Thèse de doctorat de l'UTT

William Javier GUERRERO RUEDA

Modèles et méthodes d'optimisation pour le problème de localisation-routage avec contraintes de stockage

Spécialité :
Optimisation et Sûreté des Systèmes

2014TROY0002

Année 2014

Thèse en cotutelle avec la Universidad de Los Andes - Bogota - Colombie

Universidad de los Andes Colombia

utt université de technologie Troyes

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

William Javier GUERRERO RUEDA

le 27 janvier 2014

Modèles et méthodes d'optimisation pour le problème de localisation-routage avec contraintes de stockage

JURY

M. M. SEVAUX	PROFESSEUR DES UNIVERSITES	Président (Rapporteur)
M. C.-A. AMAYA	PROFESOR ASOCIADO	Examinateur
M. A. LANGEVIN	PROFESSEUR ASSOCIE	Rapporteur
M. C. PRINS	PROFESSEUR DES UNIVERSITES	Examinateur
Mme C. PRODHON	MAITRE DE CONFERENCES - HDR	Directrice de thèse
Mme N. VELASCO	PROFESOR ASOCIADO	Directrice de thèse
M. J. G. VILLEGAS	PROFESOR ASOCIADO	Examinateur

Acknowledgments

To my family and my girlfriend. They have support me all the way in the development of this thesis and they always believed I could follow this dream. This thesis is not my personal achievement, it is theirs. I could not be more grateful that I have them by my side and I love them all very much.

To my mentor and thesis advisor, Professor Nubia Velasco, to whom I admire both as a researcher and a person. She has taught me most of the things I know today, she inspired and encouraged me to follow the path of science and I hope I will be someday as good researcher as she is today. That is why I would like to express my sincere gratitude and to state that I can not imagine having a better advisor and mentor for my Ph.D study.

To professor Caroline Prodhon, words are not enough to express my gratitude to her as my supervisor. Her expertise, understanding, and patience enlightened me to stay focus on doing what I love while being far away from my home. I appreciate her vast knowledge, skills and teaching. Her commitment to science is truly inspiring.

To my co-advisor, Professor Ciro Alberto Amaya, sincere gratitude for his support in the development of this thesis and for teaching me uncountable valuable lessons.

Special thanks to Professors Marc Sevaux and André Langevin for taking the time of reading this thesis, for making significant contributions with their suggestions, and for traveling this far.

To Professor Christian Prins, the greatest researcher I have ever met. My sincere thanks for talked me into doing a Ph.D., for accepting me on your research team, for the many times you supported my work, and the remarkable lessons you taught me.

To my research labs LOSI and PYLO. I am incredibly thankful for having the chance to belong to this amazing teams of research. They are both composed by people I deeply admire and respect. Many thanks for always having open doors, and I am proud to call most of you my friends.

To my post-doc friends Jan Melechovsky and Maria Soto. Also to my Ph.D. students colleagues in France and Colombia Fabian Castaño, Karen Niño, Juan Pablo Caballero, Freddy Perez, Nelson Tovar, Julian Autuory, Thibaut Vidal, Andrés Bernate, Julian Michallet, Atefeh Moghaddam, David Cadet, Guillermo Ciro, Elyn Solano, Juan Carlos Rivera, Natalia Duarte, Marthy Garcia, Magda Torres, Camilo Gómez, and Diego Castiblanco. Thank you for sharing with me many productive discussions, and for the others that were not.

To the crew at Uniandes and at Pole ROSAS, Veronique Banse, Marie-Joseph, Isabelle Leclercq, Pascale Denis, Thérèse Kazarian, Patricia Barreto, España Angarita, Alejandra Nuñez, Catalina Muñoz, Marcela Jaramillo, Natalia Valencia, Lina Jaramillo.

Special thanks to you for making easier, the already very difficult task of getting to the end.

To my undergraduate students to whom I was advisor on their final projects. Jhon Leonardo Vargas, Karen Amortegui, Andrea Huertas, and Alvaro Jose Rodriguez. I own you plenty since you taught me much more than what I taught you.

And last but not least, to my undergraduate and graduate students at the courses I teach in Los Andes University and at UTT. Because it is for them that I have to work harder everyday on my teaching and communication skills. You made me less shy and more open to new ways of thinking.

Abstract

Abstract in English

The problem of designing a supply chain including simultaneously routing and inventory management decisions is studied in this thesis. The objective is to select a subset of depots to open, the inventory policies for a 2-echelon system, and the set of routes to perform distribution from the upper echelon to the next using a homogeneous fleet of vehicles over a finite planning horizon. Demand is considered to be known. Applications are found in humanitarian logistics, military logistics, and supply chain design on the pharmaceutical industry, among others.

To solve the problem, two matheuristic procedures are developed. On the first part a cooperative algorithm combining exact methods for the supply chain design problem and routing heuristics is presented. On the second part, a partition is proposed using a Dantzig-Wolf reformulation on the routing variables. An hybridization between column generation, Lagrangian relaxation and local search is proposed in this part, put together as a heuristic method. Furthermore, results demonstrate the capability of the algorithms to compute high quality solutions and empirically estimate the improvement in the cost function of the proposed model when compared to a sequential optimization approach. Furthermore, results of the proposed methodologies on benchmark instances for subproblems are studied as well. Those are the capacitated location-routing problem, the inventory-routing problem, and the generalized elementary shortest path problem.

Keywords: Operations Research, Combinatorial optimization, Metaheuristics, Transportation problems (Programming), Inventory control, Materials management, Business logistics, Transportation.

Résumé en Français

Cette thèse considère le problème consistant à intégrer les décisions de routage et stockage lors de la conception de la chaîne logistique. Le but est de sélectionner des dépôts parmi un ensemble de candidats possédant une capacité limitée pour desservir un ensemble de clients/détaillants à l'aide d'une flotte de véhicules. Ces derniers ont également une capacité limitée suffisamment grande pour visiter plusieurs détaillants par route. On cherche à déterminer la localisation des dépôts à utiliser et les tournées des véhicules afin de maintenir un niveau optimal des stocks. La demande chez les détaillants est supposée connue à l'avance. Des applications dans les domaines de la logistique humanitaire et militaire sont envisageables ainsi que dans l'industrie pharmaceutique, par exemple.

Pour résoudre le problème, deux matheuristiques sont proposées. Dans la première partie, une méthode coopérative qui combine des méthodes exactes pour le problème de conception de la chaîne logistique et des méthodes heuristiques de routage est présentée. Dans la deuxième partie, une méthode utilisant une réformulation du problème adaptée pour l'application d'un décomposition de Dantzig-Wolf sur les variables de routage est proposée. L'algorithme intègre les concepts de génération de colonnes, relaxation lagrangienne et recherche locale. Les résultats montrent la capacité des algorithmes à trouver des solutions de bonne qualité et nous estimons d'une manière empirique l'impact de considérer un modèle intégré plutôt qu'une méthode d'optimisation séquentielle. De plus, les résultats des méthodes présentées sur des sous-problèmes trouvés dans la littérature sont aussi étudiés. Ces sont: le problème de localisation-routage, le problème de tournées avec gestion de stocks, et le problème de plus court chemin généralisé.

Mots-clés: Recherche opérationnelle, Optimisation combinatoire, Transport, Métaheuristiques, Problème de transport (programmation), Gestion des stocks, Logistique (organisation), Gestion de l'approvisionnement.

Resumen en Español

Esta tesis considera el problema de diseño de la cadena de abastecimiento incluyendo simultáneamente las decisiones de ruteo y de inventario. El objetivo es seleccionar un conjunto de depósitos por abrir, las políticas de gestión de inventarios en un sistema de dos niveles, y el conjunto de rutas para hacer la distribución desde el primer nivel hacia el segundo usando una flota de vehículos homogéneos en un horizonte de planeación finito y considerando demanda determinista. Algunas aplicaciones son encontradas en el campo de la logística humanitaria y militar, como también en el diseño de cadenas de suministro en la industria farmacéutica.

Para resolver el problema, dos métodos matheuristicos son desarrollados. En la primera parte se estudia un algoritmo cooperativo que combina métodos exactos para el diseño de la cadena logística y heurísticas de ruteo. En la segunda parte, se propone descomponer el problema usando la reformulación de Dantzig-Wolf sobre las variables de ruteo. Se integran en esta parte los métodos de generación de columnas, relajación Lagrangiana y búsqueda local. Adicionalmente, los resultados muestran la capacidad de los algoritmos para encontrar soluciones de buena calidad y se estima empíricamente el costo de usar métodos basados en optimización secuencial sobre el modelo propuesto. Finalmente, se presentan los resultados de las metodologías propuestas en otros problemas de la literatura. Estos son: el problema de localización y ruteo, el problema de ruteo de inventarios, y el problema de camino más corto generalizado.

Palabras clave: Investigación de operaciones, Optimización combinatoria, Transporte, Metaheurística, Problemas de transporte (programación), Control de inventarios, Abastecimiento, Logística (empresarial).

Contents

1. Introduction

1.1. About the content and objective of the thesis

The subject of this thesis is the inventory-location-routing problem (ILRP). It is the problem of deciding the location of depots while taking into consideration routing and inventory policies on a finite planning horizon. From the operations research point of view, the ILRP might be described as the integration of routing decisions with the traditional supply chain design problem [2].

On a greater picture, the presented models and solution algorithms are part of the field of combinatorial optimization. More specifically to the domain of transportation logistics. Ever since the traveling salesman problem (TSP) was proven to be NP-hard, the number of papers published on exact and heuristic methods for the TSP and related problems such as the capacitated vehicle routing problem (CVRP) and variants, is increasing [7].

The natural trend in recent research is to model more accurately the situations that decision makers face, in order to fulfill current industrial needs. On one hand, the latest advances the inventory-routing problem have emerged by combining the knowledge on inventory management with the contributions made on vehicle routing [3]. On the other hand, location theory and network design could not be left behind. The location-routing problem is a variation of the facility location problem where depots are linked to clients using tours [15]. Industrial applications include distribution problems, design of public transportation networks, etc.

