	depot_3	depot_9
	cap. level	level_2	level_1		cap. level	level_2	level_3
4-10-5	depots	depot_1	depot_3	9-85-5	depots	depot_3	depot_9
	cap. level	level_2	level_2		cap. level	level_4	level_4
4-12-5	depots	depot_1	depot_4	9-88-7	depots	depot_1	depot_2
	cap. level	level_1	level_2		cap. level	level_3	level_2
4-15-3	depots	depot_4		10-90-7	depots	depot_2	depot_4
	cap. level	level_2			cap. level	level_1	level_3
5-12-3	depots	depot_1		15-100-7	depots	depot_8	depot_9
	cap. level	level_3			cap. level	level_2	level_4
5-15-3	depots	depot_5		15-100-10	depots	depot_7	depot_14
	cap. level	level_2			cap. level	level_2	level_4
5-15-5	depots	depot_4	depot_5	15-120-10	depots	depot_9	depot_15
	cap. level	level_1	level_1		cap. level	level_3	level_4
5-18-3	depots	depot_1	depot_3	20-150-10	depots	depot_1	depot_11
	cap. level	level_3	level_2		cap. level	level_3	level_5
5-20-3	depots	depot_1	depot_4	20-180-12	depots	depot_14	
	cap. level	level_1	level_2		cap. level	level_2	
6-40-5	depots	depot_3	depot_6	25-200-12	depots	depot_11	depot_13
	cap. level	level_2	level_1		cap. level	level_5	level_5
7-52-5	depots	depot_4	depot_6	30-250-10	depots	depot_4	depot_20
	cap. level	level_3	level_2		cap. level	level_3	level_3
7-55-7	depots	depot_3	depot_7	30-270-10	depots	depot_13	depot_27
	cap. level	level_2	level_4		cap. level	level_2	level_2
8-60-5	depots	depot_3	depot_6	35-300-10	depots	depot_16	depot_24
	cap. level	level_2	level_2		cap. level	level_1	level_4
8-65-7	depots	depot_2	depot_5	35-310-12	depots	depot_26	depot_29
	cap. level	level_3	level_3		cap. level	level_4	level_3

Table 4.17 provides the fleet composition for the mixed-fleet problem case as it has been obtained by CPLEX solver for some of the small-sized instances and the $GVNS\_5^*_{Init2}$ for all problem instances. The fleet composition decided by $GVNS\_5^*_{Init2}$, corresponds to the best found solution for each problem instance. The letter "L" means light-duty vehicle and the letter "M" is used for medium-duty vehicles.

TABLE 4.17: The fleet composition by each method.

Instance	CPLEX	$GVNS\_5^*_{Init2}$	Instance	CPLEX	$GVNS\_5^*_{Init2}$
4-9-3	3 L & 1 M	2 L & 2 M	9-70-5	-	8 L
4-10-3	3 L & 3 M	4 L & 4 M	9-75-7	-	3 L
4-10-5	2 L & 1 M	2 L & 1 M	9-85-5	-	4 L
4-12-5	3 L & 1 M	2 L & 2 M	9-88-7	-	5 L
4-15-3	2 L & 1 M	2 L	10-90-7	-	1 L & 1 M
5-12-3	4 L & 1 M	5 L	15-100-7	-	2 L & 1 M
5-15-3	-	4 L & 3 M	15-100-10	-	2 L & 1 M
5-15-5	-	10 L	15-120-10	-	4 L & 1 M
5-18-3	-	8 L	20-150-10	-	5 L & 1 M
5-20-3	-	4 L & 1 M	20-180-12	-	9 L & 1 M
6-40-5	-	7 L	25-200-12	-	8 L & 1 M
7-52-5	-	4 L	30-250-10	-	4 L & 1 M
7-55-7	-	3 L	30-270-10	-	3 L & 1 M
8-60-5	-	13 L & 2 M	35-300-10	-	10 L
8-65-7	-	28 L	35-310-12	-	4 L & 1 M

Despite the efficiency of the proposed solution methods, a few limitations of this work should be mentioned. First, an alternative initial order of shaking operators in the adaptive shaking mechanisms may lead to further improvements. Moreover, focused on the strength of the shaking, five different values were examined. Further improvements can potentially achieved by investigating other values. Finally it is not possible to formally assess the quality of the obtained solution with respect to the truly optimal.

4.6 Concluding remarks

Sustainability is a crucial factor of a company's growth. In this regard, this work studies a new complex supply chain network optimization problem, which integrates both economic and environmental decisions. As commercial solvers cannot solve realistic cases of such complex problems, GVNS-based heuristic algorithms were developed for solving medium- and large-sized instances. The shaking mechanism in a VNS-based heuristic has a significant role in its performance. Thus, new adaptive shaking techniques are proposed, as a crucial intelligent learning component of the proposed solution method. This intelligent mechanism uses past experience in order to improve the performance of the algorithm. In these shaking methods, the shaking operators are ordered following two different rules. According to the first one, the operators set in an order, based on their complexity, while in the second one their ordering is performed randomly. During the execution of the algorithms, the shaking operators are re-ordered in accordance with the number of improvements achieved by using each of them in the previous iteration. The GVNS schemes using the proposed adaptive shaking mechanisms are proved more efficient on the solution of such complex supply chain network optimization problems than the GVNS using the classic intensified shaking. Furthermore, the impact of using homogeneous fleet (either light- or medium-duty vehicles) and mixed-fleet is examined not only from an economic perspective, but also from an environmentally point of view. A computational analysis illustrates that by using a mixed-fleet both economical and environmental benefits can be achieved. The impact of using an alternative initialization rule is also investigated and the obtained solutions, especially on ten large-sized instances, were further improved by 5%. The results from the extended numerical analysis illustrate the integration of the proposed models and solution techniques in an intelligent tool which can assist decision makers to derive fast and reliable decisions for the optimal design and operation of complex supply chains.

Chapter 5

Optimization of CAR T-cell therapies Supply Chains

5.1 Introduction

CAR T-cell therapies are novel cancer immunotherapies where cells are removed from patients and delivered to specialized manufacturing facilities to be properly re-engineered. Then, the therapies are delivered back to specialized hospitals and are administered to patients by infusion. Due to the personalised and sensitive nature of such therapies, several challenges are involved in their supply chain. One major CAR T-cell therapies supply chain challenge is the potential bottleneck of the administration process in the capacitated specialized hospitals, as the demand increases. This requires an intervention in the administration process of CAR T-therapies through a new network structure. Herein, we consider an integrated CART T-cell therapies supply chain network involving both design and operational decisions. This approach exploits similar characteristics with the manufacturing supply chain networks (Barbosa-Póvoa et al., 2018; Drexl & Schneider, 2015). The strategic planning problem is related mainly with design decision such as the opening of manufacturing centres and the selection of specialized hospitals as well as the fleet composition of mobile medical units and transportation vehicles. In contrast, operative planning decisions are mainly related with vehicle routing, which cannot be underestimated in the efficient operation of the underlying supply chain network. Integration of such decisions into a single MIP model results in a computationally intractable problem even for small size problems of limited practical interest. For example, the number of patients and local treatment facilities in such a supply chain network may be hundreds or even thousands, and in addition to other type of decisions, the resulting MIP model is typically characterized by thousands of binary variables. We will show later, that only very small problems can be solved by using state-of-the-art MILP solvers. Consequently, this requires special solutions techniques which can systematically generate quick solutions of good quality by sacrificing part of the optimality.

5.2 Problem statement

A new decentralized process of CAR T-cell therapies' administration is proposed in this work to address among others, the need to debottleneck the specialized hospitals due to their limited capacities, as the demand for such therapies increases rapidly. Furthermore, this decentralized approach can also increase the overall efficiency and service quality of the underlying supply chain network, by simultaneously optimizing key design and operational decisions as highlighted in Wang et al. (2019) and Papathanasiou et al. (2020). It should be noted that today the network representation is still an open research and medical challenge. Currently, very limited attention has been placed on the design and operation of CAR T-cell therapies supply chains, while the network representation is still an open research challenge. The overall process of CAR T-cell therapies relies on four main entities (Boyiadzis et al., 2018). A cancer patient visits a specialized treatment facility and a blood sample is collected. Then, the collected blood is sent to the manufacturing centre where the therapy is produced. Finally, the patient revisits the treatment facility to receive the required bridge chemotherapy (a chemotherapy administered in the time between collection of autologous T-cells and the infusion of CAR T-cell therapy) and the cell therapy is delivered to the treatment facility in order to be administered to the patient.

This work proposes a decentralized administration process of CAR T-cell therapies, which can be practically implemented by a new supply chain network structure consisting of five main entities: (i) manufacturing centres, (ii) central specialized hospitals, (iii) patients, (iv) transportation vehicles of patient samples, (v) mobile medical units and local treatment facilities. More specifically, a specialized hospital operates as a coordinator of the therapy process. A cancer patient visits a specialized hospital and a blood sample is collected. A transportation vehicle, owned by this hospital, delivers the collected samples to a manufacturing centre where production occurs. Then, the patient visits the local treatment facility, close to his site (home, nursing home etc.), in order to receive a bridge chemotherapy. The patient-specific CAR T-cell therapy is delivered to the mobile medical unit, which visits the local treatment facilities that have been assigned to it. The administration of the therapy is performed in the mobile medical unit for each patient in each local treatment facility by two specialized medical practitioners. Following the administration of the therapy, the patient remains at the local treatment facility to monitor his response to the treatment and address possible side effects. The proposed network structure is illustrated in Figure 5.1.

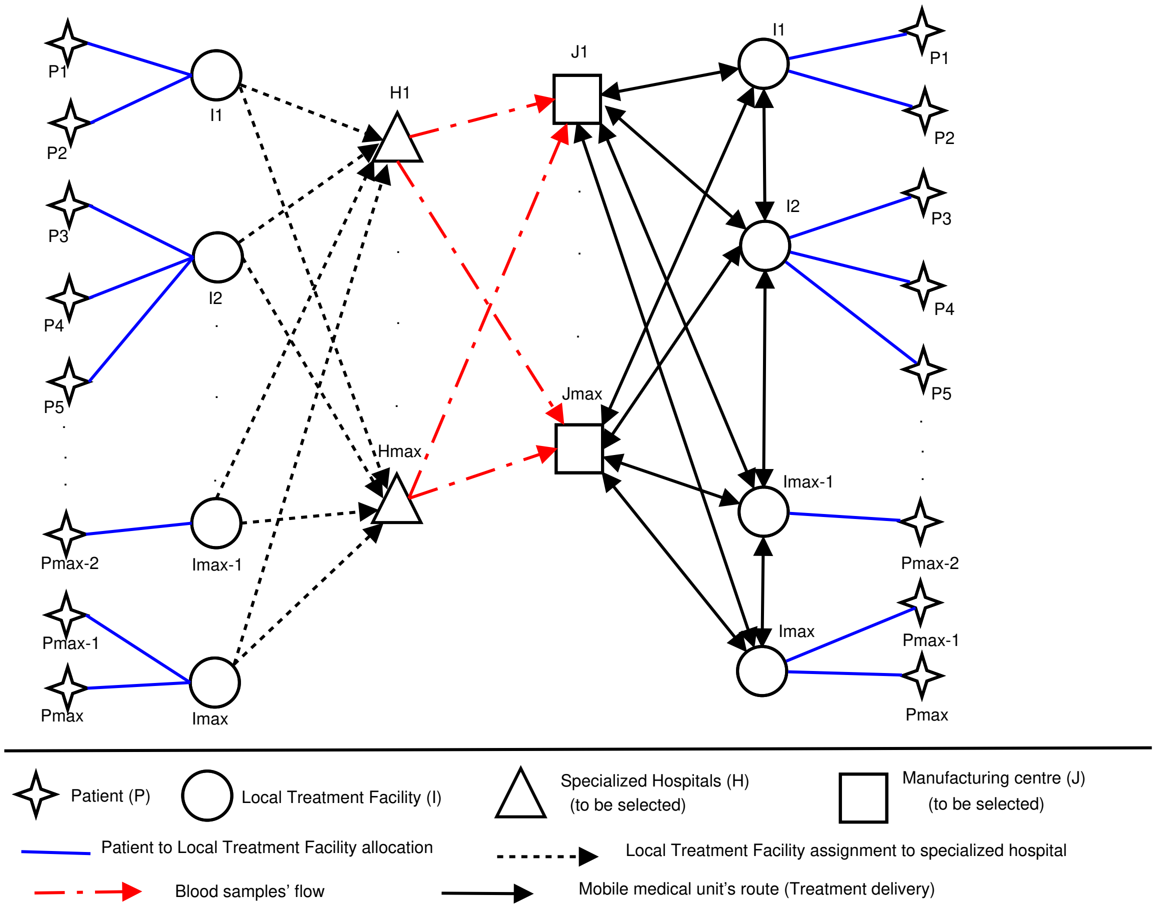

FIGURE 5.1: The proposed CAR T-cell supply chain therapies structure

The network representation in Figure 5.1 illustrates the information flow and the main design decisions to be made. For instance, a strategic-level design decision is the assignment of patients to specialized hospitals through their local treatment facilities. This allocation is indicated by the dotted arcs. Detailed operational decisions related to inner specialized hospitals actions, such as the scheduling of patients visits are not considered in this study. Thus, this information is not shown in the proposed network representation. The problem addressed in this work as formally stated as follows:

Given:

- A set of local treatment facilities, I, which service a set of patients.
- A set of capacitated hospitals, H.
- A set of potential capacitated manufacturing centres, J.
- A set of time periods, T.
- A period-variable demand of treatment dosage for each patient.
- A set of homogeneous capacitated fleet of mobile medical units, K.
- A set of homogeneous capacitated fleet of transport vehicles, M.

Determine:

- how many and which of the potential manufacturing centres should be active (an already operational manufacturing centre can either be selected to be utilized or a new manufacturing centre to be opened),
- the assignment of local treatment facilities to hospitals,
- the connection of local treatment facilities with the selected manufacturing centres,
- the number of mobile medical units which are needed to service the demand of patients,
- the allocation of local treatment facilities to the selected mobile medical units in each time period,
- the number of samples' transport vehicles for each hospital,
- the routes of mobile medical units,
- the shipping of the blood samples from hospitals to the selected manufacturing centres.

The main assumptions of the proposed model are:

- Detailed scheduling aspects in the manufacturing centres are not considered.
- It is considered that T-cell therapies make use of fresh patients' blood samples (leukapheresis product) and CAR T-cells.

- The management of potential serious side effects, such as cytokine release syndrome or neurotoxicity, is not considered at this stage.
- CAR-T cell chemotherapy will be administered in the mobile medical units.
- A local treatment facility will be serviced by exactly one hospital and one manufacturing centre in the time horizon.
- Mobile medical units will be routed from manufacturing centres to the local treatment facilities.
- A mobile medical unit will depart and return to the same manufacturing centre.
- A local treatment facility will be serviced by at most one mobile medical unit in each time period.
- Patients' blood samples will be shipped directly from hospitals to manufacturing centres.
- The demand associated to a local treatment facility, indicates the number of patients serviced by this treatment facility in each time period.
- Two parts of patient samples are required to produce a CAR T-cell therapy.

A salient feature of the proposed modeling and optimization framework is the integration of both design (e.g. selection of manufacturing centres, number of mobile medical units etc) and operational decisions (routes, shipping quantities over time etc), thus exploring the synergistic benefit between the two decisions.

It is worth mentioning that the proposed network representation is not limited to a specific geographic area/location. It can properly describe networks in wider areas, such as states or countries. In this context the decisions such as the selection of the optimal number of manufacturing centres and their capacity are important.

5.3 Mathematical formulation

The model sets, parameters and variables are summarized in Tables 5.1 , 5.2, 5.3 and 5.4.

TABLE 5.1: Sets of the mathematical model

Indices	Explanation
N	Set of nodes
H	Set of specialized hospitals
J	Set of potential manufacturing centres (MF)
I	Set of local treatment facilities (LTF)
Z	set of MF and LTF, $(J \cap I)$
K	Set of mobile medical units
M	Set of blood samples' transport vehicles
V	Set of vehicles $(K \cap M)$
T	Set of time periods

TABLE 5.2: Vehicles' parameters

Parameter	Explanation	Value
fd_k	driver wage $(Euros/h)$	11
fd_m	driver wage $(Euros/h)$	9
Q_v	loading capacity of vehicle v	instance-dependent
VFC_v	usage cost of vehicle v	30300 or 22000
$speed$	speed level of vehicles (km/h) 55	

The hourly wage of drivers of transport vehicles is set equal to the lowest wage reported by U.S. Bureau of Labor Statistics as the wage of ambulance drivers and the hourly wage of drivers of mobile medical units are set equal to the median hourly wage provided in this report [1]. The cost of using a mobile medical unit is set to 30300 Euros, while the acquisition of a blood samples' transport vehicle is charged with 22000 Euros.