Thus, one of the targets of this model is to extend the Location-Routing problem (LRP) [17] to be considered on a multi-period planning horizon together with inventory constraints. This work was mainly inspired on the pharmaceutical industry, where location costs are in the same order of magnitude as operational costs on the long run [8].

The problem, as the reader might notice, integrates decisions that are often considered as strategic (location), tactical (inventory management policies), and operational (routing). Some researchers claim that these three kinds of decisions should not be made simultaneously in most cases since strategical decisions are fixed for long periods of time (more than a year), while tactical and operational decisions are changeable on shorter periods of time. In fact, the results achieved in this thesis do not prove that this argument is wrong.

Nonetheless, this thesis will expose application cases where whether location decisions are not strategical (according to the definition just provided) like in military and humanitarian logistics; cases where assuming direct deliveries might lead to suboptimal solutions as in cases where vehicle capacity exceeds largely the maximum clients demand; and a decision-aid tool to perform "what-if" analysis when evaluating the supply chain performance.

To the best of our knowledge, there are few papers dealing with similar problems. For example, Liu and Lee[10], and Liu and Lin[11] propose a non-linear programming model and propose to solve it by a route-first locate-allocate second approach. Delivery quantities are computed as the Economic Order Quantities (EOQ) of the aggregated members of the route, allowing a single-period problem to be solved under the assumption that all the customers are visited with the same frequency. In reality, routing has to be revised if some customers do not need to be visited as often as others. In addition, the trade-off between holding costs and routing costs is never explored. Further, Reza Sajjadi et al. [18] extended the model to include the production plants location-allocation decisions and a multi-item environment. Solutions are found by solving the inventory problem first, and then the resulting LRP. Randomly generated instances of up to 40 products, 350 clients and one period are solved heuristically.

Other authors present an ILRP model with inventory decisions at depots only and considering stochastic demand [19]. They include transportation costs from suppliers to satellite depots and present a different approach by not deriving the exact routing decisions to retailers but making an approximation of the routing costs using an upper bound formula. Furthermore, Ahmadi-Javid et al.[1] propose an extension of the model to include capacity decisions of depots within discrete levels, considering a (Q, r) policy with safety stock and routing decisions. Inventory decisions are taken implicitly by evaluating each solution with the cost of an EOQ-like formula.

These approaches present three weak points: first, they do not link the ordering cost on the EOQ-like formula with the routing costs, while they are indeed correlated. Second, most methods present single-period routing decisions considering the expected value of the quantities demanded by retailers and assuming that the shipment frequency is the same for every retailer. In fact, frequencies of visits for each retailer could be different to optimize both routing and inventory management costs. Therefore, single-period routing decisions are not an appropriate approach when integrating inventory and distribution decisions because the choice between replenishing every retailer in a route or not replenishing any at all is too restrictive and may lead to sub-optimal solutions. Third, we conclude that different simplified variants of the problem are studied in the literature either by assuming inventory decisions at depots or at clients. Even when solving the problem on its stochastic version seems more difficult, in fact it is easier if routing decisions are made for a single period and inventory decisions are limited to a single echelon.

In brief, fig. 1 presents a scheme summarizing the four axis of research developed in the presented thesis and the contributions made on each of them. Axis **A** focuses on making a full description of optimization approaches integrating location decisions together with routing and inventory management on a supply chain. The main contributions of the thesis on this axis are to present two mathematical formulations for the ILRP, to compare the presented models with the literature, and to expose real-life application contexts.

On axis of research **B**, the study of solution methods is made. In detail, two new matheuristic methodologies to solve the problem are developed. One based on the decomposition of decisions of the model and a second one based on the Dantzig-Wolfe formulation. The proposed algorithms provide good quality solutions computed within reasonable time.

Further, axis **C** is to test the developed heuristic algorithms on benchmark instances for some sub-problems and to provide computational results. Examples will be the single and multiple depot inventory-routing problem and the location-routing problem.

Finally, this thesis has a fourth axis **D** of research focused on a new sub-problem inspired on the pricing of columns for the Dantzig-Wolfe formulation of the ILRP. It is denoted as the generalized elementary shortest path problem and real-life application contexts are discussed. Exact and heuristic methodologies are proposed and their performance is evaluated on new benchmark instances.

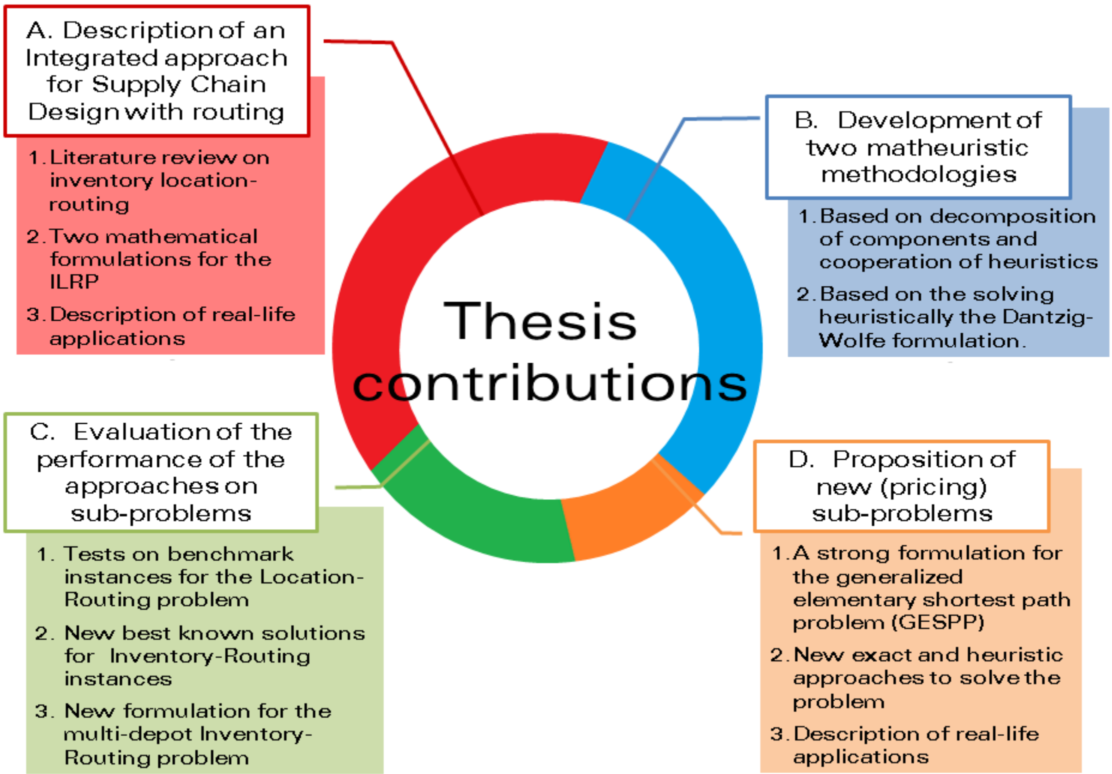

Figure 1: Main contributions of the thesis

As future research, the presented models might be extended by considering further features and/or less restrictive assumptions. For example, the assumption of a system with deterministic demand might be reconsidered by implementing more sophisticated modeling techniques such as those based on stochastic programming to capture the variability of parameters and the dynamical aspect of the decisions. In reality, decisions are made by stages. While location decisions are static and fixed from the first stage, operational decisions (meaning routing and inventory strategies) could be associated to current states of the system on a dynamic framework (see Mendoza and Villegas[13], Sörensen and Sevaux[20] for more on the applicability of stochastic variants of the CVRP).

Other extensions include: 1) multi-product setting, 2) allowed split deliveries, 3) time-windows for deliveries [22], 4) considering heterogeneous fleet of vehicles, 5) time-dependent travel times, 6) length constrained routes, 7) constraints from the arc routing problem [14], and 8) pickup and delivery routing schemes [4, 21]. Further, the objective function could be modified, as in the cumulative capacitated vehicle routing problem (CCVRP) [16] to provide solution that respond better to humanitarian logistics. Even more, multiple objectives seem to respond better to real-life situations [9]. Therefore, future research could also be to formulate a multi-objective ILRP, minimizing simultaneously: location costs, operational costs, risk measures, CO_2 emissions;

while maximizing: service level and/or total profit for instance.

1.2. Motivation to study the ILRP

Several pressures are being imposed to optimize logistics systems. Among them, economic, social and environmental factors should be focus of interest since they are all critical when making decisions such as a supply chain design.

Economic factors are relevant since successful companies must use their available resources efficiently (material, human, and financial included). Better planning of the supply chain implies taking full advantage of depots, retailers and vehicles capacity to minimize the overall logistics costs.

Social factors are addressed too. Think of the case of a catastrophe (earthquake, tsunami, among others) where humanitarian logistics must be set in place to distribute water, medicine and other supplies to a society using limited resources. In such cases, the ILRP must be solved to locate the facilities were these supplies will be stored and the way of performing distribution under capacity constraints. Failing to provide basic supplies in such situations could result in lost lives or other permanent damages to society.

Finally, it is widely known that every product that we consume has an environmental footprint [6]. Governments and customers are nowadays demanding "greener" products. It can be argued that this objective is obtained by minimizing inventory levels and distances traveled by vehicles making distribution. Shorter routes have less CO_2 emissions while smaller inventory levels require less storage energy (e.g. in case of refrigerated products). In the best case scenario, the result of the optimization provides solutions reaching a sustainable state of the supply chain.

All along the next chapters, the reader will note that little research has been done before to develop solution methods for the ILRP. This research is motivated by the idea of optimizing the supply chain from the conception stages while integrating a long-term thinking.

1.3. The structure of the thesis

The ILRP could be interpreted as the overlapping of three branches of Operations Research: vehicle routing, facility location and inventory management. Fig. 2 presents an scheme of the literature related to the ILRP and how each chapter of this thesis is classified. Between the intersection of vehicle routing problems and inventory management, Inventory-Routing models are proposed. By combining facility locations decisions and vehicle routing, the problem known as the Location-routing problem takes place. Also, Supply chain design models and Inventory-Location models are the result of combining facility location literature with inventory management decisions or constraints.

The emphasis of this thesis is on the integrated problems between vehicle routing with inventory management and location decisions. We aim to study the interaction between the decision of quantities to deliver and the routing construction. This is the less studied topic in the literature whereas the facility location problem and the inventory management problem have received important attention [5, 12].

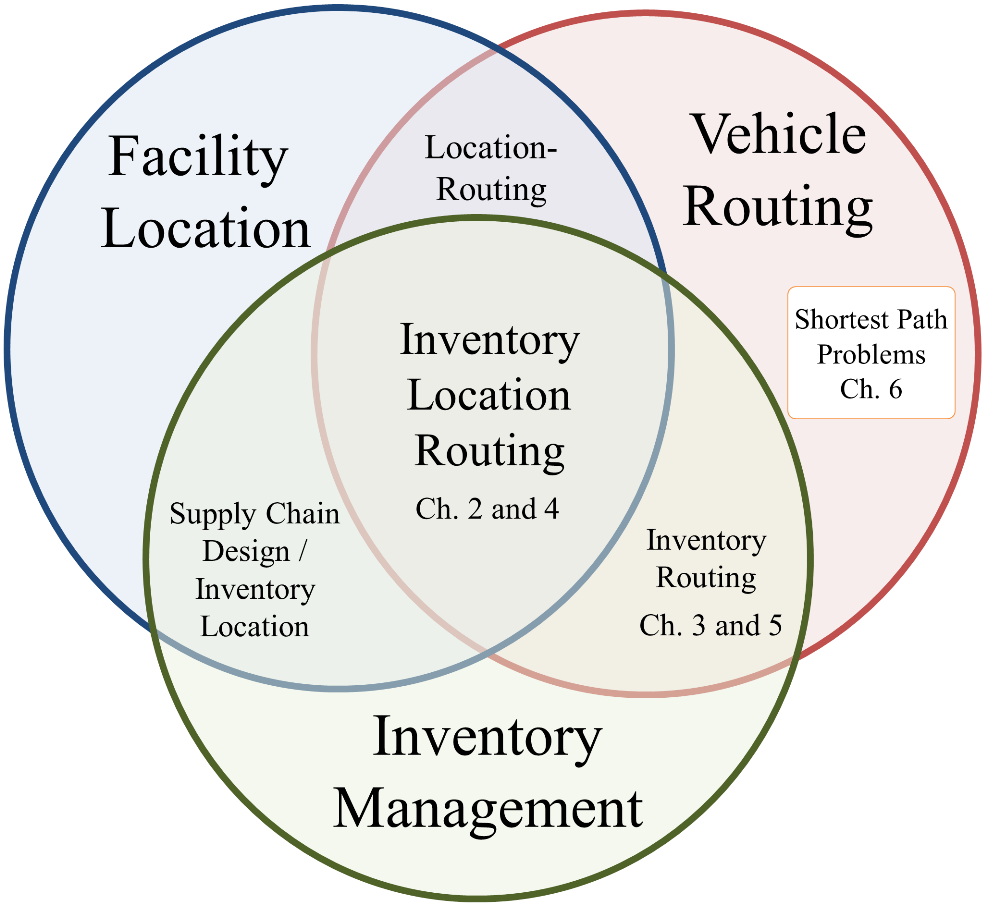

Figure 2: Scheme of OR literature linked to the ILRP

Two parts compose this thesis. Each part studies a different hybrid approach to solve the presented problem and applies a specific approach on some subproblems of the ILRP. Part one is dedicated to a cooperative method between MIP solvers with routing heuristics. Chapter 2 studies this cooperation concept on the ILRP and provides a three-index formulation on the routing variables for the problem. Chapter 3 explores the cooperation concept to solve a multi-depot inventoy-routing problem. In addition, the method is capable of finding new best solution for benchmark instances of the single-depot multi-vehicle inventory-routing problem.

Part two studies a different approach by presenting a second hybrid algorithm called a relax-and-price method. The fundamental idea is to combine the concepts of column generation with Lagrangian relaxation. Further, chapter 4 presents a set-covering type of formulation for the ILRP and develops a heuristic procedure based on relax-and-price. Chapter 5 adapts the method to solve the inventory-routing problem. Chapter 6 provides details of the pricing sub-problem that is embedded within the relax-and-price framework together with a computational study. This pricing problem is a consequence of the developed research and it is not intended to be the core of the thesis. It will be denoted as the Generalized Elementary Shortest Path Problem. General conclusions are discussed in chapter 7.

1.4. Introduction en français

Le sujet de cette thèse est le problème de localisation-routage avec de contraintes de stockage. Ce problème, nommé le ILRP (inventory - location - routing problem), considère le choix de localisation des dépôts en prenant en compte: 1) Quels seront les points de départ des tournées des véhicules afin de visiter et livrer un ensemble de détaillants; et 2) les décisions de gestion de stocks à chaque échelon de la chaîne d'approvisionamment sur un horizon de planification. Du point de vue de la recherche opérationnelle, le ILRP peut être décrit comme l'intégration des décisions de routage lors de la résolution d'un problème de conception de la chaîne logistique [2].

Les modèles et méthodes de résolution présentés appartiennent aux problémes d'optimisation combinatoire et font partie du domaine de la logistique du transport [7]. En plus, le ILRP suit la tendance de la recherche opérationnelle pour modéliser et résoudre des problèmes plus proches à la réalité d'une manière plus précise en satisfaisant les besoins industriels actuels.

D'un côté, les dernières contributions pour les problèmes de tournées avec gestion des stocks (inventory - routing) sont le résultat de la combination des connaissances concernant la gestion de stocks et le problème de tournées des véhicules [3]. De l'autre côté, les études de conception des réseaux ont aussi suivi le même chemin. Le problème de localisation-routage (LRP) est une variation du problème de localisation des dépôts dans lequel les clients doivent être liés aux dépôts par des tournées [15]. Des problèmes de conception des reseaux de transport en commun et distribution de produits sont exemples d'applications du LRP.

Le but du modèle étudié ici est de faire une extension du LRP [17] en considérant plusieurs périodes et des contraintes de stockage. Ce travail est inspiré de l'industrie pharmaceutique où les coûts de localisation sont du même ordre de grandeur que les coûts opérationels dans le long terme [8].

Le problème intègre des décisions qui sont considerées stratégiques, tactiques, et opérationnelles. Les décisions stratégiques sont faites généralement pour le long terme (plus d'un an). Les décisions tactiques considèrent en principe un horizon de planification à quelque mois et les décisions opérationnelles sont celles prises tous les jours et qui répondent aux besoins à court terme. Les trois sortes de décisions sont rarement mélangées du fait que les décisions opérationnelles sont facilement modifiables pendant que les décisions stratégiques le sont moins ou alors en payant des coûts importants. Néanmoins, le travail développé dans cette thèse ne refute pas ce principe.

En fait, la thèse expose des situations où la décision de localisation des dépôts n'est pas stratégique dans le sens décrit précédemment mais plutôt dans le sens hiérarchique (logiquement il faut connaître la localisation des dépôts avant d'élaborer des tournées). Les activités de logistique humanitaire et militaire sont des exemples pour lesquels un dépôt a souvent une position temporaire. Aussi, nous pouvons imaginer le cas où imposer la contrainte de ne livrer qu'un seul client par tournée n'est pas réaliste et les solutions fournies pourrait être sous-optimales. Finalement, pour un analyste de la chaîne logistique, la résolution du problème sert à faire l'évaluation de scénarios bien précis pour se faire une idée de la performance des solutions.

Ensuite, il est évident que plusieurs pressions pèsent sur la performance des systèmes logistiques. Il est possible d'identifier des raisons économiques, sociales, et environnementales.

Les élements économiques sont essentiels car les entreprises sont forcées à utiliser leurs ressources (matérielles, humaines et financières) d'une manière efficace. Une chaîne d'approvisionnement bien planifiée est celle qui profite complètement de la capacité de stockage des bâtiments et la capacité de livraison des véhicules pour minimiser les coûts logistiques.

Les élements sociaux sont aussi importants. Si on considère le cas d'une catastrophe comme un tremblement de terre, tsunami, inondation, etc, où des activités de logistique humanitaire doivent se mettre en place afin de faire la distribution d'eau potable, médicaments, et d'autre matériel de premier secours, avec des ressources limitées, faillir de faire la distribution de ces élements pourrait produire la mort ou des problèmes sérieux aux gents impliqués.

Finalement, il est reconnu que chaque produit consommé par la societé a un impact environnemental [6]. Les gouvernements et les utilisateurs des produits cherchent des produits "verts". Il est possible de justifier que cet objectif est en partie atteint par la minimisation des niveaux de stocks et les distances parcourues par les véhicules de livraison. Des tournées plus courtes produisent probablement moins d'émissions de CO_2 et des niveaux de stocks inférieurs ont besoin de moins d'énergie, par exemple dans le cas de produits surgelés. Dans le meilleur cas, l'optimisation de la logistique en considérant tous les aspects détaillés peut fournir des solutions pour une chaîne logistique durable.

Dans la littérature, la plupart des articles considèrent des problèmes similaires avec demande stochastique. Par exemple, Liu et Lee [10], Liu et Lin [11] proposent un modèle non-lineaire et étudient une approache connue comme "route-first locate-allocate second". Les quantités à livrer sont fixées a priori, permettant de résoudre un problème de tournées de véhicules à une seule période sous l'hypothèse que tous les clients seront visités avec la même fréquence. Cette hypothèse est cependant très restrictive car dans la pratique, les décisions de routage doivent être modifiées en fonction de la périodicité de visite aux clients car ces derniers ne sont tous pas forcément visités le même jour. De plus, le compromis entre les coûts de stockage et le coût de routage n'est jamais étudié. Les décisions de localisation des usines et plusieurs produits sont ajoutées par Reza Sajjadi et al.[18], mais les décisions liées à la gestion des stocks sont toujours fixées a priori.