The current literature does not provide any information about the acquisition and utilization costs of mobile medical units. Therefore, the value of VFC_v, for the case of mobile medical units, was decided based on information from online sources, such as websites of mobile medical unit manufacturers or websites of medical-based news. It is obvious that the final price is determined by several case specific facts. An average annual value is up

[1] https://www.bls.gov/oes/current/oes533011.htm

to 472437 euros [2] [3]. By excluding staffing costs and considering a sufficient amount for medical expendables the total operational cost of 30300 euros is reached.

A transport vehicle of blood samples is usually a transport van. The average cost of a transport van is 20.000 euros (based on the latest prices of VW (VW Cuddy), Ford (Ford Limited) and Citroen (Citroen Jumpy) - three of the most preferred manufacturers in transport industry). According to the World Health Organization (WHO), a transport vehicle of blood samples must be equipped with specific cold chain equipment [4], such as cold boxes and temperature monitoring devices. The average price of a cold box is 250 euros and of a temperature monitoring device is 100 euros, based on the prices reported in the supply catalogue of Unicef [5]. It is assumed that four cold boxes are a sufficient amount for each transport vehicle (due to their limited capacity), while each cold box needs a temperature monitoring device. Consequently, the acquisition cost for each transport vehicle is up to 22000 Euros.

While an attempt has been made to use approximate cost data, it is clear that in a realistic CAR T-cell supply chain these values will be provided more accurately.

TABLE 5.3: Model parameters

Notation	Explanation
f_j	Fixed opening cost of manufacturing center j
Cap_j	Storage capacity of manufacturing center j
$HCap_h$	Service capacity of hospitals h
d_{it}	period-variable treatment demand of local treatment facility i
c_{nn_1}	travelling distance between the pair of locations (n, n_1)
p_j	procurement cost in manufacturing centre j for each local treatment facility
$HPSC_h$	service cost per local treatment facility assigned to hospital h
TAT	treatment's administration time
TTD	treatment time durability
STD	specimens time durability
fp	medical practitioner's wage
$samples$	blood samples required to produce a CAR T-cell therapy (= 2)
$Mega$	a very big number (= 100000)

The infusion time of a CAR T-cell therapy takes in average one hour [6].

[2] https://www.laboit.com/health-dental/health-faq.html
[3] https://www.healthcarefinancenews.com/news/mobile-health-vans-value-proposition
[4] https://www.who.int/bloodsafety/processing/cold_chain/en/
[5] https://supply.unicef.org/all-materials/cold-chain-equipment.html
[6] https://www.mayoclinic.org/departments-centers/car-t-cell-therapy-program/sections/gnc-20405547

According to the values of the treatment and specimens time durability, a detailed explanation is provided in section 5.5.3. The value of medical practitioner's hourly wage was set equal to the average of the lowest and the highest wage of a registered nurse in European Union according to the wages reported by Economic Research Institute (University of California, Irvine) [7].

TABLE 5.4: Decision variables

Notation	Explanation
y_j	1 if manufacturing centre j is opened; 0 otherwise
psa_{ihj}	1 if local treatment facility i is allocated from hospital h to manufacturing centre j; 0 otherwise
vs_{kt}	1 if mobile medical unit k is selected in period t; 0 otherwise
vha_{hm}	1 if blood samples' transportation vehicle m is owned by hospital h; 0 otherwise
x_{nn_1vt}	1 if node n_1 is visited after n in period t by vehicle v; 0 otherwise
TD_{ikt}	the total dosage of treatment delivered to patient i in period t by mobile medical unit k
SQ_{hjmt}	blood samples' quantity delivered from hospital h to the manufacturing centre j with vehicle m in period t
tt_{nn_1t}	the travel time between nodes n and n_1 in time period t

Additionally, a positive variable U_{ikt} $(0 \leq U_{ikt} \leq |I|-1)$ is used to implement the classic subtour elimination constraint of Miller-Tucker-Zemlin (Bektaş & Gouveia, 2014).

The objective of the problem is to minimize the total cost of the underlying supply chain network:

$$min \sum_{j\in J} f_j y_j \; + \sum_{h\in H}\sum_{j\in J}\sum_{i\in I}\sum_{t\in T} psa_{ihj}\cdot\left(p_j + HPSC_h\right) \; + \sum_{n\in N}\sum_{n_1\in V}\sum_{t\in H}\sum_{v\in V} c_{nn_1}\cdot x_{nn_1vt} \; + \sum_{k\in K}\sum_{t\in T} vs_{kt}\cdot VFC_k \; + \sum_{h\in H}\sum_{m\in M} vha_{hm}\cdot VFC_m \; + \sum_{h\in H}\sum_{j\in J}\sum_{m\in M}\sum_{t\in T}\frac{\left(c_{hj}\cdot x_{hjmt}\cdot fd_m\right)}{speed} \; + \sum_{z\in Z}\sum_{z_1\in Z}\sum_{k\in K}\sum_{t\in T}\frac{\left(c_{zz_1}\cdot x_{zz_1kt}\cdot(fd_k + 2\cdot fp)\right)}{speed} \tag{5.1}$$

The first term of the objective function represents the location cost of opening manufacturing centers. The second term quantifies the allocation-based costs of the local treatment facilities, such as procurement and servicing costs. The third term represents the general routing costs, such as insurance and maintenance costs. The following two terms represent the costs of vehicles selection in each period for mobile medical units and blood samples' transport vehicles, respectively. The subsequent term provides the cost of wages of blood samples' transport vehicles' drivers, while the last term represents the cost of wages both of drivers of mobile medical units and medical staff.

[7] https://www.salaryexpert.com/

Model constraints:
The balance between interior and exterior flow of vehicles is guaranteed by the following constraints:

$$\sum_{n_1 \in N} x_{nn_1vt} - \sum_{n_1 \in N} x_{n_1nvt} \quad = 0, \quad \forall\, n \in N,\, v \in V,\, t \in T \tag{5.2}$$

A local treatment facility will be served by at most one mobile medical unit in each time period:

$$\sum_{z \in Z} \sum_{k \in K} x_{izkt} \leq 1, \quad \forall\, i \in I,\, t \in T \tag{5.3}$$

$$\sum_{z \in Z} \sum_{k \in K} x_{zikt} \leq 1, \quad \forall\, i \in I,\, t \in T \tag{5.4}$$

A mobile medical unit will be moved between two nodes, only if it is selected:

$$x_{zz_1kt} \leq vs_{kt}, \quad \forall\, z, z_1 \in Z,\, k \in K,\, t \in T \tag{5.5}$$

A mobile medical unit will be selected only if it is scheduled to perform a route in a specific time period:

$$vs_{kt} \leq \sum_{z \in Z} \sum_{z_1 \in Z} x_{zz_1kt}, \quad \forall\, k \in K,\, t \in T \tag{5.6}$$

A mobile medical unit will perform at most one route in each period:

$$\sum_{i \in I} \sum_{j \in J} x_{ijkt} \leq 1, \quad \forall\, k \in K,\, t \in T \tag{5.7}$$

A blood samples' transport vehicle will be owned by at most one hospital:

$$\sum_{h \in H} vha_{hm} = 1, \quad \forall\, m \in M \tag{5.8}$$

A blood samples' transport vehicle will be moved from a hospital to a manufacturing centre, only if it is owned by that hospital:

$$\sum_{j \in J} x_{hjmt} \leq vha_{hm}, \quad \forall\, h \in H,\, m \in M,\, t \in T \tag{5.9}$$

A blood samples' transport vehicle will be owned by a hospital only if it is scheduled to be used:

$$vha_{hm} \leq \sum_{j \in J} \sum_{t \in T} x_{hjmt}, \quad \forall\, h \in H,\ m \in M \tag{5.10}$$

A hospital will be linked with a manufacturing centre only if there are blood samples to be delivered to that manufacturing centre:

$$x_{hjmt} \leq SQ_{hjmt}, \quad \forall\, h \in H,\ j \in J,\ m \in M,\ t \in T \tag{5.11}$$

Blood samples will be delivered from a hospital to a manufacturing centre, only if there is an active link between them in a time period:

$$SQ_{hjmt} \leq Mega \cdot x_{hjmt}, \quad \forall\, h \in H,\ j \in J,\ m \in M,\ t \in T \tag{5.12}$$

A mobile medical unit will depart only from an opened manufacturing centre:

$$\sum_{i \in I} \sum_{k \in K} \sum_{t \in T} x_{jikt} \geq y_j, \quad \forall\, j \in J \tag{5.13}$$

$$\sum_{i \in I} x_{jikt} \leq y_j, \quad \forall\, j \in J, \forall\, k \in K,\ t \in T \tag{5.14}$$

A local treatment facility will be allocated to exactly one hospital and one manufacturing centre:

$$\sum_{j \in J} \sum_{h \in H} psa_{ihj} = 1, \quad \forall\, i \in I \tag{5.15}$$

A local treatment facility will be served by a manufacturing centre, if it is assigned to that manufacturing centre:

$$\sum_{b \in I} x_{bjkt} + \sum_{z \in Z \setminus \{i\}} x_{izkt} \leq 1 + \sum_{h \in H} psa_{ihj} \quad \forall\, i \in I,\ j \in J,\ \mathrm{k} \in \mathrm{K},\ t \in T \tag{5.16}$$

A manufacturing centre will service a local treatment facility, only if it is opened:

$$\sum_{h \in H} psa_{ihj} \leq y_j, \quad \forall\, i \in I,\ j \in J \tag{5.17}$$

The patients' blood samples' delivered from a hospital to a manufacturing centre must be equal to the required blood samples, in order to produce the corresponding demand of patients in local treatment facilities allocated both to that hospital and the corresponding manufacturing centre:

$$\sum_{i \in I} psa_{ihj} \cdot d_{it} \cdot samples = \sum_{m \in M} SQ_{hjmt}, \quad \forall\, h \in H,\, j \in J,\, t \in T \tag{5.18}$$

The blood sample delivered by a transport vehicle must not exceed its capacity:

$$\sum_{h \in H} \sum_{j \in J} SQ_{hjmt} \leq QCap_m, \quad \forall\, m \in M,\, t \in T \tag{5.19}$$

The total amount of treatments delivered by a mobile medical unit must not exceed its capacity:

$$\sum_{i \in I} TQ_{ikt} \leq QCap_k, \quad \forall\, k \in K,\, t \in T \tag{5.20}$$

The local treatment facility allocated to a hospital must not exceed its service limit:

$$\sum_{i \in I} \sum_{j \in J} psa_{ihj} \leq HCap_h, \quad \forall\, h \in H \tag{5.21}$$

The demand of patients in a local treatment facility allocated to a manufacturing centre must not exceed its capacity:

$$\sum_{i \in I} \sum_{h \in H} psa_{ihj} \cdot \sum_{t \in T} d_{it} \leq Cap_j, \quad \forall\, j \in J \tag{5.22}$$

Treatment quantities will be delivered to a patient in a local treatment facility, only if that local treatment facility is visited by a mobile medical unit:

$$TQ_{ikt} \leq Mega \cdot \sum_{z \in Z} x_{ixkt}, \quad \forall\, i \in I,\, k \in K,\, t \in T \tag{5.23}$$

$$\sum_{z \in Z} x_{izkt} \leq Mega \cdot TQ_{ikt}, \quad \forall\, i \in I,\, k \in K,\, t \in T \tag{5.24}$$

Treatment quantities delivered by a mobile medical unit to a local treatment facility must not exceed the capacity of this mobile medical unit:

$$TQ_{ikt} \leq QCap_k, \quad \forall\, i \in I,\, k \in K,\, t \in T \tag{5.25}$$

The quantity of treatment delivered to a local treatment facility must be equal to the demand of patients in this local treatment facility:

$$\sum_{k \in K} TQ_{ikt} = d_{it}, \quad \forall\, i \in I,\, t \in T \tag{5.26}$$

The time needed to move between two nodes is related to their distance and the speed of the moving vehicle:

$$tt_{nn_1t} = \left(c_{nn1} \cdot \sum_{v \in V} x_{nn_1vt} \right) / speed, \quad \forall\, n, n_1 \in N, n \neq n_1,\ t \in T \tag{5.27}$$

The time needed to a mobile medical unit to perform its route must not exceed the upper time limit:

$$\sum_{z \in Z} \sum_{z_1 \in Z} tt_{zz_1t} \leq TTD, \quad \forall\, k \in K,\ t \in T \tag{5.28}$$

The traveling time and treatment administration time must not exceed a given upper time limit:

$$\sum_{z \in Z} \sum_{i \in I} \left(tt_{zit} + ((TAT \cdot d_{it}) \cdot x_{zikt}) \right) \leq TTD, \quad \forall\, k \in K,\ t \in T \tag{5.29}$$

The time needed to a blood samples' transport vehicle to move from a hospital to a manufacturing centre must not exceed a specified upper time limit:

$$tt_{hjt} \leq STD, \quad \forall\, h \in H,\ j \in J,\ t \in T \tag{5.30}$$

The following routing-related constraints impose forbidden routes. No vehicle movements are allowed between two manufacturing centres in each time period:

$$x_{jj_1vt} = 0, \quad \forall\, j, j_1 \in J,\ v \in V,\ t \in T \tag{5.31}$$

A mobile medical unit cannot move between a hospital and a manufacturing centre in each time period:

$$x_{jhkt} = 0, \quad \forall\, j \in J,\ h \in H,\ k \in K,\ t \in T \tag{5.32}$$

$$x_{hjkt} = 0, \quad \forall\, j \in J,\ h \in H,\ k \in K,\ t \in T \tag{5.33}$$

No vehicle movements are allowed between a local treatment facility and a hospital in each time period:

$$x_{ihvt} = 0, \quad \forall\, i \in I,\ h \in H,\ v \in V,\ t \in T \tag{5.34}$$

$$x_{hivt} = 0, \quad \forall\, h \in H,\ i \in I,\ v \in V,\ t \in T \tag{5.35}$$

A blood samples' transport vehicle cannot move between a manufacturing center and a local treatment facility, neither between two local treatment facilities:

$$x_{ijmt} = 0, \quad \forall\, i \in I,\ j \in J,\ m \in M,\ t \in T \tag{5.36}$$

$$x_{jimt} = 0, \quad \forall\, j \in J,\ i \in I,\ m \in M,\ t \in T \tag{5.37}$$

$$x_{ibmt} = 0, \quad \forall\, i, b \in I,\ m \in M,\ t \in T \tag{5.38}$$

The Miller-Tucker-Zemlin subtour elimination constraints take the following form:

$$U_{ikt} - U_{bkt} + (|I| \cdot x_{ibkt}) \leq |I| - 1, \quad \forall\, i, b \in I,\ k \in K,\ t \in T \tag{5.39}$$

If a local treatment facility has no patient with a non-zero demand in a time period, then no vehicles' movements are allowed through it:

$$x_{zikt} = 0, \quad \forall\, z \in Z,\ i \in I,\ k \in K,\ t \in T,\ d_{it} = 0 \tag{5.40}$$

$$x_{izkt} = 0, \quad \forall\, i \in I,\ z \in Z,\ k \in K,\ t \in T,\ d_{it} = 0 \tag{5.41}$$

The above problem is a large-scale MILP model.

5.4 Solution Approach

Despite the significant progress that has been made in the past 20 years for the develop of efficient MIP solvers, the underlying NP-hard combinatorial optimization problem is computationally intractable. Even for small-sized problem instances, current state-of-the-art MIP solvers usually require several hours to report even a feasible solution. Therefore, the development of fast and efficient metaheuristic algorithms is essential for the solution of complex supply chain problems of practical interest often involving simultaneous location, routing and inventory decisions. A metaheuristic algorithm starts with an initial feasible solution and improves it iteratively until a termination criterion is met.