Il y a aussi des auteurs qui présentent une variation du ILRP en considèrant des décisions de stockage que chez les détaillants et une demande stochastique [19]. Le coût de transport des fournisseurs aux dépôts est ajouté et ils proposent une approche différente qui cherche à estimer le coût de routage au lieu de calculer la valeur exacte. En plus, Ahmadi-Javid et al.[1] ajoute des décisions de capacité des dépôts, en considérant une politique de gestion de stocks de type (Q, r). Ainsi, les décisions de stockage sont faites avec une équation de type EOQ (modèle de Wilson) ajoutée à la fonction objectif.

Trois points faibles ressortent de la plupart des modèles. D'abord, il n'y a pas de liaison entre le coût de passation de commandes et le coût de transport, alors qu'ils sont généralement corrélés. Ensuite, la plupart des méthodes étudient un problème de tournées à une seule période en considérant que tous les clients seront visités avec la même fréquence. Or, la fréquence optimale d'approvisionnement pourrait être différente pour chaque client afin d'optimiser les coûts de routage et stockage en même temps. Donc, des approaches mono-période ne sont pas appropriées pour intégrer les décisions de routage et distribution. Finalement, il est évident que plusieurs modèles ont été proposés dans la littérature et il n'y a pas de convention pour traiter les trois problèmes simultanément. Nous pouvons conclure que si la version stochastique du ILRP trouvée dans la littérarure semble plus difficile que notre version déterministe, elle est en effet plus facile à résoudre car elle ne considère qu'une seule période pour le routage et les décisions de stockage sont limitées à un seul échelon de la chaîne logistique.

La contribution principale de cette thèse est de présenter deux formulations mathématiques et des nouvelles méthodes matheuristiques pour résoudre un problème complexe d'une manière efficace. Les algorithmes proposés fournissent des solutions de bonne qualité en des temps de calcul raisonnables. En plus, les méthodes heuristiques sont testés sur des sous-problèmes qui sont étudiés dans la littérature et fournissent des résultats intéresants. Ce sont les problèmes de tournées avec gestion de stocks dans le cas mono et multi dépôt; le problème de localisation-routage, et le problème de plus court chemin généralisé.

Comme recherche future, les modèles dévéloppés pourront être modifiés afin d'introduire plus d'attributs ou des conditions moins restrictives, par exemple en considérant que l'hypothèse d'avoir un système où la demande est connue n'est plus valable. Des techniques de modélisation plus sophistiquées, comme les modèles basés sur la programmation stochastique, sont capables de représenter la variabilité des paramètres et l'aspect dynamique des décisions. En effet, les décisions considérées sont prises par étapes. Ainsi, la décision de localisation est fixée dès le premier moment tandis que les décisions opérationelles sont faites selon l'état du système à chaque période, quand les informations de la demande deviennent connues, dans un cadre dynamique (voir Mendoza et Villegas[13], Sörensen and Sevaux[20] pour des applications de metaheuristiques pour le problème des tournées des véhicules avec demandes stochastiques).

Des extensions envisageables sont: 1) le cas multi-produit, 2) des livraisons fractionnées, 3) des fenêtres de temps [22], 4) une flotte de véhicules hétérogène, 5) des durées de parcours dépendant du temps, 6) des contraintes sur la durée totale des tournées, 7) des contraintes comme celles du problème de tournées sur les arcs [14], et 8) des systèmes de collecte et livraison [4, 21]. De plus, non seulement la fonction objectif peut être modifiée, comme par exemple, dans le problème de tournées de véhicules cumulatives (CCVRP) [16] pour trouver des solutions qui répondent mieux aux besoins de la logistique humanitaire, mais aussi en ajoutant plusieur objectifs [9], en minimisant simultanément: le coût de localisation, les coûts opérationnels, les indicateurs de risques, les émissions de CO_2; tout en maximisant les niveaux de service et le profit total.

Ce travail est composé de deux parties. Chacune est dédiée à présenter une méthode hybride différente pour résoudre le ILRP. La première partie introduit une approche coopérative entre une méthode exacte et des heuristiques de routage. Cette méthodologie est basée sur une formulation à trois indices pour les variables de routage. Le chapitre 2 étudie cette idée de coopération sur le ILRP. Le chapitre 3 est dédié à la résolution du problème de tournées avec gestion de stocks dans le cas multi-dépôt. De nouvelles meilleures solutions pour le problème de tournées avec gestion de stocks dans le cas mono-dépôt en considérant plusieurs véhicules ont aussi été trouvées par cette méthode basée sur la recherche locale.

La deuxième partie présente une approche differente pour le ILRP qui a été nommée "relax-and-price". L'idée fondamentale est d'intégrer les concepts théoriques de génération de colonnes avec la relaxation Lagrangienne. Donc, le chapitre 4 présente une formulation pour le ILRP de type de couverture d'ensemble et on développe ainsi une procedure heuristique basée sur la méthode "relax-and-price". Le chapitre 5 présente l'adaptation de la méthode pour résoudre le problème de tournées avec gestion de stocks et fournit la preuve mathématique que la méthode est toujours exécutée pour un nombre fini d'itérations. Le chapitre 6 montre en détail le sous-problème de "pricing" qui est utilisé dans le cadre de la méthode de "relax-and-price". Ce problème a été appelé le problème de plus court chemin généralisé. Les conclusions générales sont discutées dans le chapitre 7.

References

[1] Ahmadi-Javid, A., Azad, N., 2010. Incorporating location, routing and inventory decisions in supply chain network design. Transportation Research Part E: Logistics and Transportation Review 46(5), 582-597.

[2] Ambrosino, D., Scutellà, M-G., 2005. Distribution network design: New problems and related models. European Journal of Operational Research 165(3), 610-624.

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

[4] Berbeglia, G., Cordeau, J-F., and Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. TOP 15(1), 1-31.

[5] Brent, W., Travis T., 2008. A review of inventory management research in major logistics journals: Themes and future directions. The International Journal of Logistics Management 19(2), 212-232.

[6] Clift, R., 2004. Metrics for supply chain sustainability. In Technological Choices for Sustainability. Eds. Sikdar, S.K., Glavič, P., Jain, R., pages 239-253. Springer Berlin Heidelberg.

[7] Golden, B.L., Raghavan, S., Wasil, E.A., 2008. The vehicle routing problem: latest advances and new challenges. (Vol. 43). Springer.

[8] Guerrero, W.J., Yeung, T.G., Guéret, C., 2013. Joint-optimization of inventory policies on a multi-product multi-echelon pharmaceutical system with batching and ordering constraints. European Journal of Operational Research 231(1), 98-108.

[9] Jozefowiez, N., Semet, F., Talbi, E-G., 2008. Multi-objective vehicle routing problems. European Journal of Operational Research 189(2), 293-309.

[10] Liu, S.C., Lee, S.B., 2003. A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. The International Journal of Advanced Manufacturing Technology 22(11), 941-950.

[11] Liu, S.C., Lin, C.C., 2005. A heuristic method for the combined location routing and inventory problem. The International Journal of Advanced Manufacturing Technology 26(4), 372-381.

[12] Melo, M.T., Nickel, S., Saldanha-Da-Gama, F., 2009. Facility location and supply chain management–A review. European Journal of Operational Research, 196(2), 401-412.

[13] Mendoza, J., Villegas, J.G., 2013. A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. Optimization Letters 7(7), 1503-1516.

[14] Monroy, I.M., Amaya, C.A., Langevin, A., 2013. The periodic capacitated arc routing problem with irregular services. Discrete Applied Mathematics 161(4-5), 691-701.

[15] Nagy, G., Salhi, S., 2007. Location-routing: Issues, models and methods. European Journal of Operational Research 177(2), 649-672.

[16] Ngueveu, S.U., Prins, C., Wolfler-Calvo, R., 2009. An effective evolutionary algorithm for the cumulative capacitated vehicle routing problem. In Applications of evolutionary computing. Eds. M. Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G., Ekrt, A., Esparcia-Alczar, A., Farooq, M., Fink, A., Machado, P., volume 5484 of Lecture Notes in Computer Science, pages 778-787. Springer Berlin Heidelberg.

[17] Prodhon, C., 2011. A hybrid evolutionary algorithm for the periodic location-routing problem. European Journal of Operational Research 210(2), 204-212.

[18] Reza Sajjadi, S., Hossein Cheraghi, S., 2011. Multi-products location-routing problem integrated with inventory under stochastic demand. International Journal of Industrial and Systems Engineering 7(4), 454-476.

[19] Shen, Z-J.M., Qi, L., 2007. Incorporating inventory and routing costs in strategic location models. European Journal of Operational Research 179(2), 372-389.

[20] Sörensen, K., Sevaux, M., 2009. A practical approach for robust and flexible vehicle routing using metaheuristics and Monte Carlo sampling. Journal of Mathematical Modelling and Algorithms 8(4), 387-407.

[21] Velasco, N., Dejax, P., Guéret, C., Prins, C., 2012. A memetic algorithm for a pick-up and delivery problem by helicopter. Engineering Optimization 44(3), 305-325.

[22] Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2013. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Computers & Operations Research 40 (1), 475-489.

Part I.
First Hybrid Approach: Three-index formulation based heuristic

2. Hybrid Heuristic for the Inventory Location-Routing Problem with Deterministic Demand

W.J. Guerrero[1,2], C. Prodhon[1], N. Velasco[2], C.A. Amaya[2]

[1] ICD-LOSI (UMR-STMR CNRS 6279), Université de Technologie de Troyes,
12 rue Marie Curie, CS 42060 10004 Troyes Cedex, (France)
[2] PyLO, School of Engineering, Universidad de los Andes,
Cra 1 # 18A- 12, Bogotá, (Colombia)

The Inventory Location-Routing Problem with deterministic demand can be seen as an approach to both optimize a supply chain design and minimize its operational costs. This problem considers that vehicles might deliver products to more than one retailer per route and that inventory management decisions are included for a multi-depot, multi-retailer system with storage capacity over a discrete time planning horizon. The problem is to determine a set of candidate depots to open, the quantities to ship from suppliers to depots and from depots to retailers per period, and the sequence in which retailers are replenished by a homogeneous fleet of vehicles. A mixed-integer linear programming model is proposed to describe the problem and to provide bounds on the solutions. It is strengthened by two sets of valid inequalities with an analysis of their impact. Since the model is not able to solve the targeted instances exactly within a reasonable computation time, a hybrid method, embedding an exact approach within a heuristic scheme, is presented. Its performance is tested over three sets of instances for the inventory location-routing, location-routing and inventory-routing problems. Results show important savings achieved when compared to a decomposed approach and the capability of the algorithm to solve the problem.

Keywords: Location-Routing Problem, Inventory-Routing Problem, Matheuristics.

2.1. Introduction

The design of a supply chain is considered as a strategic level decision. It consists of identifying the optimal number of plants to open and their locations so that logistics costs are minimal. On the other hand, the management of a supply chain is usually to tackle tactical and/or operational decisions and it concerns the cooperation between facilities in order to obtain, transform, store and distribute materials, which also entails logistical costs [25]. Balancing strategic with operational objectives is the challenge.

Most of the facility location models consider distribution to be performed by dedicated routes, i.e. one vehicle visits one client at most (see Gebennini et al. [17]). However, in the case where orders are much smaller than vehicle capacity, this assumption is not longer true. The effects of ignoring routing decisions when locating depots is studied by Shen and Qi [43], Salhi and Rand [42]. When vehicles are not performing single-visit tours, locating depots, so that the sum of the distances between depots and retailers is minimized, is not an optimal solution. A more appropriate model is the one depicted in Location-Routing problems, that propose to optimize location decisions simultaneously with routing decisions. Examples are described by Prins et al. [35], Belenguer et al. [9] and a review is presented by Nagy and Salhi [30]. Nevertheless, these papers deal with the single period version or they simplify the multi-period problem by weighting the service to customers to be the same on each period of the horizon. Recently, Prodhon [38] solved a periodic version, but no inventory decision is managed.

Then, Miranda and Garrido [29] discuss the impact of ignoring inventory decisions when designing a supply chain. They conclude that the assignment scheme of retailers to depots has a direct impact on depot operation cost because ordering and holding costs might be significantly modified when the aggregated demand varies.

In addition, inventory and routing decisions are strongly interdependent [10]. Distribution and stock management decisions affect each other for two reasons: First, the set of minimal cost routes is built as a function of the quantities to deliver per period, which are determined by the inventory policies; and second, ordering costs required to design inventory policies include, among others, the transportation cost resulting from the choice of the sequence in which the retailers will be served. The optimal trade-off between inventory and distribution costs is known as the Inventory-Routing Problem (IRP) in Bertazzi et al. [11], and Andersson et al. [3].

Designing a supply chain becomes more complex if inventory and routing problems are included in the location decision-making. However, it is essential to balance short-term decisions with longer term thinking. As a result, for the Inventory Location-Routing Problem (ILRP), the resulting supply chain design includes an insight into detailed topics in order to decide how to satisfy future demand at minimum cost. Interest arises mainly from two contexts:

i) When a temporary location is required. It is the case for companies that strategically lease depots and pay rent. Consequently, they are more flexible and might conveniently change locations periodically. It is also the case for humanitarian missions managing disaster relief inventories [46, 7] with limited financial resources through donations. These activities are often performed for a short time. Further, in the field of military logistics, temporary location decisions are often made in order to distribute ammunition and other supplies. In all cases, location costs (e.g. rent) and operational costs (distribution and inventory holding) could have similar orders of magnitude.

ii) When long-term objectives require a supply chain design allowing different frequencies of replenishment for each retailer and distribution to be performed by vehicles capable of visiting more than one retailer per route. It is the case when assuming single period routing decisions (assuming routing to be the same every period) or dedicated routes (routes visiting a single retailer) is not realistic enough. The large retail sector or pharmaceutical and medical equipment supply are some examples. Again, depot opening costs should be scaled on the modeled horizon to be in balance with the operational costs. Furthermore, even if the future demand is not considered in the long-term, including inventory and routing costs allows incorporating within the location-allocation structure the effects of non-constant distribution activities and the effects of the interactions between inventory and routing decisions. Then, location decisions based on a set of routing scenarios (one per period) will perform outstandingly better on the long-run than one based on a single routing scenario.

Note that these applications suggest that demand might have an unpredictable nature while our model assumes known data. Our contribution is to solve the deterministic version of the problem in order to take the first step before solving a stochastic version with recourse. Even more, we also develop a decision-aid tool for "what-if" analysis. Think that an analyst might be interested in having better estimates of costs given the possibility of restructuring the supply chain under specific future demand assumptions.

Few papers simultaneously work on the three problems: depot location, vehicle routing and optimizing inventory policies. Table 1 summarizes a literature review on models and solution methods for the ILRP. Columns **Ret.** and **Dep.** denote if inventory decisions are made either at retailers, at depots, or both.

Authors	Demand	Ret.	Dep.	Model	Solution Method
Liu and Lee(2003)[21]	Stochastic	✓		non-linear	Route first-locate second
Liu and Lin(2005)[22]	Stochastic	✓		non-linear	sequential/improv. stage
Ambrosino and Scutellà(2005)[2]	Determ.	✓	✓	linear	commercial solver
Ma and Davidrajuh(2005)[24]	Stochastic	✓	✓	non-linear	sequential
Shen and Qi(2007)[43]	Stochastic		✓	non-linear	Branch-and-Bound
Ahmadi-Javid and Azad(2010)[1]	Stochastic		✓	non linear	tabu search Simulated Annealing
Mete and Zabinsky(2010)[27]	Stochastic		✓	non-linear	sequential Stochastic programming
Sajjadi and Cheraghi(2011)[41]	Stochastic	✓		non-linear	sequential/improv. stage

Table 1: A classification on combined Inventory-Location-Routing Problems and methods.

Most consider a single period routing, location decisions within a discrete set, demand splitting or backlogging not allowed and stochastic demand. The cost structure to be minimized comprises fixed opening costs for depots, expected holding and stock-out costs, and routing costs. Considering deterministic demand, Ambrosino and Scutellà [2] propose a linear model for the ILRP and show that for the single period case (LRP), the model implemented in CPLEX 7.0 is not able to find optimal solutions within 25 hours for instances with 13 depots and 95 retailers. For stochastic demand, Ma and Davidrajuh [24] propose an iterative sequential optimization approach where the problem is tackled as a series of sub-problems and never with a global perspective.

In addition, two different characterizations of this problem exist. First, some research papers tackle a LRP integrating in the objective function an EOQ-like component (Wilson model) aiming to minimize the expected inventory management cost at retailers, resulting in a non-linear model. The second approach fixes quantities to be delivered to retailers and optimizes inventory policies at depots instead.

This paper studies the ILRP as the issue of locating depots considering depot fixed opening costs, operational and tactical costs such as routing and stock management cost. The mathematical model and some valid inequalities are presented in section 2.2. Section 2.3 describes a hybrid heuristic and a computational study is presented in section 2.4. Conclusions are given in section 2.5.

2.2. Problem Definition

This paper tackles the design of a two-echelon supply chain considering both strategic and tactical/operational costs. This design comprises the location of the depots supplied by a factory and serving the deterministic demand of retailers, and the assignment of the latter to a depot over a given horizon. Each retailer is assigned to a single depot in the interest of facilitating monitoring and tracing of products. The costs include the depot opening costs, the delivery costs (dedicated routes to depots, non-dedicated to retailers) and the inventory costs at both depots and retailers, including an obsolescence penalty cost (that could be 0 or positive).

Formally, let J be a set of n retailers facing a deterministic non-constant demand $d_{jt}, \forall j \in J, \forall t \in H$, with t a period and $H = \{1, ..., p\}$ a discrete and finite planning horizon. Also, a set of m candidate depots I is available to replenish retailers. The ILRP is defined on a complete, weighted and directed graph $G = (V, A, C)$. $V = \{J \cup I\}$ is the set of nodes in the graph. C is the cost matrix c_{ij} associated with the traveling cost from node i to node j in the set of arcs A in the network. We consider a homogeneous unlimited fleet of vehicles, thus a set K of r ($r \geq n$) identical vehicles are available. Each node $i \in V$ is associated with a storage capacity W_i. Also, each depot $j \in I$ is associated to an opening cost O_j and ordering cost s_i (dedicated route from the factory or production cost). The vehicle capacity is Q units of product and the fixed cost of using a vehicle at least once in the planning horizon is F. Let B_i be the initial inventory at facility $i \in V$. $H_0 = \{0\} \cup H$ and $H' = H \cup \{p+1\}$ are horizons including a dummy period used to model initial and final conditions in inventory levels. The holding plus obsolescence penalty cost for one unit of product kept at node $j \in V$ from period $t \in H_0$ until period $l \in H'$ is q_{jtl}. Backlogging or stock-out are not allowed.