5.4.1 Initial Solution

To build an initial feasible solution, a fast construction heuristic is proposed. The developed heuristic consists of four main stages. In the first stage, the opening of the needed manufacturing centres is decided, by applying a minimum opening cost criterion. This stage is completed when the capacity of opened manufacturing centres is greater or at least equal to the overall demand of patients in the local treatment facilities. The next stage refers to the allocation of local treatment facilities to the opened manufacturing centres following a nearest distance allocation strategy. More specifically, each local treatment facility is allocated to its nearest manufacturing centre if the total demand of the patients in this treatment facility does not exceed the capacity of the selected manufacturing centre and their distance does not lead to any time restrictions' violation. This stage is fully accomplished when all local treatment facilities are assigned to the opened manufacturing centres.

After the execution of the first two stages, the selection of specialized hospitals and the allocation of local treatment facilities to them is decided in the third stage. To this end, the manufacturing centre, which services most of the local treatment facilities, is initially selected and the nearest specialized hospital to that manufacturing centre is chosen. Next, the local treatment facilities of the currently selected manufacturing centre, are sequentially allocated to the closest hospital until either no more treatment facilities are unallocated or the capacity of the hospital is exceeded. This stage is executed iteratively in order to allocate all the local treatment facilities to specialized hospitals. In the final stage, the fleet selection and routing decisions are made. The fleet selection both of blood samples' transportation vehicles and mobile medical units is performed sequentially by considering the given demand and vehicles' capacities. The blood samples' transportation vehicles are owned by selected hospitals, while mobile medical units are used by manufacturing centres. Finally, the route of each mobile medical unit in each time period is scheduled.

5.4.2 General Variable Neighborhood Search

Shaking method. The intensified shaking method is used as the shaking step of the proposed algorithm. It relies on two shaking search operators, the Inter-route Swap, which randomly selects two local treatment facilities allocated to different routes and swaps them, and the Intra-route Relocate, which is applied in a randomly selected single route in a time period and then removes a randomly selected local treatment facility from its position in this route and re-inserts it in a different position in the same route. This method make use of a parameter, which indicates in each algorithm iteration how many times the randomly selected shaking operator is applied. The pseudocode of the developed shaking procedure is provided in Algorithms

23. S is the incumbent solution, k is the parameter of shaking iterations and l_{max} is the number of shaking operators.

Algorithm 23 Shaking Procedure

```
procedure SHAKE(S, k, l_max)
    l = random_integer(1, l_max)
    for i ← 1, k do
        select case(l)
        case(1)
        S' ← Inter − route_Exchange(S)
        case(2)
        S' ← Intra_Relocate(S)
        end select
    end for
    Return S'
end procedure
```

Improvement method. As it has already been highlighted, GVNS uses VND schemes as its main improvement step. In a VND method, local search operators are applied sequentially. The move to the next operator is performed according to a specific criterion. One the most efficient VND scheme proved to be the pVND, in which the search is performed by the same local search operator, as improvements achieved (Hansen et al., 2017). The proposed GVNS uses a pVND as its improvement step. The developed pVND contains the following well-known local search operators:

- **Inter-route Relocate**. Two local treatment facilities allocated to different routes are selected. The first of them is removed from its current position and it is placed exactly after the position of the second selected local treatment facility.

- **Inter-route Exchange**. This operator swaps two local treatment facilities allocated to different routes.

- **Manufacturing centres' Exchange**. A currently selected manufacturing centre is swapped with a currently closed one.

- **Intra-route Relocate**. A local treatment facility is removed from its current position and it relocates to an other position in the same route.

- **Intra-route 2-Opt**. The links between two different pairs of local treatment facilities are broken and reconnected differently.

The pseudocode of the proposed pVND is provided in Algorithm 24.

Algorithm 24 pVND

```
procedure pVND(S, l_max)
    l = 1
    while l ≤ l_max do
        select case(l)
        case(1)
        S′ ← Inter − route_Relocate(S)
        case(2)
        S′ ← Inter − route_Exchange(S)
        case(3)
        S′ ← ManufacturingCentres_Swap(S)
        case(4)
        S′ ← Intra − route_Relocate(S)
        case(5)
        S′ ← Intra − route_2 − Opt(S)
        end select
        if f(S′) < f(S) then
            S ← S′
        else
            l = l + 1
        end if
    end while
    Return S
end procedure
```

The proposed GVNS algorithm is shaped as shown in Algorithm 25.

Algorithm 25 GVNS

procedure GVNS($S, k_{max}, max\_time, l_{max}$)
    while $time \leq max\_time$ **do**
        for $k \leftarrow 1, k_{max}$ **do**
            $S^* = Shake(S, l)$
            $S' = pVND(S^*, l_{max})$
            if $f(S^*) < f(S)$ **then**
                $S \leftarrow S^*$
            end if
        end for
    end while
    return S
end procedure

The execution of the GVNS algorithm depends on a given stopping time criterion which is indicated by the parameter max_time. Other parameters that must be provided to the GVNS are the incumbent solution S, the number of the predefined local search operators l_{max} and the strength of the shaking procedure, k_{max} which denotes the maximum number of a shaking operator's application per iteration.

A graphical illustration of the proposed algorithm is shown in Figure 5.2.

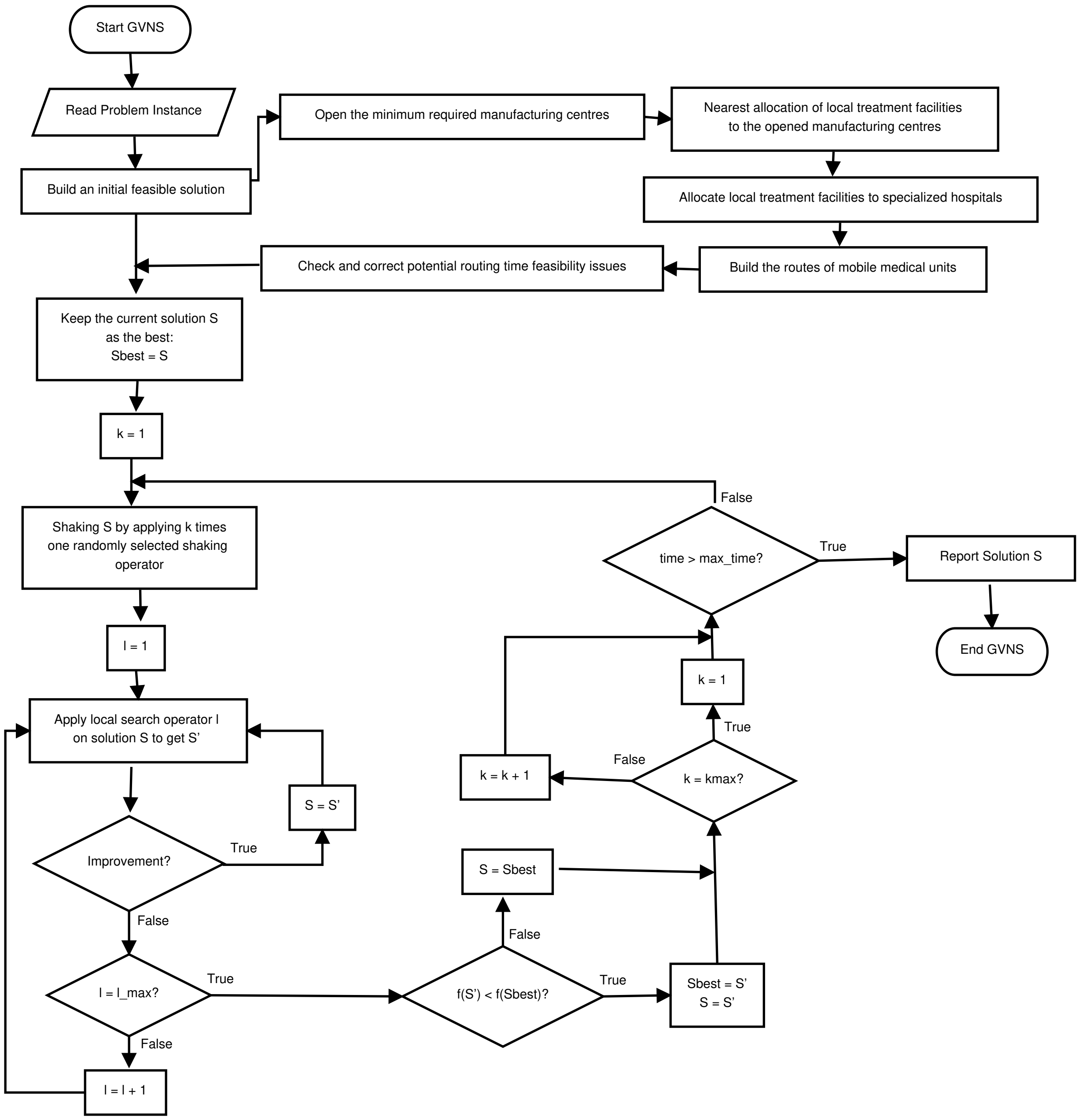

FIGURE 5.2: The flowchart of the proposed GVNS algorithm.

5.5 Computational analysis and applications

The proposed solution method has been implemented in Fortran. The computational experiments were performed on a laptop PC running Windows 10 Home 64-bit with an Intel Core i7-6700HQ CPU at 2.6 GHz and 16 GB RAM and using the Intel Fortran compiler 18.0. The time stopping criterion was set at 30s. The proposed MIP model was also implemented in GAMS (GAMS 24.9.1) (Brooke et al., 1998) and solved using the CPLEX 12.7.1.0 solver. The execution time of CPLEX was initially set at two hours.

The reported solution values is the average values of 10 algorithm runs for each problem instance.

5.5.1 Problem instances & parameter setting

To validate the proposed model and evaluate the performance of the developed solution method, 20 problem instances were randomly generated following specific guidelines (Zhang et al., 2014). Moreover, procurement and service costs has been randomly generated in the intervals $[10, 50]$ and $[30, 100]$ respectively. The names of the generated instances have the form X-Y-Z-K, where X denotes the number of available hospitals (to be selected), Y is the number of potential manufacturing centers (to be selected), Z is the number of local treatment facilities and finally, K denotes the number of time periods. All problem data are provided at http://pse.cheng.auth.gr/index.php/publications/benchmarks/.

The shaking procedure is a crucial part of any VNS-based metaheuristic algorithm, as it is the main mechanism for escaping from local optimum solutions. The shaking strength, denoted by the parameter k_{max}, critically affects the performance of the shaking procedure and consequently the performance of the algorithm. Thus, a k_{max} analysis was performed by examining the impact of three commonly used and well-performed values (10, 15, 20) in the final solution of the algorithm. Table 5.5 presents the total cost obtained by using each of these values for each problem case.

TABLE 5.5: k_{max} analysis on the $GVNS$ performance

Instance	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$
3-3-12-3	195,804.2	195,792.1	195,841
2-3-11-4	302,897.5	305,890.4	302,893
2-3-12-4	237,184.7	237,244.3	237,176.4
2-3-10-5	367,296.7	367,296.7	367,296.7
3-4-18-4	474,588.1	474,615.6	474,609.3
3-4-20-5	516,962.4	516,949.3	516,952.9
4-5-60-5	1,303,514	1,303,518	1,303,503
5-8-80-5	1,689,163	1,689,169	1,689,169
5-8-90-5	2,208,740	2,208,746	2,208,730
5-10-95-5	1,961,402	1,961,402	1,961,402
5-10-100-5	2,239,688	2,239,688	2,239,688
5-10-105-7	3,085,174	3,085,174	3,085,174
5-10-110-7	3,153,722	3,153,720	3,153,718
6-15-120-7	3,438,138	3,438,138	3,438,140
6-15-125-7	3,543,181	3,543,181	3,543,166
6-15-130-7	3,657,759	3,657,760	3,657,760
6-15-135-7	3,937,670	3,937,667	3,937,669
7-20-140-7	3,944,267	3,944,270	3,944,280
10-20-200-7	5,386,204	5,386,203	5,386,204
10-20-250-7	6,641,862	6,641,853	6,641,834
Average	2,414,260.88	2,414,414	2,414,260.32

All three k_{max} choices provide almost equal values, especially the choices of $k_{max} = 10$ and $k_{max} = 20$. However, the choice $k_{max} = 20$ provides slightly better results and therefore it is the one selected for the rest of the study.

5.5.2 Results

This section presents in details the results of GVNS on the solution of the 20 random generated instances. Table 5.6 summarizes the best objective function found.

TABLE 5.6: Total cost of best found solution of each instance.

Instance	TC_{best}
3-3-12-3	195,778.3
2-3-11-4	302,882.6
2-3-12-4	237,159.9
2-3-10-5	367,296.7
3-4-18-4	474,583
3-4-20-5	516,936.8
4-5-60-5	1,303,458
5-8-80-5	1,689,169
5-8-90-5	2,208,696
5-10-95-5	1,961,400
5-10-100-5	2,239,688
5-10-105-7	3,085,174
5-10-110-7	3,153,712
6-15-120-7	3,438,138
6-15-125-7	3,543,164
6-15-130-7	3,657,759
6-15-135-7	3,937,667
7-20-140-7	3,944,271
10-20-200-7	5,386,204
10-20-250-7	6,641,762

Table 5.7 illustrates the number of the opened manufacturing centres (first column), the number of used hospitals (second column), the number of selected transport vehicles of blood samples (third column) and the number of used mobile medical units in its last column.

TABLE 5.7: The number of manufacturing centres, hospitals, transportation vehicles and mobile medical units in the best found solutions

Instance	Manufacturing centres	Hospitals	Transportation vehicles of samples	Mobile medical units
3-3-12-3	1	2	3	2
2-3-11-4	2	2	2	2
2-3-12-4	1	2	2	3
2-3-10-5	2	2	2	2
3-4-18-4	2	3	4	4
3-4-20-5	2	2	3	3
4-5-60-5	2	3	3	9
5-8-80-5	2	2	2	12
5-8-90-5	3	3	10	15
5-10-95-5	3	3	3	13
5-10-100-5	3	4	4	16
5-10-105-7	2	2	3	16
5-10-110-7	2	2	2	17
6-15-120-7	2	2	2	23
6-15-125-7	2	2	2	19
6-15-130-7	2	2	3	19
6-15-135-7	3	3	6	20
7-20-140-7	2	2	2	21
10-20-200-7	2	2	2	34
10-20-250-7	2	3	3	36

In order to evaluate the performance of the proposed GVNS algorithm, a comparison with solutions obtained using CPLEX is made. The results are summarized in Table 5.8.

TABLE 5.8: Compare the results achieved by CPLEX and GVNS on 10 small-sized problem instances

Instance	GAMS/CPLEX	GVNS
3-3-12-3	173,997.74	195,841
2-3-11-4	305,207.27	302,893
2-3-12-4	237,620.38	237,176.4
2-3-10-5	368,281.35	367,296.7
3-4-18-4	N/A	474,609.3
3-4-20-5	N/A	516,952.9
4-5-60-5	N/A	1,303,503
5-8-80-5	N/A	1,689,169
5-8-90-5	N/A	2,208,730
5-10-95-5	N/A	1,961,402

CPLEX can provide near-optimal solution for the smallest problem instance of Table 5.6 within a CPU time limit of two hours. However, it cannot provide any feasible solution for any of the other small-sized instances. Focusing on the smallest instance, CPLEX generates a solution with a total cost of 173,997.74 euros (integrality gap = 0.01%), which is 12.55% better than the solution produced by GVNS algorithm (195,841 euros). By increasing the time limit to three hours, the CPLEX provides a feasible solution only for instance "2-3-12-4" (237,620.38 euros) with an integrality gap of 20.3%, while with a time limit of six hours, feasible solutions of instances "2-3-11-4" and "2-3-10-5" are found (305,207.27 euros and 368,281.35 euros correspondingly). The integrality gaps for these two problem instances are 36.62% and 38.33% respectively. The proposed GVNS algorithm produces slightly better solutions than the CPLEX solver as the size of problem instances increases. Furthermore, for larger problem instances CPLEX does not generate even a feasible solution, while GVNS leads to solution within 30 CPU secs. Thus, the GVNS algorithm can be considered as an efficient solution method for the problem under consideration. More importantly, considering the time-sensitive nature of the specific supply chain problem, it is obvious that the high execution times of CPLEX illustrates the importance for the development of fast and efficient computational methods, such as the proposed GVNS algorithm.

It should be noted that, the reported positive performance of the CPLEX solver on the smallest problem instance may be attributed to a possible better allocation of local treatment facilities to the central specialized hospitals, as this decision affects the selection of blood samples' transportation vehicles. More specifically, the main difference of solutions using CPLEX and GVNS are related to the selection of an extra blood samples' transportation which is decided by GVNS.

5.5.3 Sensitivity Analysis

A sensitivity analysis is performed to reveal the effect of key model parameters on the structure and cost of the underlying network. Eight different scenarios are studied. The effect of changes in the treatment time durability (TTD) and specimen time durability (STD) is investigated. Furthermore, the effect of vehicles and facilities capacity and costs is also considered.

Changes on TTD and STD parameters

A fresh CAR T-cell therapy must be administered within 24 hours either from its production or its remove from the deep cryo-preservation. However, considering seven hours as the average sleeping time needed for an adult to be well-operational (Bener et al., 2017; Kalsi et al., 2018) and one hour for preparation purposes, TTD was set at 16 hours in the basic scenario. Due to the fact that transport vehicles may need to cover longer distances than mobile medical units, STD was set at 18 hours.

According to the European Union rules [8] concerning the maximum driving hours, three different cases are defined. In the first case, a 9 hours time limit is set for daily driving, in the second case a 10 hours time limit is set for driving twice a week and in the final one, the case of emergency-aid vehicles, which are exempt of the previous rules, is considered. Therefore, three scenarios are studied in this subsection.

Scenario_1 (S1). Here, TTD and STD follow the first case of driving hours EU rule and they are set at 9 hours. For this analysis, problem instances "3-3-12-3", "2-3-11-4", "2-3-10-5", "3-4-10-5", "3-4-18-4", "3-4-20-5" and "4-5-60-5" are selected for comparison purpose. Table 5.9 provides the best found total cost and the number of mobile medical units, for each one of problem instances in the basic scenario and S1.

TABLE 5.9: Changes on the total costs and the number of mobile medical units in basic scenario and S1.

Instance	$BestTC_{basic}$	$\|MMU\|_{basic}$	$BestTC_{S1}$	$\|MMU\|_{S1}$
3-3-12-3	195778.3	2	317644.6	4
2-3-11-4	237159.9	2	364245	3
2-3-10-5	367296.7	2	488994.8	4
3-4-18-4	474583	4	626892.7	6
3-4-20-5	516936.8	3	760605.1	6
4-5-60-5	1303458	9	2185467	19

It is observed that in most cases additional mobile medical units are selected in the new scenario in order to satisfy the stricter upper time limits comparing the basic scenario. These changes lead to an increase in the total cost of 50%. Figures 5.3, 5.4 and 5.5 illustrate the impact of TTD on routing, staff wages and mobile medical units usage costs, correspondingly.

[8]https://www.gov.uk/drivers-hours/eu-rules

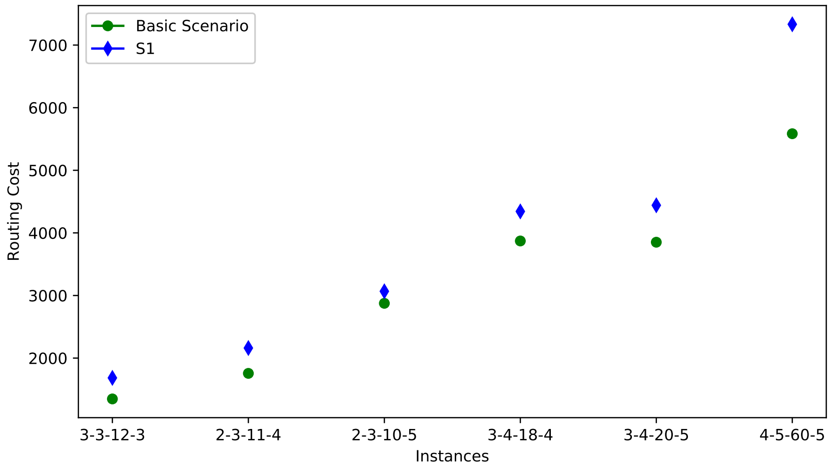

FIGURE 5.3: Changes on routing costs in the basic and S1 scenario

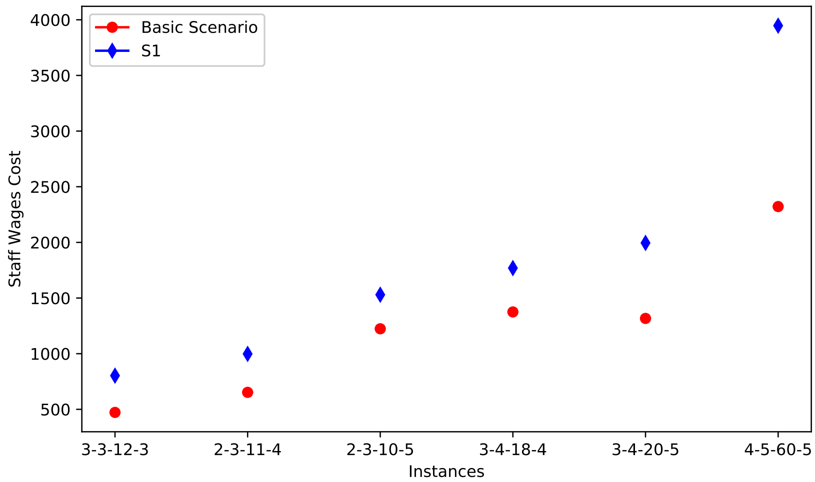

FIGURE 5.4: Changes on staff wages costs in the basic and S1 scenario

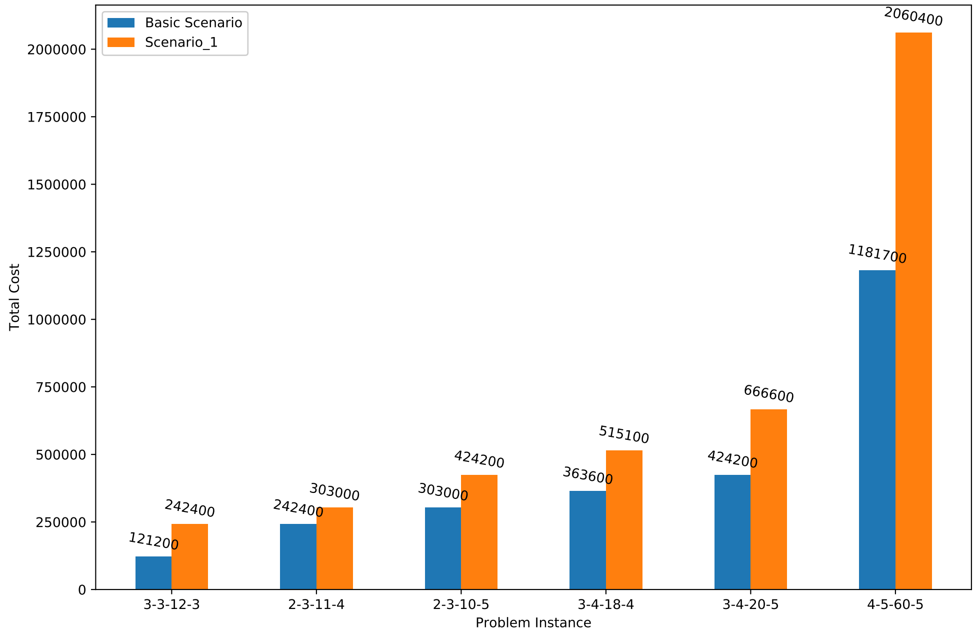

FIGURE 5.5: Changes on mobile medical units usage costs in the basic and S1 scenario

By setting TTD and STD at 9 hours, no changes are observed on the decisions related to the allocation to hospitals or the transportation of blood samples. However, a significant cost increase is reported on routing- and mobile medical units' usage-based decisions. Also, by imposing a stricter upper time limit, the selection of more and possible not such cost-efficient routes is decided comparing to the basic scenario. For example, the routes of problem instance "3-3-12-3" in both scenarios are (a route has the form manufacturing centre-local treatment facility/facilities-manufacturing centre):

Routes in the basic scenario for each time period:

- Period 1:
  - Route_1: $Mf3 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 7 \rightarrow 10 \rightarrow Mf3$ (Routing Time: 14h 4m 22s)
  - Route_2: $Mf3 \rightarrow 9 \rightarrow 11 \rightarrow 2 \rightarrow 8 \rightarrow 12 \rightarrow 6 \rightarrow Mf3$ (Routing Time: 14h 5m 27s)
- Period 2:
  - Route_1: $Mf3 \rightarrow 11 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow Mf3$ (Routing Time: 13h 44m 44s)
- Period 3:
  - Route_1: $Mf3 \rightarrow 10 \rightarrow 7 \rightarrow 6 \rightarrow 12 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow Mf3$ (Routing Time: 14h 33m 49s)

Routes in S1 for each time period:

- Period 1:
  - Route_1: $Mf3 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow Mf3$ (Routing Time: 7h 48m)
  - Route_2: $Mf3 \rightarrow 2 \rightarrow 11 \rightarrow 9 \rightarrow Mf3$ (Routing Time: 6h 32m 44s)
  - Route_3: $Mf3 \rightarrow 8 \rightarrow 12 \rightarrow 6 \rightarrow Mf3$ (Routing Time: 8h 7m 38s)
  - Route_4: $Mf3 \rightarrow 7 \rightarrow 10 \rightarrow Mf3$ (Routing Time: 5h 25m 5s)
- Period 2:
  - Route_1: $Mf3 \rightarrow 2 \rightarrow 4 \rightarrow 11 \rightarrow Mf3$ (Routing Time: 5h 44m 44s)
  - Route_2: $Mf3 \rightarrow 8 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow Mf3$ (Routing Time: 8h)
- Period 3:
  - Route_1: $Mf3 \rightarrow 10 \rightarrow 4 \rightarrow 2 \rightarrow Mf3$ (Routing Time: 5h 49m 5s)
  - Route_2: $Mf3 \rightarrow 8 \rightarrow 12 \rightarrow 6 \rightarrow 7 \rightarrow Mf3$ (Routing Time: 8h 57m 49s)

It is clear that the timing-related parameters significantly affects the number of routes as well as the selected mobile medical units.

Scenario_2 (S2). In this scenario both TTD and STD are set at 10 hours. Table 5.10 presents the differences between the best found total costs and the number of mobile medical units comparing with the basic scenario.

TABLE 5.10: Changes on the total costs and the number of mobile medical units in basic scenario and S2.

Instance	$BestTC_{basic}$	$\lvert MMU\rvert_{basic}$	$BestTC_{S2}$	$\lvert MMU\rvert_{S2}$
3-3-12-3	195,778.3	2	287,252.4	3
2-3-11-4	237,159.9	2	333,573.2	3
2-3-10-5	367,296.7	2	397,736.6	3
3-4-18-4	474,583	4	626,265.4	6
3-4-20-5	516,936.8	3	729,874.4	5
4-5-60-5	1,303,458	9	1,942,218	15

The results indicate that by imposing stricter time limits, more routes are selected thus resulting in the usage of more vehicles comparing to the basic scenario. Therefore, the total cost of the supply chain is increased. Figure 5.6 illustrates the differences of mobile medical units usage cost for the basic and S2 scenario.

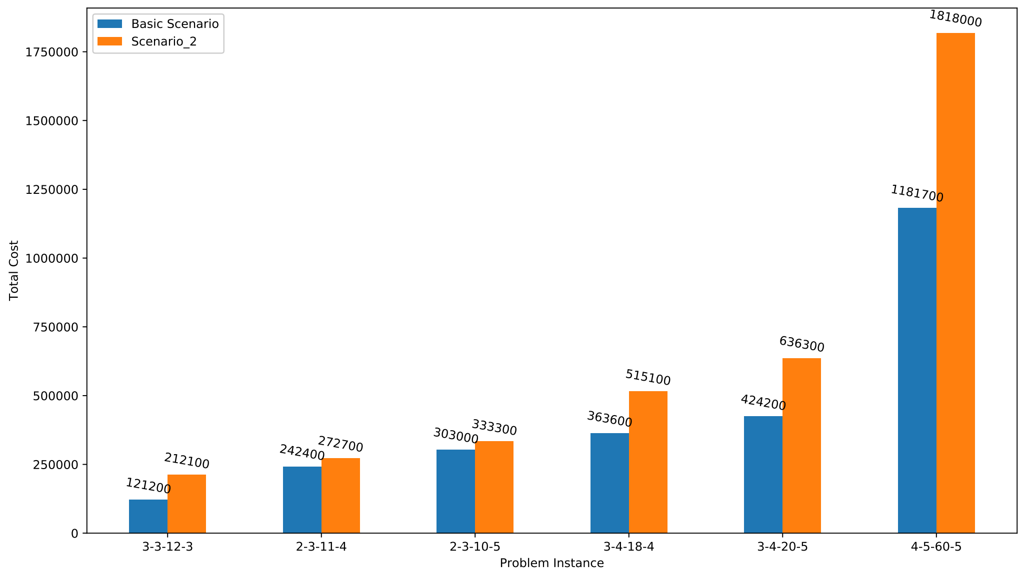

FIGURE 5.6: Changes on mobile medical units usage costs in S2

The routing cost is increased by 11.72%, while the staff wages' cost is 32.32% higher than the basic scenario.

Scenario_3 (S3). Here time limits are ignored. Table 5.11 summarizes the cost differences between S3 and the basic scenario. It also provides the number of selected mobile medical units in both cases.

TABLE 5.11: Changes on the total costs and the number of mobile medical units in basic scenario and S3.

Instance	$BestTC_{basic}$	$\|MMU\|_{basic}$	$BestTC_{S3}$	$\|MMU\|_{S3}$
3-3-12-3	195,778.3	2	195,788.3	2
2-3-11-4	237,159.9	2	302,868	2
2-3-10-5	367,296.7	2	367,296.7	2
3-4-18-4	474,583	4	352,306.6	2
3-4-20-5	516,936.8	3	395,415.3	2
4-5-60-5	1,303,458	9	421,924.2	2

The results indicate a significant reduction in the total cost compared to the basic scenario. More specifically, routing-related costs are approximately 11.3% lower than the basic scenario while a similar trend is noted for the staff wage cost. Furthermore, the mobile medical units usage cost is reduced by 42.5%.

Changes on the capacities and selection/usage cost of vehicles

The previous studies reveal the strong impact of key timing-related parameters on the fleet size and the selection of routes. Two other interdependent factors, that can potential affect the fleet size and routes, are the capacity and the cost of vehicles. The analysis is performed on the same problem instances used in S1, S2 and S3. Three scenarios are studied:

- Scenario_4: Both the capacity and cost are decreased by 10% and 15% correspondingly.
- Scenario_5: Both the capacity and cost are decreased by 25% and 50% correspondingly.
- Scenario_6: Both the capacity and cost are increased by 15% and 5% correspondingly.

It should be mentioned that the acquisition cost of a vehicle does not depend only on its capacity (Letmathe & Suares, 2017). There are additional factors, which can significantly affect the acquisition cost of a transport vehicle, such as exterior, interior and mechanical upgrades. For instance, a medium-roof transport van is more inexpensive than a high-roof vehicle. Also, a vehicle produced by a European manufacturer seems to be more expensive than a vehicle produced by an Asian manufacturer. Moreover, in our case the capacity does not refer to the actual capacity of the vehicle, but refers to the capacity of cold boxes in the case of transport vehicles and the capacity of refrigerator in the mobile units. Therefore, taking the *Scenario_5* as an example, it can represent a situation where smaller, but more efficient and expensive, cold boxes and refrigerators are supplied to the transport vehicles and the mobile medical units, respectively. Thus, the capacity of both vehicle types is significantly decreased. Also, the acquisition of the standard version of vehicles from an Asian supplier can lead to a major price decrease.

Scenario_4 (S4). In this scenario the capacity of vehicles is decreased by 10% while the cost is decreased by 15%. Table 5.12 presents the best found total costs and the number of mobile medical units in both scenarios.

TABLE 5.12: Changes on the total costs and the number of mobile medical units in basic scenario and S4.