Let the decision variables be $y_i = 1$ iff depot $i \in I$ is opened; $f_{ij} = 1$ iff retailer $j \in J$ is assigned to depot $i \in I$, $x_{ijkt} = 1$ iff the arc $(i, j) \in A$ is crossed from i to j by vehicle $k \in K$ on period $t \in H$, T_i be the maximum number of vehicles used from depot $i \in I$ over H. Inventory decisions at echelon e are denoted by the variable w^e. The quantity replenished from depot i to retailer j in period t to satisfy the demand in period l using the vehicle k is denoted by w^2_{ijtlk} (the superscript 2 denotes inventory decisions for the second echelon). The quantity of product used from initial stock at retailer j to satisfy demand in period $t \in H'$ is denoted by w^2_{j0t}. At the first echelon, $z_{li} = 1$ iff depot $i \in I$ is replenished in period $l \in H$, 0 otherwise. The quantity to replenish in depot $i \in I$ that is delivered in period $t \in H$ to satisfy the demand in period $l \in H'$ is w^1_{itl} (the superscript 1 denotes the first echelon). Then, the ILRP model can be stated as follows:

$$\begin{aligned} \min \sum_{i \in I} \left(O_i y_i + F T_i + \sum_{l \in H} s_i z_{li} \right) + \sum_{i \in I} \sum_{t \in H_0} \sum_{l=t|l>0}^{p+1} q_{itl} w^1_{itl} + \sum_{t \in H'} q_{j0t} w^2_{j0t} + \\ \sum_{i \in I} \sum_{j \in J} \sum_{t \in H} \sum_{l=t}^{p+1} \sum_{k \in K} q_{jtl} w^2_{ijtlk} + \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} \sum_{t \in H} c_{ij} x_{ijkt} \end{aligned} \quad (1)$$

Subject to:

$$\sum_{i\in I}\sum_{k\in K}\sum_{l=1}^{t} w^2_{ijltk} + w^2_{j0t} = d_{jt} \;\; \forall j \in J, \forall t \in H \tag{2}$$

$$\sum_{l=0}^{t} w^1_{ilt} = \sum_{k\in K}\sum_{l=t}^{P+1}\sum_{j\in J} w^2_{ijtlk} \;\; \forall i \in I, \forall t \in H \tag{3}$$

$$\sum_{t\in H'} w^2_{j0t} = B_j, \;\; \forall j \in J \tag{4}$$

$$\sum_{t\in H'} w^1_{i0t} = B_i \cdot y_i, \;\; \forall i \in I \tag{5}$$

$$\sum_{r=0}^{t}\sum_{l=t}^{P+1} w^1_{irl} \leq W_i \cdot y_i \;\; \forall i \in I, \;\; \forall t \in H \tag{6}$$

$$\sum_{l=t}^{P+1}\left(w^2_{j0l} + \sum_{r=1}^{t}\sum_{k\in K} w^2_{ijrlk} \right) \leq W_j \;\; ,\forall i \in I, \;\; \forall j \in J, \;\; \forall t \in H \tag{7}$$

$$\sum_{l=t}^{P+1} w^1_{itl} \leq W_i \cdot z_{ti} \;\; \forall i \in I, \;\; \forall t \in H \tag{8}$$

$$\sum_{i\in I} f_{ij} = 1, \;\; \forall j \in J \tag{9}$$

$$f_{ij} \leq y_i, \;\; \forall j \in J, \;\; \forall i \in I \tag{10}$$

$$\min(Q, W_j) \cdot \sum_{u\in J} x_{iukt} \geq \sum_{l=t}^{p+1} w^2_{ijtlk} \;\; \forall i \in I, \; \forall j \in J, \forall t \in H, \forall k \in K \tag{11}$$

$$\min(Q, W_j) \cdot \sum_{u\in J\cup\{i\}} x_{ujkt} \geq \sum_{l=t}^{p+1} w^2_{ijtlk} \;\; \forall i \in I, \; \forall j \in J, \forall t \in H, \forall k \in K \tag{12}$$

$$\sum_{j\in V} x_{ijkt} - \sum_{j\in V} x_{jikt} = 0, \;\; \forall t \in H, \;\; \forall i \in V, \forall k \in K \tag{13}$$

$$\sum_{i\in V}\sum_{k\in K} x_{ijkt} \leq 1 \;\; \forall \;\; t \in H, \;\; \forall j \in V \tag{14}$$

$$\sum_{i\in V}\sum_{k\in K} x_{jikt} \leq 1 \;\; \forall \;\; t \in H, \;\; \forall j \in V \tag{15}$$

$$\sum_{i\in I}\sum_{j\in J} x_{ijkt} \leq 1 \;\; \forall \;\; t \in H, \;\; \forall k \in K \tag{16}$$

$$\sum_{k\in K}\sum_{j\in J} x_{ijkt} \leq T_i \;\; \forall t \in H, \;\; \forall i \in I \tag{17}$$

$$\sum_{i\in I}\sum_{l=t}^{p+1}\sum_{j\in J} w^2_{ijtlk} \leq Q \;\; \forall k \in K, \;\; \forall t \in H \tag{18}$$

$$\sum_{u\in J} x_{iukt} + \sum_{u\in V\setminus\{j\}} x_{ujkt} \leq 1 + f_{ij} \;\; \forall i \in I, \;\; \forall j \in J, \;\; \forall t \in H, \;\; \forall k \in K \tag{19}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijkt} \leq |S| - 1 \;\; \forall t \in H, \;\; \forall k \in K, \;\; \forall S \subseteq J \tag{20}$$

$$x_{ijkt} \in \{0,1\} \;\; \forall i \in I, \;\; \forall j \in J, \;\; \forall t \in H, \;\; \forall k \in K \tag{21}$$

$$y_i \in \{0,1\} \;\; \forall i \in I \tag{22}$$

$$z_{it} \in \{0,1\} \;\; \forall i \in I, \;\; \forall t \in H \tag{23}$$

$$f_{ij} \in \{0,1\} \;\; \forall i \in I, \;\; \forall j \in J \tag{24}$$

$$T_i \in \mathbb{N} \;\; \forall i \in I \tag{25}$$

$$w^2_{ijtlk} \in \mathbb{R}^+ \;\; \forall i \in I, \;\; \forall j \in J, \;\; \forall t \in H, \;\; \forall l \in H'|l \geq t, \;\; \forall k \in K \tag{26}$$

$$w^2_{j0t} \in \mathbb{R}^+ \;\; \forall j \in J, \;\; \forall t \in H' \tag{27}$$

$$w^1_{itl} \in \mathbb{R}^+ \;\; \forall i \in I, \;\; \forall\, t \in H_0, \;\; \forall\, l \in H|l \geq t \tag{28}$$

The objective function equation (1) is the sum of the opening costs, ordering costs at depots, the costs of using a vehicle at least once, holding costs at depots and retailers with the distribution costs. Constraints (2) force the satisfaction of the demand at every retailer. Inventory flow conservation through echelons is forced by constraints (3). The sum over the horizon of the quantity kept on stock from period zero up to period $p + 1$ is equal to the initial stock as stated by constraints (4)-(5). Capacity of depots is guaranteed by (6) meaning that satellite depots are not cross-docking points. Retailers are capacitated as shown in equations (7). Ordering decisions at depots are forced by constraint set (8). Each retailer must be allocated to a single opened depot as stated by equations (9)-(10). Constraints (11)-(12) guarantee that if a retailer is replenished on period l with route k, it must be visited accordingly. If the triangle inequality is not guaranteed, then visits without replenishment should be forbidden by adding the constraints (29):

$$\sum_{i \in V} x_{ijkt} \leq \sum_{i \in I} \sum_{l=t}^{p+1} w^2_{ijtlk} \quad \forall j \in J, \forall t \in H, \forall k \in K \tag{29}$$

Concerning distribution, constraints (13)-(20) force a feasible routing. First, traditional vehicle flow conservation constraints are presented in equations (13)-(15). Note that equations (16) force each vehicle to perform one route per period at the most. This is a common assumption in most vehicle routing problems. In real-life, vehicles could perform several trips per day; this problem is known as the multi-trip vehicle routing problem [13]. In that case, duration constraints are included to limit the time each vehicle works per day. A more sophisticated approach than the one presented, where vehicles are allowed to perform multi-trips in order to reduce the fleet size cost, such that the duration of the tasks scheduled to vehicles remains feasible, is future research.

Further, equations (17) link the cost of using vehicles with the routing decisions. Vehicles limited capacity is forced by equations (18). The set of equations (19) state that a retailer j can be linked to a depot i only if j is assigned to depot i ($f_{ij} = 1$). Finally, equations (20) are standard subtour elimination constraints and constraints (21)-(28) state the nature of the decision variables.

This model can be enforced by additional inequalities. However, the ones for the traveling salesman problem (TSP) or vehicle routing problems (VRP) are not valid in this case because quantities to deliver and retailers to visit are decision variables. Inequalities presented for the IRP are not valid either since: i) the holding cost is considered to be time-dependent to include seasonal effects and obsolescence penalty costs, ii) there is a multi-depot environment, iii) the depots do not have a fixed location. Nonetheless, two valid inequalities for the ILRP are presented.

Theorem 1. *The inequalities (30) are valid for the ILRP with deterministic demand.*

$$min\,(W_i, Q) \cdot \sum_{k \in K} \sum_{j \in J} x_{ijkt} \geq w^1_{ilt} \quad \forall i \in I, \ \forall l \in H_0, \ \forall t \in H | l \leq t \tag{30}$$

Proof. If a depot i is replenished at any period l to satisfy demand on period t - i.e., $w^1_{ilt} > 0$ - then:

- At least $\frac{w^1_{ilt}}{Q}$ vehicles must depart from depot i in period t to satisfy such a demand - i.e., $\sum_{k \in K} \sum_{j \in J} x_{ijkt} \geq \frac{w^1_{ilt}}{Q}$, if the capacity of the vehicle is tight (i.e., $min\,(W_i, Q) = Q$)
- At least one vehicle must depart from depot i in period t to satisfy such a demand – i.e., $\sum_{k \in K} \sum_{j \in J} x_{ijkt} \geq \frac{w^1_{ilt}}{W_i}$, if the vehicle capacity constraint is loose (i.e., $min\,(W_i, Q) = W_i$).