Instance	$BestTC_{basic}$	$\|MMU\|_{basic}$	$BestTC_{S4}$	$\|MMU\|_{S4}$
3-3-12-3	195,778.3	2	193,533.7	3
2-3-11-4	237,159.9	2	259,922.6	2
2-3-10-5	367,296.7	2	315,246.7	2
3-4-18-4	474,583	4	380,998	3
3-4-20-5	516,936.8	3	443,517.8	3
4-5-60-5	1,303,458	9	1,142,201	10

This scenario leads to a reduction in the total costs compared to the basic scenario. It is interesting to note that the number of the mobile medical units are almost the same in both scenarios. Routing and staff wages' costs do not illustrate any significant improvement, as they are decreased by 0.85% and 1.9% respectively.

Scenario_5 (S5). This scenario studies the effects of decreasing both the capacity and cost of vehicles by 25% and 50% respectively. The total costs and number of selected mobile medical units are summarized in Table 5.13.

TABLE 5.13: Changes on the total costs and the number of mobile medical units in basic scenario and S5.

Instance	$BestTC_{basic}$	$\|MMU\|_{basic}$	$BestTC_{S5}$	$\|MMU\|_{S5}$
3-3-12-3	195,778.3	2	132,655.7	3
2-3-11-4	237,159.9	2	159,668	2
2-3-10-5	367,296.7	2	193,796.7	2
3-4-18-4	474,583	4	233,537.3	3
3-4-20-5	516,936.8	3	271,970.6	3
4-5-60-5	1,303,458	9	694,970.6	10

Results indicate a significant reduction of approximately 47% to the total cost compared to the basic scenario. However, the total cost is not a proper evaluation metric in this scenario, as it is obviously affected by the major vehicles usage cost decrease. Therefore, the potential impact of this scenario on routing and staff wages' costs should be also investigated. It is observed that both cost terms are slightly decreased compared to the basic scenario. To clarify these improvements, we focus on the routes built in the first period of problem instance "3-3-12-3" under the consideration of the basic and the current scenario.

Routes in the first period in the basic scenario.

- Route_1: $Mf3 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 7 \rightarrow 10 \rightarrow Mf3$ (Routing Time: $14h$ $4m$ $22s$)
- Route_2: $Mf3 \rightarrow 9 \rightarrow 11 \rightarrow 2 \rightarrow 8 \rightarrow 12 \rightarrow Mf3$ (Routing Time: $14h$ $5m$ $27s$)

Routes in the first period in S5.

- Route_1: $Mf3 \rightarrow 7 \rightarrow 3 \rightarrow 5 \rightarrow 1 \rightarrow Mf3$ (Routing Time: $9h$ $53m$ $27s$)
- Route_2: $Mf3 \rightarrow 10 \rightarrow 6 \rightarrow 12 \rightarrow 8 \rightarrow 2 \rightarrow 6 \rightarrow Mf3$ (Routing Time: $12h$ $24m$)
- Route_3: $Mf3 \rightarrow 11 \rightarrow 9 \rightarrow Mf3$ (Routing Time: $4h$ $43m$ $38s$)

The routing time is related to the distance, as the speed is assumed constant in the proposed approach. It is clear that the decreased capacity of vehicles force the algorithm to construct more and shorter routes comparing to the basic scenario. For example, the total routing time in the first period is decreased by almost one hour.

Scenario_6 (S6). The impact, of increasing the capacity of vehicles by 15% and the cost of vehicles by 5% on costs, is investigated in this scenario. Table 5.14 provides the best total costs and the number of mobile medical units, as reported for each problem instance under the basic and the current scenarios.

TABLE 5.14: Changes on the total costs and the number of mobile medical units in basic scenario and S6.

Instance	$BestTC_{basic}$	$\lvert MMU\rvert_{basic}$	$BestTC_{S6}$	$\lvert MMU\rvert_{S6}$
3-3-12-3	195,778.3	2	205,141.3	2
2-3-11-4	237,159.9	2	317,192.6	2
2-3-10-5	367,296.7	2	384,634.7	2
3-4-18-4	474,583	4	465,237.3	3
3-4-20-5	516,936.8	3	541,493.4	3
4-5-60-5	1,303,458	9	1,397,790	9

Solutions under S6 lead to a 4.76% increase of the total cost compared to the basic scenario. This is due to increases in the vehicles usage cost. However, this may be attributed to the simultaneous increase of vehicles' capacity. This increase provides the opportunity to build less routes and to potentially achieve some cost savings. For instance, in problem case "3-4-18-4" the four routes in the first period of the basic scenario are combined into three routes in the same period of the current scenario. More specifically:

Routes in the first period in the basic scenario.

- Route_1: $Mf3 \rightarrow 17 \rightarrow 18 \rightarrow 15 \rightarrow Mf3$ (Routing Time: 10h 49m 5s)
- Route_2: $Mf3 \rightarrow 5 \rightarrow 8 \rightarrow 10 \rightarrow 4 \rightarrow 7 \rightarrow 1 \rightarrow Mf3$ (Routing Time: 14h 26m 11s)
- Route_3: $Mf3 \rightarrow 16 \rightarrow 9 \rightarrow 11 \rightarrow Mf3$ (Routing Time: 5h 51m 16s)
- Route_4: $Mf3 \rightarrow 2 \rightarrow 6 \rightarrow 12 \rightarrow 13 \rightarrow Mf3$ (Routing Time: 9h 18m 33s)

Routes in the first period in the current scenario.

- Route_1: $Mf3 \rightarrow 17 \rightarrow 18 \rightarrow 15 \rightarrow Mf3$ (Routing Time: 10h 49m 5s)
- Route_2: $Mf3 \rightarrow 5 \rightarrow 8 \rightarrow 10 \rightarrow 4 \rightarrow 7 \rightarrow 1 \rightarrow Mf3$ (Routing Time: 14h 26m 11s)
- Route_3: $Mf3 \rightarrow 16 \rightarrow 9 \rightarrow 13 \rightarrow 12 \rightarrow 2 \rightarrow 6 \rightarrow 11 \rightarrow Mf3$ (Routing Time: 14h 54m 33s)

Changes on the capacities and opening cost of facilities

The capacity and cost of facilities significantly affect the structure of the supply chain network. Moreover, these two parameters often illustrate significant fluctuations (Govindan et al., 2017). Two scenarios are considered here. In Scenario_7 (S7) both the capacity and cost are decreased by 20% and 15% correspondingly, while in Scenario_8 (S8) the capacity and cost are decreased by 25% and 5% correspondingly. A potential reason of imposing such simultaneous reductions is the decision to build smaller facilities in order to reduce their costs. For existing facilities this can be attributed to the concept of shared facilities (Assid et al., 2019). For these scenarios, the following seven problem instances are considered: "3-3-12-3", "2-3-10-5", "3-4-18-4", "3-4-20-5", "4-5-60-5", "6-15-130-7" and "10-20-200-7". Table 5.15 provides the best found total cost and the number of opened manufacturing centres for S7, S8 and the basic scenario.

TABLE 5.15: Total costs and the number of manufacturing centres in the basic scenario, S7 and S8.

Instance	$BestTC_{basic}$	$\|J\|_{basic}$	$BestTC_{S7}$	$\|J\|_{S7}$	$BestTC_{S8}$	$\|J\|_{S8}$
3-3-12-3	195,778.3	1	237,549.5	2	238,235.5	2
2-3-10-5	367,296.7	2	365,710.4	2	366,857.8	2
3-4-18-4	474,583	2	411,695.6	2	443,175.6	2
3-4-20-5	516,936.8	2	605,260.1	3	599,140.5	3
4-5-60-5	1,303,458	2	1,303,435	2	1,334,836	2
6-15-130-7	3,657,759	2	3,806,458	3	3,717,070	3
10-20-200-7	5,386,204	2	5,349,408	2	5,424,733	3

The opened manufacturing centres and the selected specialized hospitals are summarized in Table 5.16.

TABLE 5.16: Opened manufacturing centres and selected hospitals in the basic scenario, Scenario_7 and Scenario_8.

Instance	MF_{basic}	H_{basic}	Mf_{S7}	H_{S7}	Mf_{S8}	H_{S8}
3-3-12-3	3	1, 3	2, 3	2, 3	2, 3	2, 3
2-3-10-5	1, 3	1, 2	1, 3	1, 2	1, 3	1, 2
3-4-18-4	3, 4	1, 2, 3	3, 4	1, 2, 3	3, 4	1, 2, 3
3-4-20-5	2, 4	2, 3	2, 3, 4	1, 2, 3	2, 3, 4	1, 2, 3
4-5-60-5	1, 3	2, 3, 4	1, 3	2, 3, 4	1, 3	2, 3, 4
6-15-130-7	1, 14	4, 6	1, 4, 15	2, 4, 6	1, 4, 15	2, 4, 6
10-20-200-7	5, 17	2, 5	5, 17	2, 5, 7	5, 14, 17	2, 5, 6

The imposed changes on the capacity do not seem to affect the number of opened hospitals. However, significant changes are observed on the number of selected manufacturing centres, especially when the problem size is increased and the capacity of manufacturing centres is decreased enough (more than 20%). Furthermore, minor changes are reported between S7 and S8 regarding the opened manufacturing centres and selected hospitals. Problem instance "10-20-200-7" is the only exception.

The decrease of facilities capacity leads to opening more manufacturing centres and an associated increase to the location-related costs. However, the total cost does not significantly change. This is mainly due to a balancing between location and routing costs. More specifically, the inescapable opening of more manufacturing centres provides the opportunity for the selection of more efficient routes, which lead to a significant routing cost reduction. This can be clearly observed in Figure 5.7 for problem instance "6-15-130-7".

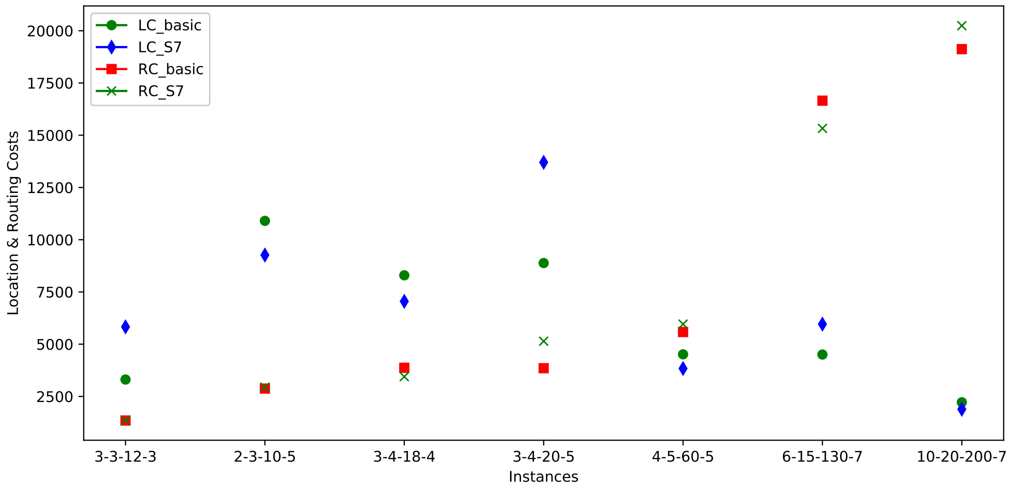

FIGURE 5.7: Differences on location and routing costs between the basic scenario and S7

5.6 Concluding remarks

This chapter presents a new decentralized CAR T-cell therapies' administration process, which addressed by a specific supply chain network representation to tackle the rising challenges associate with the design and operation of CAR T-cell therapies supply chains. This is one of the first attempts to introduce a novel network structure that takes into account key design and operational decisions of this new supply chain problem. The key point of the proposed network structure is that the administration of therapies is performed not in specialized hospitals but in local treatment facilities, which are located close to the patients' sites. More specifically, this representation considers the selection of the optimal number of specialized hospitals and capacitated manufacturing centres. Each local treatment facility is assigned to a specific hospital and a selected manufacturing centre. Therefore, proper fleet size of transport vehicles is decided to guarantee the transportation of the required samples from specialized hospitals to selected manufacturing centres, where the final therapies are produced. To this end, mobile medical units are selected in order to deliver therapies and transport specialized medical staff to the local treatment facilities. The time needed for completing a scheduled route must not exceed a specified time limit defined by the time-sensitive nature of the cellular therapy. A new MILP model is presented along with a metaheuristic VNS-based method for the efficient solution of large problem instances of practical interest. Extensive numerical analyses provide useful insights into the key factors affecting the design and operational planning of these supply chains.

Chapter 6

Conclusions & future directions

6.1 Overview

This thesis focuses on the modeling and optimization of complex and large-scale SCN problems using efficient metaheuristic solution techniques. Problems addressed via the proposed approach consider simultaneously strategic, tactical and operational decisions. This chapter summarizes the key contributions of the thesis and identifies potential future research directions.

6.2 Summary of key scientific contributions

This thesis has studied four new complex SCN optimization problems and introduces efficient VNS-based solution techniques.

In Chapter 2, a generalization of the well-known LIRP has been addressed. The new problem variant, called LIRPDO, considers further strategic decisions such as the selection of the proper vehicles' provider. An MIP has been developed to model the problem. Due to the high computational complexity of the introduced problem, a hybrid GVNS-based solution method is proposed. More specifically, a GVNS algorithm using the pVND as its core improvement method, combined with an inventory rescheduling post-optimization procedure has been developed. Extensive numerical analyses on newly proposed benchmark instances indicate the efficiency of the proposed solution approach. Several LIRP benchmarks from the open literature have been solved and extensively analysed. The proposed algorithm outperforms the state-of-the-art method, known as SA-Hyb-ILRP (Zhang et al., 2014). Its predominance is especially evident on the solution of large-sized problem instances. The key strength of the GVNS-InvRP algorithm is its ability to open the minimum required number of depots. The main features of this chapter are summarized as follows:

◇ An MIP model for the underlying supply chain network problem.

◇ A ratio-based locations' selection strategy introduced in the first phase of a two phase construction heuristic.

◇ Application of a GVNS-based solution approach on 20 random generated LIRPDO instances and comparison with the CPLEX solver. The proposed method is also applied on 20 LIRP benchmark instances from the literature.

◇ The proposed approach is a self-contained solver.

◇ A new benchmark set with the current largest instances of LIRP in the open literature have been generated and made publicly available.

The PLIRP, a green variant of the LIRP, has been introduced in the third chapter of this dissertation. PLIRP considers not only economic aspects of supply chain activities, but also their environmental dimension. One of the most accurate fuel consumption models was adopted from the literature in order to compute both the fuel consumption and the CO_2 emissions. An MIP formulation of the problem has been proposed along with three VNS-based heuristic algorithms and their corresponding adaptive variants. The effects of using different shaking strength parameter values on the efficiency of the proposed algorithms have been thoroughly investigated. Furthermore, computational analysis on the impact of using the SSP either after each local search operator or in the end of each VND iteration is conducted. The solution schemes using the SPP after each local search operator have been proved as the most efficient ones. A hybridization of the $AGVNS_{pVND}$ for the solution of both small and medium problem cases and the $GVNS_{pVND}$ for solving large problem cases has increased the computational efficiency. As holding costs are critical, a sensitivity analysis has also been performed to assess the potential impact of the variations of holding costs on the total cost. Despite the fact that holding costs significantly affect the total cost of the supply chain system, the use of the flexible replenishment policy keeps this cost increase in relatively low levels. Some exceptions have been noticed by using more vehicles. The main contributions highlighted in this chapter are briefly summarized as follows:

◇ A new complex logistics optimization problem with environmental considerations.

- ◇ An MIP model by integrating and extending two models from the literature.
- ◇ Three VNS-based solution algorithms.
- ◇ A new benchmark set with the current largest instances of the Pollution LIRP (PLIRP) reported in the open literature have been generated and made publicly available.
- ◇ The impact of the flexible replenishment policy on building better routing patterns is illustrated.
- ◇ A sensitivity analysis is performed to illustrate the effect of holding costs on the total supply chain cost for several problem instances.