□

Theorem 2. *The inequalities (31) are valid for the ILRP with deterministic demand.*

$$\sum_{i \in V} \sum_{k \in K} \sum_{l=1}^{t} x_{ijkl} \geq \left\lfloor \frac{\sum_{l=1}^{t} d_{jl} - B_j}{Q} \right\rfloor, \ \forall\, j \in J, \ \forall\, t \in H \tag{31}$$

Proof. The set of constraints (31) require that the minimal number of times a retailer is visited up to period t equals the total demand that can not be satisfied with the initial inventory ($\sum_{l=1}^{t} d_{jl} - B_j$) divided by the vehicle capacity. □

2.3. Hybrid Heuristic

Exact procedures can only solve the model for very small sized instances within a reasonable computation time (as will be shown in section 5). Thus, heuristic methods seem to be a more suitable alternative to find high quality solutions on larger instances. The proposed framework is based on this kind of approach and tries to solve subproblems, not considering several decision levels in independent phases, but exchanging information when moving between solution spaces.

It should be emphasized that most of the previous ILRP models consider an inventory policy (mainly EOQ) with ordering costs that are modeled independently from distribution, while in real life it depends on the routing performed by vehicles. In addition, the embedded Supply Chain Design Problem (SCDP) with estimated distribution costs (neglecting the routing construction) might be solved to optimality using commercial solvers in reasonable computation time. Taking advantage of this last property, the problem is decomposed into decisions that are computed by exact methods and the ones obtained heuristically. Then, the suggested pattern makes exact and heuristic procedures cooperate and this leads to a hybrid heuristic that can be seen as a matheuristic [40]. Thus, the ILRP resolution induces:

- A supply chain design S fully described by three elements: i) the set I of the depots to be opened, ii) F the array indicating, for each retailer, its assigned depot, iii) W the $(m+n) \times (p+1) \times (p+1)$ matrix in which each element w_{itl} indicates, for each facility $i \in V$, the quantity of product arriving in period $t \in H_0$ that will remain in stock until period $l \in H'$.

- A routing evaluation $R_{(S)}$ as the set of routes indicating the sequence in which retailers will be replenished on each period for a given supply chain structure S.

Note that a single structure S might have several feasible sets of routes $R_{(S)}$. In the following subsections, the main components of the approach will be described in order to subsequently assemble the proposed algorithm.

2.3.1. Supply chain design

Assume the $m \times n \times p$ matrix C^* to be known in which each element c^*_{ijt} represents the cost of delivering product from depot $i \in I$ to retailer $j \in J$ in period $t \in H$. It is an estimated sum of distribution cost and ordering costs from depots to retailers. Then, the MIP presented in section (2.2) could be modified to obtain a SCDP.

Decision variables x_{ijkt} would be replaced by $\hat{x}_{ijt}$, $\forall\, i \in I$, $\forall\, j \in J$, $\forall\, t \in H$ representing a binary variable equal to 1 iff depot i replenishes the retailer j in period t, 0 otherwise. w^2_{ijtlk} is replaced by $\hat{w}^2_{ijtl}$ $\forall\, i \in I$, $\forall\, j \in J$, $\forall\, t \in H$, $\forall\, l \in H'$ representing the quantity replenished by depot i in period t to stock until period l at retailer j. Accordingly, the objective function (1) would be replaced by:

$$\min \sum_{i \in I} \left(O_i y_i + FT_i + \sum_{l \in H} s_i z_{li} \right) + \sum_{i \in I} \sum_{l \in Ho} \sum_{t \in H'} q_{jtl} w^1_{itl} + \sum_{t \in H'} q_{j0t} w^2_{jot} + \\ \sum_{i \in I} \sum_{j \in J} \sum_{t \in H} \sum_{l \in H'} q_{jtl} \hat{w}^2_{ijtl} + \sum_{i \in I} \sum_{j \in J} \sum_{t \in H} c^*_{ijt} \hat{x}_{ijt} \tag{32}$$

The index k is easily removed from constraints (2),(3), (7), (18), and (26). Equations (4)-(6),(8)-(10), (21)-(25), (27), and (28) remain unchanged. The following constraints are added to complete the SCDP formulation.

$$\min(Q, W_j) \cdot \hat{x}_{ijt} \geq \sum_{l=t}^{p+1} \hat{w}^2_{ijtl} \ \forall i \in I, \ \forall j \in J, \forall t \in H \tag{33}$$

$$f_{ij} \geq \hat{x}_{ijt} \ \forall \ i \in I, \ \ t \in H, \ \ \forall j \in J \tag{34}$$

$$\sum_{j \in J} \sum_{l=t}^{p+1} \hat{w}^2_{ijtl} \leq T_i \cdot Q \ \forall t \in H, \ \ \forall i \in I \tag{35}$$

Equations (33) relate distribution activities to inventory flow from depot i to retailer j on period t. Equations (34) forbid replenishment from i to j if retailer j is not allocated to depot i. Constraint (35) states that the minimum number of vehicles T_i to use from depot i times the vehicle capacity Q is larger than the total quantity to replenish. To sum up, S is a partial solution for the ILRP and it might be computed to optimality by solving the presented MIP model. This supply chain design generator will be denoted by the acronym SCDP for simplicity.

Since the distribution cost depends on the embedded routing which can only be solved once the quantities to deliver per period are known, the initial matrix C^* is estimated to be a random fraction of the direct delivery cost. Then, each element $c^*_{ijt} = \xi_1 \cdot c_{ij} \ \forall i \in I, \ \forall j \in J, \ \forall t \in H$, where ξ_1 is a uniform random variable $\sim Unif\,[\alpha, \ 2 \cdot \alpha]$. The parameter α is fixed a priori.

C^* is updated every time feasible routing costs are computed. This update is performed through equations (36) with the information of a feasible ILRP solution and represents the cost of detour of the route. If no replenishment is made from depot i to retailer j on period t ($\sum_{l \in H} \hat{w}^2_{ijtl} = 0$), c^*_{ijt} remains unchanged.

$$c^*_{ijt} = \sum_{u \in V} \sum_{k \in K} (c_{uj} x_{ujkt} + c_{ju} x_{jukt}) - \sum_{u \in V} \sum_{v \in V} \sum_{k \in K} c_{uv} x_{ujkt} x_{jvkt}, \quad \text{if} \sum_{l \in H} \hat{w}^2_{ijtl} > 0 \tag{36}$$

For example, consider a feasible solution where at period t, the arcs (u, j) and (j, v) are traversed by a vehicle that departed from depot i. In this particular example, the updated cost for c^*_{ijt} is equal to the cost of making a detour to visit retailer j, which is $c_{uj} + c_{ju} - c_{uv}$. In this context, the estimation of C^* is modified every time feasible routes are computed (by the routing operators). By doing so, it is expected C^* to be a better input for the supply chain design generator (SCDP solver). This is how the routing operator will cooperate with subsequent supply chain design generations.

2.3.2. Randomized routing heuristic

Even when the supply chain S is designed, the remaining routing decisions $R_{(S)}$ are difficult to solve since the problem reduces to the well known VRP for every period and every depot. Extensive research is being proposed to solve this NP-hard problem [44, 18].

In addition, to reinforce the search, it is better to consider the presolved assignment as a means to provide a subset of promising depots to open and then to tackle a multi-depot VRP per period, and even an LRP. A simple heuristic procedure that provides good solutions for the LRP is the RECWA, the randomized extended Clarke and Wright algorithm, implemented as in Prins et al. [37]. It is an extension for the multi-depot case of the Clarke and Wright saving's algorithm [14] (see Mendoza et al. [26] for results on the single-depot version). A randomization on the selected merge allows some diversification over the iterations. However, retailers allocation to depots must be the same along the horizon, which is not warranted by solving an LRP per period and requires a repairing operator.

The repairing operator evaluates for each retailer, the assigned depots during the planning horizon. Retailers are evaluated in decreasing order of their total demand. The most frequent depot allocation is fixed for each retailer if capacity constraints hold. A randomized Clarke and Wright algorithm is then performed for each depot, for each period while fixing the allocation decisions.

2.3.3. Local search

To improve the routing and the inventory on the global solution, a local search (LS) is used. The hierarchy and description of the neighborhoods explored in our LS are:

- *Move:* The visit of a retailer is shifted from its current position to a different position within the same route or to other routes departing from the same depot at the same period.

- *Swap:* The positions of two different retailers are exchanged. Both exchanged visits must share depot and period, and might or might not be in the same route.

- *2-Opt:* Two non-consecutive arcs are removed from the solution and new arcs are included to assure feasibility of the solution. The removed arcs might or might not belong to the same route but they must share the same depot and period.

- *Shift delivery date:* A single retailer is removed from the solution and new inventory policies are designed. Its first delivery date is shifted to the earliest possible date. On the shifted first delivery date, the retailer will be replenished with its maximum storage capacity or the maximum available capacity to minimize any chances of stockout. Replenishment in subsequent periods is decided analogously if stock is not sufficient to satisfy future demands. The retailer with the modified inventory policy is re-inserted in the solution and is allocated to the same depot as before. Routing costs are evaluated by best insertion procedure. The new inventory policy for the retailer also affects the inventory holding cost of the corresponding depot. If quantity on stock at the depot is not enough to satisfy the new demand, it is increased accordingly. Figure 3 presents an example in which the first delivery date of retailer **B** (at period 2) is shifted to the earliest possible date (period 1 assuming that the depot has available capacity). Delivery at period 3 is shifted to period 2 and at period 3, shipment is not longer required. Note that the maximum number of vehicles might be reduced (it does for this specific example) and the inventory policies for retailer B have changed, so the inventory policies for the depot have to be reviewed. We have preferred to shift the first delivery date only in order to perform fast computations. This neighborhood might be extended to any delivery date increasing the complexity of the operator.