In Chapter 4, a new integrated supply chain problem is studied. This new variant of the LIRP, called FSMPLIRP, considers fleet composition and capacity planning decisions while adopting the well-known JiT replenishment policy. For the efficient solution of this problem, GVNS-based solution methods have been investigated. A critical performance component of a VNS-based algorithm is the shaking mechanism. Herein, new adaptive shaking strategies have been proposed as essential intelligent learning components of the solution methods. The main difference between the proposed shaking schemes and the commonly used shaking method is that the new methods rely on past experience to guide the search in more promising directions into the search space. The shaking operators are initially ordered either based on their complexity or in a random fashion. In each next iteration the shaking operators are re-ordered according to the number of achieved improvements by using each of them in the previous iteration. Extensive computational tests conducted on a new set of benchmark instances, indicate the efficiency of the GVNS schemes using the new adaptive shaking strategies. Results demonstrate the economic and environmental benefits of using a mixed fleet. Furthermore, the application of an alternative initialization rule has been investigated. It was observed that further improvements can be achieved, especially on the solution of large problem cases. A synopsis of the key contributions, included in this chapter, is following:

- ◇ An MIP formulation for the new complex SCN optimization problem.
- ◇ New adaptive shaking methods, as intelligent components in the developed GVNS-based algorithms for the solution of the above problem.

◇ Investigation of different variants of the solution approaches.

◇ Development of a self-contained solver for the problem under consideration.

◇ Useful managerial insights are derived for a number of complex problem instances.

An integrated CAR T-cell SCN optimization problem has been studied in Chapter 5. A novel SCN representation is proposed in an effort to efficiently manage the increased demand and avoid potential bottlenecks in the therapy manufacturing and distribution processes. To this end, specialized practitioners-manned mobile medical units are considered to visit local treatment facilities, located close to patients' sites. The administration of the therapy is performed in these mobile clinics. After the administration, patients will remain in the local treatment facility to be monitored for potential adverse reactions. To model this problem, a new MILP model has been developed. However, the solution of problems with potential practical interest cannot be solved with exact algorithms. Therefore, a VNS-based heuristic solution method has been developed for solving large problem cases efficiently. Several problems have been solved and the effect of key parameters on strategic and operational decisions has been studied. The key contributions presented in this chapter are the following:

◇ A new MIP formulation.

◇ A novel SCN representation.

◇ The consideration of mobile medical units.

◇ Development of a self-contained GVNS-based solver for the problem under consideration.

◇ Extended computational and sensitivity analysis.

6.3 Future research directions

A range of issues requiring further investigation have been revealed in the course of this work. These issues are divided into two main classes. In particular,

- ***Solution methods***:
  - ***Improvements***. Due to the high computational complexity of the underlying problems, heuristic solution methods have been developed for their efficient solution. However, these techniques cannot guarantee optimality of the reported solutions. Therefore, the development of new local search operators, both in the improvement and shaking phases, may lead to improved solutions. Moreover, the combined use of adaptive improvements and shaking mechanisms can positively affect the exploration of the research space. A parallel implementation of the proposed algorithms can be adopted to accelerate the overall solution process (Antoniadis & Sifaleras, 2017). Furthermore, the development of hybrid approaches which combine direct solution approaches (e.g. branch and bound) with the proposed metaheuristics can potentially lead to better solutions.
  - ***Evaluation***. The design and development of new effective approximate solution algorithms, such as Lagrangian relaxation methods, will provide better lower bounds. Thus, a more accurate performance evaluation over the proposed heuristic solution methods can be achieved.
- ***Problem extensions***. Several generalizations of the proposed problems can be addressed by considering additional realistic features. More specifically:
  - Multiple products (Zhalechian et al., 2016). This feature denotes a supply chain system which handles multiple types of commodities.
  - Stochastic demand (Nenes et al., 2010; Rafie-Majd et al., 2018). In the SCN problems the demand of customers/patients is time-varying but deterministic. A more realistic approach should consider stochastic product demands.

– Closed-loop SCN (Panagiotidou et al., 2017; Zikopoulos & Tagaras, 2015). A key sustainable supply chain approach is the consideration of closed-loop supply chains, which combine traditional or forward logistics with the reverse flow of no functional or needed products, due to limited resources.

– Intermediate stops for refueling or charging purposes (Hof et al., 2017),

– Time windows (Marinakis et al., 2019). A time range is defined for each customer in which he must be served. Time windows are divided into soft, strict or hard and mixed. Soft time windows can be violated by considering penalty costs, while hard time windows must be respected.

– Alternative shipping strategies (Nikolopoulou et al., 2017). These strategies includes direct shipment, cross-docking and transloading. More specifically, direct shipment refers to the delivery of products from a source node to a consumer directly. In cross-docking shipment, products are unloaded from inbound delivery trucks and they are directly loaded onto outbound trucks. Transloading refers to en route uploading and re-loading of products between different modes of transportation.

– Detailed scheduling decisions in the manufacturing centres in CAR T-cell therapies SCN, such as capacity and resource planning decisions (Papathanasiou et al., 2020).

Appendix A

Research Outputs

Herein, an overview of the research outputs of this dissertation is provided.

Peer-reviewed journal publications:

1. Karakostas, P., Sifaleras, A. and Georgiadis, M.C. 2019. *A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing*, Computers & Chemical Engineering, 126, pp. 263-279, doi: 10.1016/j.compchemeng.2019.04.015 (Karakostas et al., 2019b)

2. Karakostas, P., Sifaleras, A. and Georgiadis, M.C. 2020. *Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory-routing problem*, Expert Systems with Applications, 153, 113444, doi: 10.1016/j.eswa.2020.113444 (Karakostas et al., 2020c)

3. Karakostas, P., Panoskaltsis, N., Mantalaris, A. and Georgiadis, M.C., 2020. *Optimization of CAR T-cell therapies Supply Chains*, Computers & Chemical Engineering, 139, 106913, doi: 10.1016/j.compchemeng.2020.106913 (Karakostas et al., 2020a)

4. Karakostas, P., Sifaleras, A. and Georgiadis, M.C. 2020. *Variable neighborhood search-based solution methods for the pollution location-inventory-routing problem*. Submitted to Optimization Letters on March, 23.

International conference proceedings:

1. Karakostas P., Sifaleras A., Georgiadis M.C. (2019) Basic VNS Algorithms for Solving the Pollution Location Inventory Routing Problem. In: Sifaleras A., Salhi S., Brimberg J. (eds) Variable Neighborhood Search. ICVNS 2018. Lecture Notes in Computer Science, vol 11328. Springer, Cham (Karakostas et al., 2019a)

2. Karakostas P., Sifaleras A., Georgiadis M.C. (2020) Adaptive GVNS Heuristics for Solving the Pollution Location Inventory Routing Problem. In: Matsatsinis N., Marinakis Y., Pardalos P. (eds) Learning and Intelligent Optimization. LION 2019. Lecture Notes in Computer Science, vol 11968. Springer, Cham (Karakostas et al., 2020b)

National conferences:

- Karakostas P., Sifaleras A., Georgiadis M.C. (2017) General Variable Neighborhood Search for the Efficient Solution of Location Inventory Routing Problems. OR in the digital era - ICT challenges, 6th International Symposium and 28th National Conference on Operational Research, Thessaloniki, Greece, June 8-10.

Bibliography

Aguirre, A., Liu, S., & Papageorgiou, L. (2018). Optimisation approaches for supply chain planning and scheduling under demand uncertainty. *Chemical Engineering Research and Design*, *138*, 341–357.

Aldis, J. (2017). Strategic facility location, capacity acquisition, and technology choice decisions under demand uncertainty: Robust vs. non-robust optimization approaches. *European Journal of Operational Research*, *260*, 1095–1104.

Alfares, H., & Ghaithan, A. (2016). Inventory and pricing model with price-dependent demand, time-varying holding cost, and quantity discounts. *Computers & Industrial Engineering*, *94*, 170–177.

Almouhanna, A., Quintero-Araujo, C., Panadero, J., Juan, A., Khosravi, B., & Ouelhadj, D. (2020). The location routing problem using electric vehicles with constrained distance. *Computers & Operations Research*, *115*. https://doi.org/10.1016/j.cor.2019.104864

Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., & Morabito, R. (2020). Formulations, branch-and-cut and a hybrid heuristic algorithm for an inventory routing problem with perishable products. *European Journal of Operational Research*, *283*, 511–529.

Ambrosino, D., & Scutella, M. (2005). Distribution network design: New problems and related models. *European Journal of Operational Research*, *105*, 815–834.

Amiri, M., Amin, S., & Tavakkoli-Moghaddam, R. (2019). A Lagrangean decomposition approach for a novel two-echelon node-based location-routing problem in an offshore oil and gas supply chain. *Transportation Research Part E: Logistics and Transportation Review*, *128*, 96–114.

Amiri-Aref, M., Klibi, W., & Babai, M. (2018). The multi-sourcing location inventory problem with stochastic demand. *European Journal of Operational Research*, *266*, 72–87.

Antoniadis, N., & Sifaleras, A. (2017). A hybrid CPU-GPU parallelization scheme of variable neighborhood search for inventory optimization problems. *Electronic Notes in Discrete Mathematics*, *58*, 47–54.

Araya-Sassi, C., Paredes-Belmar, G., & Gutiérrez-Jarpa, G. (2020). Multi-commodity inventory-location problem with two different review inventory control policies and modular stochastic capacity constraints. *Computers & Industrial Engineering*, *143*. https://doi.org/10.1016/j.cie.2020.106410

Archetti, C., Coelho, L., & Speranza, M. (2019). An exact algorithm for the inventory routing problem with logistic ratio. *Transportation Research Part E: Logistics and Transportation Review*, *131*, 96–107.

Asadi, E., Habibi, F., Nickel, S., & Sahebi, H. (2018). A bi-objective stochastic location-inventory-routing model for microalgae-based biofuel supply chain. *Applied Energy*, *228*, 2235–2261.

Assid, M., Gharbi, A., & Hajji, A. (2019). Production control of failure-prone manufacturing-remanufacturing systems using mixed dedicated and shared facilities. *International Journal of Production Economics*. https://doi.org/10.1016/j.ijpe.2019.107549

Baboo, J., Kilbride, P., Delahaye, M., Milne, S., Fonseca, F., Blanco, M., Meneghel, J., Nancekievill, A., Gaddum, N., & Morris, J. (2019). The Impact of Varying Cooling and Thawing Rates on the Quality of Cryopreserved Human Peripheral Blood T Cells. *Scientific Reports*, *9*, 3417.

Barbosa-Póvoa, A., da Silva, C., & Carvalho, A. (2018). Opportunities and challenges in sustainable supply chain: An operations research perspective. *European Journal of Operational Research*, *268*, 399–431.

Barbosa-Povoa, A., & J.M., P. (2019). Process Supply Chains: Perspectives from Academia and Industry. *Computers & Chemical Engineering*, *132*. https://doi.org/10.1016/j.compchemeng.2019.106606

Barbosa-Povoa, A., & Pinto, J. (2018). Opportunities and challenges in sustainable supply chain: An operations research perspective. *European Journal of Operational Research*, *268*, 399–431.

Baron, O., Berman, O., Fazel-Zarandi, M., & Roshanaei, V. (2019). Almost Robust Discrete Optimization. *European Journal of Operational Research*, *276*, 451–465.

Barth, M., Younglove, T., & Scora, G. (2005). Development of a heavy-duty diesel modal emissions and fuel consumption model. technical report. *UC Berkeley: California Partners for Advanced Transit and Highways, San Francisco*.

Basligil, H., Kara, S., Alcan, P., Ozkan, B., & Caglar, E. (2011). A distribution network optimization problem for third party logistics service providers. *Expert Systems with Applications*, *38*, 12730–12738.

Bektaş, T., Ehmke, J., Psaraftis, H., & Puchinger, J. (2019). The role of operational research in green freight transportation. *European Journal of Operational Research*, *274*, 807–823.

Bektaş, T., & Gouveia, L. (2014). Requiem for the miller–tucker–zemlin subtour elimination constraints? *European Journal of Operational Research*, *236*, 820–832.

Bener, A., Yildirim, E., Özkan, T., & Lajunen, T. (2017). Driver sleepiness, fatigue, careless behavior and risk of motor vehicle crash and injury: Population based case and control study. *Journal of Traffic and Transportation Engineering (English Edition)*, *4*, 496–502.

Bertazzi, L., Coelho, L., De Maio, A., & Laganá. (2019). A matheuristic algorithm for the multi-depot inventory routing problem. *Transportation Research Part E: Logistics and Transportation Review*, *122*, 524–544.

Biuki, M., Kazemi, A., & Alinezhad, A. (2020). An integrated location-routing-inventory model for sustainable design of a perishable products supply chain network. *Journal of Cleaner Production*, *260*. https://doi.org/10.1016/j.jclepro.2020.120842

Blocho, M. (2020). Heuristics, metaheuristics, and hyperheuristics for rich vehicle routing problems. In *Smart Delivery Systems: Solving Complex Vehicle Routing Problems* (pp. 101–156). Elsevier. https://doi.org/10.1016/C2017-0-03660-1

Boyiadzis, M., Dhodapkar, M., Brentjens, R., James N. Kochenderfer, J., Neelapu, S., Maus, M., Porter, D., Maloney, D., Grupp, S., Mackall, C., June, C., & Bishop, M. (2018). Chimeric antigen receptor (CAR) T therapies for the treatment of hematologic malignancies: Clinical perspective and significance. *Journal for ImmunoTherapy of Cancer*, *6*. https://doi.org/10.1186/s40425-018-0460-5

Brimberg, J., Mladenović, N., & Todosijević, R. (2017). A basic variable neighborhood search heuristic for the uncapacitated multiple allocation p-hub center problem. *Optimization Letters*, *11*, 313–327.

Brooke, A., Kendrik, D., Meeraus, A., Raman, R., & Rosenthal, R. (1998). *GAMS-A User's Guide*.

Chao, C., Zhihui, T., & Baozhen, Y. (2019). Optimization of two-stage location–routing–inventory problem with time-windows in food distribution network. *Journal of Cleaner Production*, *273*, 111–134. https://doi.org/10.1007/s10479-017-2514-3

Chen, D., Sun, G., & Liu, G. (2014). Combined location routing and inventory problem of e-commerce distribution system with fuzzy random demand. *International Journal of Hybrid Information Technology*, *7*, 429–442.

Cheng, C., Yang, P., Qi, M., & Rousseau, L. (2017). Modeling a green inventory routing problem with a heterogeneous fleet. *Transportation Research Part E*, *97*, 97–112.

Costa, L., Aloise, D., & Mladenović, N. (2017). Less is more: Basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering. *Information Sciences*, *415–416*, 247–253.

Cuda, R., Guastaroba, G., & Speranza, M. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, *55*, 185–199.

Dai, Z., Gao, K., & Giri, B. (2020). A hybrid heuristic algorithm for cyclic inventory-routing problem with perishable products in VMI supply chain. *Expert Systems with Applications*, *153*. https://doi.org/10.1016/j.eswa.2020.113322

Darvish, M., Archetti, C., Coelho, L., & Speranza, M. (2019). Flexible two-echelon location routing problem. *European Journal of Operational Research*, *277*, 1124–1136.

Della Croce, F. (2014). Mixed Integer Linear Programming Models for Combinatorial Optimization Problems. In V. Paschos (Ed.), *Concepts of combinatorial optimization*. https://doi.org/10.1002/9781119005216.ch5

Derbel, H., Jarboui, B., & Bhiri, R. (2019). A skewed general variable neighborhood search algorithm with fixed threshold for the heterogeneous fleet vehicle routing problem. *Annals of Operations Research*, *272*, 243–272. https://doi.org/10.1007/s10479-017-2576-2

Deroussi, L. (2016). *Metaheuristics for Logistics*. ISTE; Wiley.

Dong, Y., Maravelias, C., Pinto, J., & Sundaramoorthy, A. (2017). Solution methods for vehicle-based inventory routing problems. *Computers & Chemical Engineering*, *101*, 259–278.

Drexl, M., & Schneider, M. (2015). A survey of variants and extensions of the location-routing problems. *European Journal of Operational Research*, *241*, 283–308.

Dukkanci, O., Kara, B. Y., & Bektaş, T. (2019). The green location-routing problem. *Computers & Operations Research*, *105*, 187–202.

Eskandarpour, M., Dejax, P., & Peton, O. (2017). A large neighborhood search heuristic for supply chain network design. *Computers & Operations Research*, *80*, 23–37.

Farahani, R. Z., Bajgan, H. R., Fahimnia, B., & Kaviani, M. (2015). Location-inventory problem in supply chains: A modelling review. *International Journal of Production Research*, *53*, 3769–3788.

Festa, P. (2014). A brief introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems, In *2014 16th International Conference on Transparent Optical Networks (ICTON)*.

Flood, M. (1956). The Traveling-Salesman Problem. *Operations Research*, *4*, 61–75.

Foo, P., Lee, V., Tan, G., & Ooi, K. (2018). A gateway to realising sustainability performance via green supply chain management practices: A PLS-ANN approach. *Expert Systems with Applications*, *107*, 1–14.

Forouzanfar, F., Tavakkoli-Moghaddam, R., Bashiri, M., Baboli, A., & Hadji Molana, S. (2018). New mathematical modeling for a location–routing–inventory problem in a multi-period closed-loop supply chain in a car industry. *Journal of Industrial Engineering International*, *14*, 537–553.

Franco, C., & Alfonso-Lizarazo, E. (2019). Optimization under uncertainty of the pharmaceutical supply chain in hospitals. *Computers & Chemical Engineering*. https://doi.org/10.1016/j.compchemeng.2019.106689

Fritz, M. M. C. (2019). Sustainable Supply Chain Management. In W. Leal Filho, A. Azul, L. Brandli, P. Özuyar, & T. Wall (Eds.), *Responsible Consumption and Production. Encyclopedia of the UN Sustainable Development Goals*. Springer.

Gardner, J., & Webster, A. (2016). The social management of biomedical novelty: Facilitating translation in regenerative medicine. *Social Science & Medicine*, *156*, 90–97.

Gharaei, A., Karimi, M., & Hoseini Shekarabi, S. (2019). An integrated multi-product, multi-buyer supply chain under penalty, green, and quality control polices and a vendor managed inventory with consignment stock agreement: The outer approximation with equality relaxation and augmented penalty algorithm. *Applied Mathematical Modelling*, *69*, 223–254.

Ghorbani, A., & Akbari Jokar, M. (2016). A hybrid imperialist competitive-simulated annealing algorithm for a multisource multi-product location-routing-inventory problem. *Computers & Industrial Engineering*, *101*, 116–127.

Glover, F., Gutin, G., Yeo, A., & Zverovich, A. (2001). Construction heuristics for the asymmetric TSP. *European Journal of Operational Research*, *129*, 555–568.

Goerigk, M., Kasperski, A., & Zielinski, P. (2020). Two-stage combinatorial optimization problems under risk. *Theoretical Computer Science*, *804*, 29–45.

Gonul Kochan, C., Nowicki, D., Sauser, B., & Randall, W. (2018). Impact of cloud-based information sharing on hospital supply chain performance: A system dynamics framework. *International Journal of Production Economics*, *195*, 168–185.

Govindan, K., Azevedo, S., Carvalho, H., & Cruz-Machado, V. (2014). Impact of supply chain management practices on sustainability. *Journal of Cleaner Production*, *85*, 212–225.

Govindan, K., Fattahi, M., & Keyvanshokooh, E. (2017). Supply chain network design under uncertainty: A comprehensive review and future research directions. *European Journal of Operational Research*, *263*, 108–141.

Guerrero, W., C., P., Velasco, N., & Amaya, C. (2013). Hybrid heuristic for the inventory location-routing problem with deterministic demand. *International Journal of Production Economics*, *146*, 359–370.

Guo, H., Congdong, L., Zhang, Y., Zhang, C., & Wang, Y. (2018). A Nonlinear Integer Programming Model for Integrated Location, Inventory, and Routing Decisions in a Closed-Loop Supply Chain. *Complexity*. https://doi.org/10.1155/2018/2726070

Habibi, F., Asadi, E., & Sadjadi, S. (2018). A location-inventory-routing optimization model for cost effective design of microalgae biofuel distribution system: A case study in Iran. *Energy Strategy Reviews*, *22*, 82–03.

Haghjoo, N., Tavakkoli-Moghaddam, R., Shahmoradi-Moghadam, H., & Rahimi, Y. (2020). Reliable blood supply chain network design with facility disruption: A real-world application. *Engineering Applications of Artificial Intelligence*, *90*.

Hansen, P., & Mladenović. (2018). Variable neighborhood search. In R. Martí, P. Pardalos, & M. Resende (Eds.), *Handbook of Heuristics*. Springer, Cham.

Hansen, P., & Mladenović, N. (2003). Variable Neighborhood Search. In F. Glover & G. Kochenberger (Eds.), *Handbook of Metaheuristics* (pp. 145–184). Boston, International Series in Operations Research & Management Science.

Hansen, P., & Mladenović, N. (2014). Variable Neighborhood Search. In E. Burke & G. Kendall (Eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (pp. 313–337). New York, Springer Science+Business Media.

Hansen, P., Mladenović, N., Brimberg, J., & Pérez, J. (2010). Variable Neighborhood Search. In M. Gendreau & J. Potvin (Eds.), *Handbook of Metaheuristics* (pp. 61–86). Boston, International Series in Operations Research & Management Science.

Hansen, P., Mladenović, N., Todosijevic, R., & Hanafi, S. (2017). Variable neighborhood search: Basics and variants. *EURO Journal on Computational Optimization*, *5*, 423–454.

Harrison, R., Zylberberg, E., Ellison, S., & Levine, B. (2019). Chimeric antigen receptor–T cell therapy manufacturing: Modelling the effect of offshore production on aggregate cost of goods. *Cytotherapy*, *21*, 224–233.

Hiassat, A., Diabat, A., & Rahwan, I. (2017). A genetic algorithm approach for location-inventory-routing problem with perishable products. *Journal of Manufacturing Systems*, *42*, 93–103.

Hjaila, K., Lainez-Aguirre, J., Zamarripa, M., Puigjaner, L., & Espuna, A. (2016). Optimal integration of thirdparties in a coordinated supply chain management environment. *Computers & Chemical Engineering*, *86*, 48–61.

Hof, J., Schneider, M., & Goeke, D. (2017). Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops. *Transportation Research Part B: Methodological*, *97*, 102–112.

Holder, M., Devpura, A., Lee, A., & Chandran, S. (2019). Aligning Data Analytics and Supply Chain Strategy in the Biopharmaceutical Industry. In M. Anandarajan & T. Harrison (Eds.), *Aligning Business Strategies and Analytics. Advances in Analytics and Data Science*. https://doi.org/10.1007/978-3-319-93299-6_5

Hong, J., Zhang, Y., & Ding, M. (2018). Sustainable supply chain management practices, supply chain dynamic capabilities and enterprise performance. *Journal of Cleaner Production*, *172*, 3508–3519.

Hoos, H., & Stützle, T. (2005). Fundamentals of algorithms. In *Stochastic Local Search: Foundations and Applications* (pp. 13–59). Elsevier. https://doi.org/10.1016/B978-155860872-6/50018-4

Hristu-Varsakelis, D., Karagianni, S., Pempetzoglou, M., & Sfetsos, A. (2012). Optimizing production in the greek economy: Exploring the interaction between greenhouse gas emissions and solid waste via input–output analysis. *Economic Systems Research*, *24*, 57–75.

Hu, W., Toriello, A., & Dessouky, M. (2018). Integrated inventory routing and freight consolidation for perishable goods. *European Journal of Operational Research*, *271*, 548–560.

Huang, C.-Y. (, Lai, C.-Y., & Cheng, K.-T. ((2009). Fundamentals of algorithms. In L.-T. Wang, Y.-W. Chang, & K.-T. Cheng (Eds.), *Electronic design automation* (pp. 173–234). Elsevier. https://doi.org/10.1016/B978-0-12-374364-0.50011-4

Iakovou, E., Bochtis, D., Vlachos, D., & Aidonis, D. (2016). Sustainable Agrifood Supply Chain Management. In E. Iakovou, D. Bochtis, D. Vlachos, & D. Aidonis (Eds.), *Supply Chain Management for Sustainable Food Networks*. Wiley. https://doi.org/10.1002/9781118937495.ch1

Imran, M., Kang, C., & Babar Ramzan, M. (2018). Medicine supply chain model for an integrated healthcare system with uncertain product complaints. *Journal of Manufacturing Systems*, *46*, 13–28.

Jabir, E., Panicker, V., & Sridharan, R. (2017). Design and development of a hybrid ant colony-variable neighbourhood search algorithm for a multi-depot green vehicle routing problem. *Transportation Research Part D*, *57*, 422–457.

Jankauskas, K., Papageorgiou, L., & Farid, S. (2019). Fast genetic algorithm approaches to solving discrete-time mixed integer linear programming problems of capacity planning and scheduling of biopharmaceutical manufacture. *Computers & Chemical Engineering*, *121*, 212–223.

Javid, A., & Azad, N. (2010). Incorporating location, routing and inventory decisions in supply chain network design. *Transportation Research Part E*, *46*, 582–597.

Jia, P., Diabat, A., & Mathiyazhagan, K. (2015). Analyzing the SSCM practices in the mining and mineral industry by ISM approach. *Resources Policy*, *46*, 76–85.

Kalsi, J., Tervo, T., Bachour, A., & Partinen, M. (2018). Sleep versus nonsleep-related fatal road accidents. *Sleep Medicine*, *51*, 148–152.

Karakostas, P., Panoskaltsis, N., Mantalaris, A., & Georgiadis, M. (2020a). Optimization of CAR T-cell therapies Supply Chains. *Computers & Chemical Engineering*, *139*. https://doi.org/10.1016/j.compchemeng.2020.106913

Karakostas, P., Sifaleras, A., & Georgiadis, C. (2020b). Adaptive GVNS Heuristics for Solving the Pollution Location Inventory Routing Problem. In N. Matsatsinis, Y. Marinakis, & P. Pardalos (Eds.), *Learning and Intelligent Optimization. LION 2019. Lecture Notes in Computer Science* (pp. 157–170). Springer, Cham.

Karakostas, P., Sifaleras, A., & Georgiadis, C. (2019a). Basic VNS algorithms for solving the pollution location inventory routing problem. In A. Sifaleras, S. Salhi, & J. Brimberg (Eds.), *ICVNS 2018. Lecture Notes in Computer Science* (pp. 64–76). Springer, Cham.

Karakostas, P., Sifaleras, A., & Georgiadis, C. (2019b). A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing. *Computers & Chemical Engineering*, *126*, 263–279. https://doi.org/10.1016/j.compchemeng.2019.04.015

Karakostas, P., Sifaleras, A., & Georgiadis, M. (2020c). Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory-routing problem. *Expert Systems with Applications*, *153*. https://doi.org/10.1016/j.eswa.2020.113444

Karantalis, V., Hernandez Schulman, I., Balkan, W., & Hare, J. (2015). Allogeneic cell therapy: A New Paradigm in Therapeutics. *Circulation Research*, *116*, 12–15.

Kidd, M., Lusby, R., & Larsen, J. (2020). Equidistant representations: Connecting coverage and uniformity in discrete biobjective optimization. *Computers & Operations Research*, *117*. https://doi.org/10.1016/j.cor.2019.104872

Koç, C., Bektaş, T., Jabali, O., & Laporte, G. (2014). The fleet size and mix pollution-routing problem. *Transportation Research Part B*, *70*, 239–254.

Koç, C., Bektaş, T., Jabali, O., & Laporte, G. (2016). The impact of depot location, fleet composition and routing on emissions in city logistics. *Transportation Research Part B*, *84*, 81–102.

Kong, L., Li, H., Luo, H., Ding, L., & Zhang, X. (2018). Sustainable performance of just-in-time (JIT) management in time-dependent batch delivery scheduling of precast construction. *Journal of Cleaner Production*, *193*, 684–701.

Kramer, R., Cordeau, J., & Iori, M. (2019). Rich vehicle routing with auxiliary depots and anticipated deliveries: An application to pharmaceutical distribution. *Transportation Research Part E*, *129*, 162–174.

Lai, X., & Hao, J. (2016). Iterated variable neighborhood search for the capacitated clustering problem. *Engineering Applications of Artificial Intelligence*, *56*, 102–120.

Larrain, H., Coelho, L., & Cataldo, A. (2017). A variable MIP Neighborhood Descent algorithm for managing inventory and distribution of cash in automated teller machines. *Computers & Operations Research*, *85*, 22–31.

Leenders, B., Velazquez-Martinez, J., & Fransoo, J. (2017). Emissions allocation in transportation routes. *Transportation Research Part D*, *57*, 39–51.

Letmathe, P., & Suares, M. (2017). A consumer-oriented total cost of ownership model for different vehicle types in Germany. *Transportation Research Part D: Transport and Environment*, *57*, 314–335.

Li, H., Mai, L., Zhang, W., & Tian, X. (2019). Optimizing the credit term decisions in supply chain finance. *Journal of Purchasing and Supply Management*, *25*, 146–156.

Li, J., Wang, D., & Zhang, J. (2018). Heterogeneous fixed fleet vehicle routing problem based on fuel and carbon emissions. *Journal of Cleaner Production*, *201*, 896–908.

Lin, Q., Zhao, Q., & Lev, B. (2020). Cold chain transportation decision in the vaccine supply chain. *European Journal of Operational Research*, *283*, 182–195.

Liu, B., Chen, H., Li, Y., & Liu, X. (2015). A pseudo-parallel genetic algorithm integrating simulated annealing for stochastic location-inventory-routing problem with consideration for returns in e-commerce. *Discrete Dynamics in Nature and Society*. https://doi.org/10.1155/2015/586581

Liu, R., Xie, X., & Garaix, T. (2014). Hybridization of tabu search with feasible and infeasible local searches for periodic home health care logistics. *Omega*, *47*, 17–32.

Liu, S., & Lee, S. (2003). A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. *International Journal of Advanced Manufacturing Technology*, *22*, 941–950.

Liu, S., & Lin, C. (2005). A heuristic method for the combined location routing and inventory problem. *International Journal of Advanced Manufacturing Technology*, *26*, 372–381.

Liu, Y., Dehghani, E., Jabalameli, M., Diabat, A., & Lu, C.-C. (2020). A coordinated location-inventory problem with supply disruptions: A two-phase queuing theory–optimization model approach. *Computers & Industrial Engineering*, *142*. https://doi.org/10.1016/j.cie.2020.106326

Malik, N., & Durby, M. (2015). Cell Therapy Landscape: Autologous and Allogeneic Approaches. In A. Atala & J. Allickson (Eds.), *Translational Regenerative Medicine*. https://doi.org/10.1016/B978-0-12-410396-2.00007-4

Mallidis, I., Sariannidis, N., Vlachos, D., Yakavenka, V., Aifadopoulou, G., & Zopounidis, K. (2020). Optimal inventory control policies for avoiding food waste. *Operational Research - An International Journal*. https://doi.org/10.1007/s12351-020-00554-w

Mallidis, I., Vlachos, D., Iakovou, E., & Dekker, R. (2014). Design and planning for green global supply chains under periodic review replenishment policies. *Transportation Research Part E: Logistics and Transportation Review*, *72*, 210–235.

Marinakis, Y., Marinaki, M., & Migdalas, A. (2019). A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows. *Information Sciences*, *481*, 311–329.

Markov, I., Bierlaire, M., Cordeau, J.-F., Maknoon, Y., & Varone, S. (2020). Waste collection inventory routing with non-stationary stochastic demands. *Computers & Operations Research*, *113*. https://doi.org/10.1016/j.cor.2019.104798

Martins, C., & Pato, M. (2019). Supply chain sustainability: A tertiary literature review. *Journal of Cleaner Production*, *225*, 995–1016.

Max Shen, Z., & Qi, L. (2007). Incorporating inventory and routing costs in strategic location models. *European Journal of Operational Research*, *179*, 372–389.

Micheli, J., & Mantella, F. (2018). Modelling an environmentally-extended inventory routing problem with demand uncertainty and a heterogeneous fleet under carbon control policies. *International Journal of Production Economics*, *204*, 316–327.

Mjirda, A., Jarboui, B., Macedo, R., Hanafi, S., & Mladenovic, N. (2014). A two phase variable neighborhood search for the multi-product inventory routing problem. *Computers & Operations Research*, *52*, 291–299. https://doi.org/10.1016/j.cor.2013.06.006

Mjirda, A., Todosijević, R., Hanafi, S., Hansen, P., & Mladenović, N. (2017). Sequential variable neighborhood descent variants: An empirical study on the traveling salesman problem. *International Transactions in Operational Research*, *24*, 615–633.

Mladenović, N., Alkandari, A., Todosijević, R., Pei, J., & Pardalos, M. (2020). Less is more approach: Basic variable neighborhood search for the obnoxious p-median problem. *International Transactions in Operational Research*, *27*, 480–493.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, *24*, 1097–1100. https://doi.org/10.1016/S0305-0548(97)00031-2

Mo, F., Srinivasan, M., Ma, R., Smith, T., McKenna, M., Atilla, E., Atilla, P., Heslop, H., Brenner, M., & Mamonkin, M. (2019). Rejection-Resistant Off-the-Shelf T Cells for Adoptive Cell Therapy. *Biology of Blood and Marrow Transplantation*, *25*, S168. https://doi.org/10.1016/j.bbmt.2018.12.471

Morizane, A., Doi, D., Kikuchi, T., Okita, K., Hotta, A., Kawasaki, T., Hayashi, T., Onoe, H., T., S., Yamanaka, S., & Takahashi, J. (2013). Direct comparison of autologous and allogeneic transplantation of iPSC-derived neural cells in the brain of a non-human primate. *Stem Cell Reports*, *1*, 283–292. https://doi.org/10.1016/j.stemcr.2013.08.007

Moussavi, S., Mahdjoub, M., & Grunder, O. (2019). A matheuristic approach to the integration of worker assignment and vehicle routing problems: Application to home healthcare scheduling. *Expert Systems With Applications*, *125*, 317–332.

Narbonne, P. (2018). The effect of age on stem cell function and utility for therapy. *Cell Medicine*, *10*. https://doi.org/10.1177/2155179018773756

Nekooghadirli, N., Tavakkoli-Moghaddam, R., Ghezavati, V., & Javanmard, S. (2014). Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics. *Computers & Industrial Engineering*, *76*, 204–221.

Nenes, G., Panagiotidou, S., & Tagaras, G. (2010). Inventory management of multiple items with irregular demand: A case study. *European Journal of Operational Research*, *205*, 313–324.

Nikolopoulou, A., Repoussis, P., Tarantilis, C., & Zachariadis, E. (2017). Moving products between location pairs: Cross-docking versus direct-shipping. *European Journal of Operational Research*, *256*, 803–819.

Numbi, F., & Kupa, B. (2017). The potential of next-generation supply chains to ease drc's "casse-tête". *Vaccine*, *35*, 2105–2106.

Oudouar, F., Lazaar, M., & Miloud, Z. (2020). A novel approach based on heuristics and a neural network to solve a capacitated location routing problem. *Simulation Modelling Practice and Theory*, *100*. https://doi.org/10.1016/j.simpat.2019.102064

Panagiotidou, S., Nenes, G., Zikopoulos, C., & Tagaras, G. (2017). Joint optimization of manufacturing/remanufacturing lot sizes under imperfect information on returns quality. *European Journal of Operational Research*, *258*, 537–551.

Papalitsas, C., Karakostas, P., & Andronikos, T. (2019). A Performance Study of the Impact of Different Perturbation Methods on the Efficiency of GVNS for Solving TSP. *Applied System Innovation*, *2*. https://doi.org/10.3390/asi2040031

Papathanasiou, M., Stamatis, C., Lakelin, M., Farid, N., S. Titchener-Hooker, & Shah, N. (2020). Autologous CAR T-cell therapies supply chain: Challenges and opportunities? *Cancer Gene Therapy*. https://doi.org/10.1038/s41417-019-0157-z

Pereira Chilima, T., Moncaubeig, F., & Farid, S. (2018). Impact of allogeneic stem cell manufacturing decisions on cost of goods, process robustness and reimbursement. *Biochemical Engineering Journal*, *137*, 132–151.

Rabbami, M., Heidari, R., & Yazdanparast, R. (2019). A stochastic multi-period industrial hazardous waste location-routing problem: Integrating NSGA-II and Monte Carlo simulation. *European Journal of Operational Research*, *272*, 945–961.

Rafie-Majd, Z., Pasandideh, S., & Naderi, B. (2018). Modelling and solving the integrated inventory-location-routing problem in a multi-period and multi-perishable product supply chain with uncertainty: Lagrangian relaxation algorithm. *Computers & Chemical Engineering*, *109*, 9–22.

Rayat, F., Musavi, M., & Bozorgi-Amiri, A. (2017a). Bi-objective reliable location-inventory-routing problem with partial backordering under disruption risks: A modified AMOSA approach. *Applied Soft Computing*, *59*, 622–643.

Rayat, F., Musavi, M., & Bozorgi-Amiri, A. (2017b). Bi-objective reliable location-inventory-routing problem with partial backordering under disruption risks: A modified AMOSA approach. *Applied Soft Computing*, *59*, 622–643.

Reichert, A., Holtz-Rau, C., & Scheiner, J. (2016). GHG emissions in daily travel and long-distance travel in Germany - Social and spatial correlates. *Transportation Research Part D*, *49*, 25–43.

Reza Sajjadi, S., Hamidi, M., & Hossein Cheraghi, S. (2013). Multi-product capacitated location routing inventory problem. *International Journal of Modern Engineering*, *13*.

Riquelme-Rodriguez, J., Gamache, M., & Langevin, A. (2016). Location arc routing problem with inventory constraints. *Computers & Operations Research*, *76*, 84–94.

Sadjadi, S., Ziaei, Z., & Pishvaee, M. (2019). The design of the vaccine supply network under uncertain condition: A robust mathematical programming approach. *Journal of Modelling in Management*, *14*, 841–871.

Saif-Eddine, A., El-Beheiry, M., & El-Kharbotly, A. (2019). An improved genetic algorithm for optimizing total supply chain cost in inventory location routing problem. *Ain Shams Engineering Journal*, *10*, 63–76.

Sánchez-Oro, J., Sevaux, M., Rossi, A., Marti, R., & Duarte, A. (2017). Improving the performance of embedded systems with variable neighborhood search. *Applied Soft Computing*, *53*, 217–226.

Saragih, N., Bahagia, S., Suprayogi, & Syabri, I. (2019). A heuristic method for location-inventory-routing problem in a three-echelon supply chain system. *Computers & Industrial Engineering*, *127*, 875–886.

Savadkoohi, E., Mousazadeh, M., & Torabi, S. (2018). A possibilistic location-inventory model for multi-period perishable pharmaceutical supply chain network design. *Chemical Engineering Research and Design*, *138*, 490–505.

Scruggs, B., Semon, J., Zhang, X., Zhang, S., Bowles, A., Pandey, A., Imhof, K., Kalueff, A., Gimble, J., & Bunnell, B. (2013). Age of the Donor Reduces the Ability of Human Adipose-Derived Stem Cells to Alleviate Symptoms in the Experimental Autoimmune Encephalomyelitis Mouse Model. *Stem Cells Translational Medicine*, *2*, 797–807. https://doi.org/10.5966/sctm.2013-0026

Seyedhosseini, S., Bozorgi-Amiri, A., & Darei, S. (2014). An integrated location-routing-inventory problem by considering supply disruption. *iBusiness*, *6*, 29–37.

Shadid, W. (2017). A framework for managing organizations in complex environments. *Construction Management and Economics*, *36*, 182–202.

Sifaleras, A., & Konstantaras, I. (2017). Variable neighborhood descent heuristic for solving reverse logistics multi-item dynamic lot-sizing problems. *Computers & Operations Research*, *78*, 385–392.

Sifaleras, A., & Konstantaras, I. (2015). General variable neighborhood search for the multi-product dynamic lot sizing problem in closed-loop supply chain. *Electronic Notes in Discrete Mathematics*, *47*, 69–76.

Sifaleras, A., & Konstantaras, I. (2018). A survey on variable neighborhood search methods for supply network inventory. In I. Bychkov, V. A. Kalyagin, P. M. Pardalos, & O. Prokopyev (Eds.), *Network Algorithms, Data Mining, and Applications. NET 2018. Springer Proceedings in Mathematics & Statistics* (pp. 71–82).

Skouri, K., Sifaleras, A., & Konstantaras, I. (2018). Open problems in green supply chain modeling and optimization with carbon emission targets. In P. M. Pardalos & A. Migdalas (Eds.), *Open problems in optimization and data analysis* (pp. 83–90). Springer Optimization; Its Applications.

Smiti, N., Dhiaf, M., Jarboui, B., & Hanafi, S. (2020). Skewed general variable neighborhood search for the cumulative capacitated vehicle routing problem. *International Transactions in Operational Research*, *27*, 651–664.

Soon, K., Lim, J., Parthiban, R., & Ho, M. (2019). Proactive eco-friendly pheromone-based green vehicle routing for multi-agent systems. *Expert Systems With Applications*, *121*, 324–337.

Soysal, M., Cimen, M., Belbag, S., & Togrul, E. (2019). A review on sustainable inventory routing. *Computers & Industrial Engineering*, *132*, 395–411.

Speranza, M. (2018). Trends in transportation and logistics. *European Journal of Operational Research*, *264*, 830–836.

Stadler, H., Kilger, C., & Meyr, H. (2015). *Supply Chain Management adn Advanced Planning: Concepts, Models, Software and Case Sudies*. Springer.

Tavakkoli-Moghaddam, R., Forouzanfar, F., & Ebrahimnejad, S. (2013). Incorporating location, routing and inventory decisions in a bi-objective supply chain design problem with risk pooling. *Journal of Industrial Engineering International*, 9–19. https://doi.org/10.1186/2251-712X-9-19

Tavana, M., Abtahi, A., Caprio, D., Hashemi, R., & Yousefi-Zenouz, R. (2018). An integrated location-inventory-routing humanitarian supply chain network with pre- and post-disaster management considerations. *Socio-Economic Planning Sciences*, *64*, 21–37.

Tirkolaee, E., Mahdavi, I., Esfahani, M., & Weber, G.-W. (2020). A robust green location-allocation-inventory problem to design an urban waste management system under uncertainty. *Waste Management*, *102*, 340–350.

Todosijević, R., Hanafi, S., Urøosević, D., Jarboui, B., & Gendron, B. (2017). A general variable neighborhood search for the swap-body vehicle routing problem. *Computers & Operations Research*, *78*, 468–479.

Todosijević, R., Mladenović, M., Hanafi, S., Mladenović, N., & Crévits, I. (2016). Adaptive general variable neighborhood search heuristics for solving the unit commitment problem. *International Journal of Electrical Power & Energy Systems*, *78*, 873–883.

Toro, M., Franco, F., Echeverri, M., & Guimaraes, F. (2017). A multi-objective model for the green capacitated location-routing problem considering environmental impact. *Computers & Industrial Engineering*, *110*, 114–125.

Turan, B., Minner, S., & Hartl, R. (2017). A VNS approach to multi-location inventory redistribution with vehicle routing. *Computers & Operations Research*, *78*, 526–536.

Turkay, M., Oruc, C., Fujita, K., & Asakura, T. (2004). Multi-company collaborative supply chain management with economical and environmental considerations. *Computers & Chemical Engineering*, *28*, 985–992.

Vahdani, B., Veysmoradi, D., Noori, F., & Mansour, F. (2018). Two-stage multi-objective location-routing-inventory model for humanitarian logistics network design under uncertainty. *International Journal of Disaster Risk Reduction*, *27*, 290–306.

Vicente, J., Relvas, S., & Barbosa-Povoa, A. (2015). Multi-period and Multi-product Inventory Management Model with Lateral Transshipments. In J. Almeida, J. Oliveira, & A. Pinto (Eds.), *Operational research. cim series in mathematical sciences*. Springer.

Vivas, R., Sant'Anna, A., Esquerre, K., & Freires, F. (2020). Integrated method combining analytical and mathematical models for the evaluation and optimization of sustainable supply chains: A Brazilian case study. *Computers & Industrial Engineering*, *139*. https://doi.org/10.1016/j.cie.2019.01.044

Vormittag, P., Gunn, R., Ghorashian, S., & F.S., V. (2018). A guide to manufacturing CAR T cell therapies. *Current Opinion in Biotechnology*, *53*, 164–181.

Wang, K., Liu, Y., Li, J., Wang, B., Bishop, R., White, C., Das, A., Levine, A., Ho, L., Levine, B., & Fesnak, A. (2019). A multiscale simulation framework for the manufacturing facility and supply chain of autologous cell therapies. *Cytotherapy*, *21*, 1081–1093.

Wang, S., & Ye, B. (2018). A comparison between just-in-time and economic order quantity models with carbon emissions. *Journal of Cleaner Production*, *187*, 662–671.

Wang, X., Kong, Q., Papathanasiou, M., & Shah, N. (2018). Precision healthcare supply chain design through multi-objective stochastic programming. In M. Eden, M. Ierapetritou, & G. Towler (Eds.), *Proceedings of the* 13*th international symposium on process systems engineering - pse 2018*. Elsevier.

Wang, X., & Riviere, I. (2016). Clinical manufacturing of CAR T cells: Foundation of a promising therapy. *Molecular Therapy — Oncolytics*, *3*. https://doi.org/10.1038/mto.2016.15

Wei Teng, C., Foley, L., O'Neill, P., & Hicks, C. (2014). An analysis of supply chain strategies in the regenerative medicine industry—Implications for future development. *International Journal of Production Economics*, *149*, 211–225.

Williamson, D., & Shmoys, D. (2011). *The Design of Approximation Algorithms*. Cambridge University Press.

Xie, A., & Murphy, W. (2019). Engineered biomaterials to mitigate growth factor cost in cell biomanufacturing. *Current Opinion in Biomedical Engineering*, *10*, 1–10.

Xu, Z., & Cai, Y. (2018). Variable neighborhood search for consistent vehicle routing problem. *Expert Systems with Applications*, *113*, 66–76.

Yu, X., Zhou, Y., & Liu, X.-F. (2019a). A novel hybrid genetic algorithm for the location routing problem with tight capacity constraints. *Applied Soft Computing*, *85*. https://doi.org/10.1016/j.asoc.2019.105760

Yu, Y., Wang, S., Wang, J., & Huang, M. (2019b). A branch-and-price algorithm for the heterogeneous fleet green vehicle routing problem with time windows. *Transportation Research Part B: Methodological*, *122*, 511–527.

Yuchi, Q., Wang, N., He, Z., & Chen, H. (2018). Hybrid heuristic for the location-inventory-routing problem in closed-loop supply chain. *International Transactions in Operational Research*. https://doi.org/10.1111/itor.12621

Zachariadis, E., Tarantilis, C., & Kiranoudis, C. (2009). An integrated local search method for inventory and routing decisions. *Expert Systems with Applications*, *36*, 10239–10248.

Zahiri, B., Zhuang, J., & Mohammadi, M. (2017). Toward an integrated sustainable-resilient supply chain *Apharmaceuticalcasestudy*. *Transportation Research Part E*, *103*, 109–142.

Zgaya, H., & Hammadi, S. (2016). Logistics Engineering. In *Logistics Engineering and Health* (pp. 1–53). Elsevier. https://doi.org/10.1016/B978-1-78548-044-7.50001-4

Zhalechian, M., Tavakkoli-Moghaddam, R., Zahiri, B., & Mohammadi, M. (2016). Sustainable design of a closed-loop location-routing-inventory supply chain network under mixed uncertainty. *Transportation Research Part E*, *89*, 182–214.

Zhang, B., Li, H., Li, S., & Peng, J. (2018). Sustainable multi-depot emergency facilities location-routing problem with uncertain information. *Applied Mathematics and Computation*, *333*, 506–520.

Zhang, Y., Qi, M., Miao, L., & Liu, E. (2014). Hybrid metaheuristics solutions to inventory location routing problem. *Transportation Research Part E*, *70*, 305–323.

Zhao, J., & Ke, G. (2017). Incorporating inventory risks in location-routing models for explosive waste management. *International Journal of Production Economics*, *193*, 123–136.

Zheng, X., Yin, M., & Zhang, Y. (2019). Integrated optimization of location, inventory and routing in supply chain network design. *Transportation Research Part B: Methodological*, *121*, 1–20.

Zhu, W., Ng, S., Wang, Z., & Zhao, X. (2017). The role of outsourcing management process in improving the effectiveness of logistics outsourcing. *International Journal of Production Economics*, *188*, 29–40.

Zikopoulos, C., & Tagaras, G. (2015). Reverse supply chains: Effects of collection network and returns classification on profitability. *European Journal of Operational Research*, *246*, 435–449.