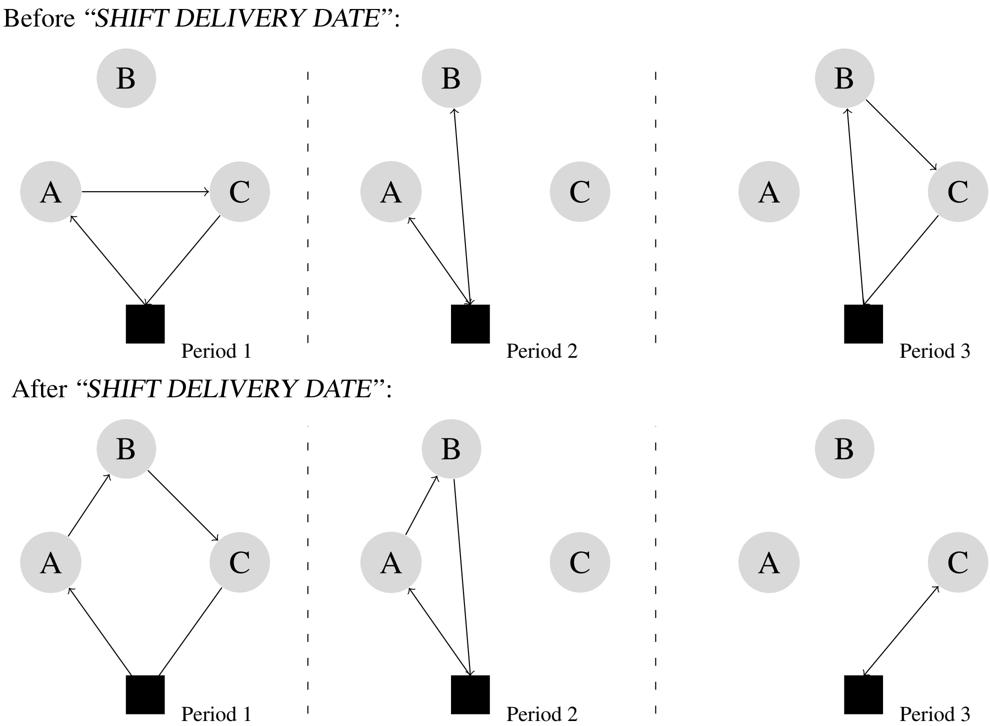

Figure 3: *Shift delivery date* example

- *2-shifted delivery date:* Same as *Shift delivery date* but considering a couple of customers sharing the same depot. This movement aims to synchronize the deliveries of retailers over time.

- *Depot reallocation:* The allocation of a retailer is shifted to another depot. Inventory policies at the retailer remain the same in this case. The set of scheduled visits over the planning horizon is inserted into the routes departing from the new depot in the given period. A best insertion cost is computed. Depot capacity over the complete planning horizon must be taken into consideration.

- *Depot allocation swap:* Two retailers allocated to different depots are exchanged in their depot allocation. The set of scheduled visits over the planning horizon is removed from the solution. Without changing inventory policies for the retailer (schedule visits remain unchanged), a best insertion procedure is implemented. Depot inventory policies must be re-evaluated if available stock is not sufficient to supply the new assigned retailer.

Our LS is embedded on a variable neighborhood descent (VND) structure [19]. It evaluates each neighborhood and when an improving movement is found, the search starts again from the first neighborhood. If no improving movement is found in the current neighborhood, the search continues with the next one. LS applies the first movement that improves the solution except for *2-shifted delivery date* that uses best-improving movement strategy.

Furthermore, LS requires an important computational effort. It could be performed with some probability $\pi_{LS} \leq 1$ to make the algorithm run faster.

2.3.4. Intensification

With the purpose of evaluating the interactions between inventory and routing decisions, an intensification component is proposed. Algorithm 1 presents how to re-evaluate inventory-routing decisions with a dedicated procedure. Similar to a Large-Neighborhood Search [34], inventory-routing decisions are destroyed and repaired. Thus, for n_{inten} iterations, a dynamic lot-sizing problem (DLSP) is first solved with a MIP solver (line 3). This problem consists of determining the optimal quantities to stock at the depots and retailers if location-allocation decisions are fixed. It is called "dynamic" because demand is allowed to vary over periods, not because decisions are re-optimized dynamically [45]. The MIP presented in section (2.3.1) is reduced by fixing y and f variables (location-allocation decisions) with the values in the supply chain design S at the current solution.

Algorithm 1 . Procedure: $Intensification(C^*, S, R_{(S)})$

1: $n_p = 0, \mathscr{L} = \{\}, \mathscr{L}' = \{\}, C_{best} = \infty$
2: **for** $k_3 = 1$ to n_{inten} **do**
3: $\quad S \leftarrow$ DLSP($C^*, S, \mathscr{L}'$)
4: $\quad R_{(S)} \leftarrow$RCWA(S)
5: $\quad$LocalSearch($S, R_{(S)}$)
6: $\quad$**if** C($S, R_{(S)}$)$< C_{best}$ **then**
7: $\quad\quad C_{best} = C(S, R_{(S)})$
8: $\quad\quad S_{best} \leftarrow S$
9: $\quad\quad R_{best} \leftarrow R_{(S)}$
10: $\quad$**else**
11: $\quad\quad n_p = n_p + 1$
12: $\quad\quad$**if** $n_p = n_0$ **then**
13: $\quad\quad\quad$createTabuLists($\mathscr{L}, \mathscr{L}'$)
14: $\quad\quad\quad C^* \leftarrow$perturbation($C^*, \mathscr{L}, \mathscr{L}'$)
15: $\quad\quad\quad n_p = 0$
16: $\quad\quad$**end if**
17: $\quad$**end if**
18: **end for**

Once the inventory policies for retailers and depots have been repaired, routing decisions are recomputed. A randomized version of the Clarke and Wright algorithm is proposed to build $R_{(S)}$ (line 4). The solution is improved by the local search detailed in section (2.3.3) (line 5). The function $C(S, R_{(S)})$ returns the cost of a solution S given routes $R_{(S)}$.

To avoid local optima and diversify initial S at line 3, a perturbation procedure is applied (line 13) every n_p iterations without improvement. The first perturbation consists on randomly selecting a retailer and its nearest neighbor. Both are included into a list $\mathscr{L}$. The procedure is repeated until $\mathscr{L}$ has more than 10% of the elements of J. For a single random period t, the delivery cost from j to its corresponding depot i will be $C^*_{ijt} = 0 \ \ , \forall\, j \in \mathscr{L}$. This strategy targets the synchronization of deliveries in a given period t for retailers with close proximity. Additionally, cuts are added to the MIP by forcing the solution to visit all retailers j at a random period t_j, $\forall j \in \mathscr{L}'$. $\mathscr{L}'$ is generated as a random and independent subset of J.

2.3.5. Post-optimization

Given the best solution $(S, R_{(S)})$ found by the hybrid heuristic, a dedicated post-optimization procedure to intensify allocation-routing decisions is proposed in the form of an iterated local search (ILS) [23]. The pseudo-code of the procedure is presented in algorithm 2. In line 3, S and $R_{(S)}$ are mutated by modifying the allocation of a fixed percentage γ of randomly selected retailers. Without changing their inventory policies, they are reassigned to a different random depot that is already open and has available capacity. If reallocation is not feasible for the set of opened depots, a new random depot is opened. A best insertion procedure is implemented to include the visits within $R_{(S)}$ over the planning horizon. In line 4, the local search procedure is applied. If the current solution improves the best solution found, the best solution is updated at lines 5-9. If the opposite is true, the current solution is discarded and mutation is repeated on the best solution. The procedure is repeated for up to N_1 improving iterations or up to N_2 iterations without improvement of the best solution. In the worst case, $N_1 + N_2$ mutations and local search procedure calls are performed.

2.3.6. Algorithm Overview

The components described in sections 2.3.1 to 2.3.4 are integrated in the multi-start hybrid heuristic reinforced by the ILS procedure presented in section 2.3.5. Algorithm 3 details the complete procedure. At line 7, a supply chain design S is computed by solving a SCDP linear model using a commercial solver as explained in section 2.3.1. The components of the initial solution constructed by this operator are the location-allocation and inventory decisions. Then, the randomized routing heuristic detailed in section 2.3.2 is performed to optimize the routing decisions $R_{(S)}$ (line 8) and to potentially improve allocation decisions. With some probability π_{LS} the local search procedure is applied. Routing, inventory and allocation decisions are potentially improved at this step. Subsequently, the update of the C^* matrix is performed (line 12), as explained in section (2.3.1).

Algorithm 2 . Procedure: $ILS(S, R_{(S)})$

$i = 1$ and $j = 1$
while $i \leq N_1$ and $j \leq N_2$ **do**
  mutate$(S, R_{(S)})$
  LocalSearch$(S, R_{(S)})$
  if C$(S, R_{(S)})$< C_{best} **then**
    $C_{best} = C(S, R_{(S)})$
    $S_{best} \leftarrow S$
    $R_{best} \leftarrow R_{(S)}$
    $i = i + 1$
  else
    $S \leftarrow S_{best}$
    $R_{(S)} \leftarrow R_{best}$
    $j = j + 1$
  end if
end while

The intensification procedure is called in line 14 to potentially improve inventory-routing decisions (see section 2.3.4) in multi period context. Lines 17-21 update the best found solution S_{best} and R_{best}. A new solution is explored (lines 6 to 22) until no improvement is perceived or a maximum of MAX_{it} iterations are performed.A tabu list τ is created at line 23 to limit the search of the supply chain design generator. τ is used in the next call of the SCDP procedure, where a new solution S is forced to close the depot belonging to the tabu list. τ is cleared once the procedure SCDP is performed. The post-optimization procedure described in section 2.3.5 is performed in the multi-depot context in lines 25-28. Every decision component is fixed except for allocation-routing which might be improved by this operator.

2.4. Computational Study

The algorithms were coded in language Mosel and solved with Xpress-IVE 7.0, 64-bits. Tests are performed on an Intel Xeon with 2.80Ghz processor and 12 GB of RAM.

2.4.1. Instances

Since there are no available benchmark instances for the problem under consideration, 20 ILRP instances were randomly generated. They have the following features: $m : \{5\}$ depots, $n : \{5, 7, 15\}$ retailers, $p : \{5, 7\}$ periods. The names of the instances correspond to its size. They are labeled as $m - n - p - x$ where m indicates the number of depots, n the number of retailers, p the number of periods and x is used to itemize and differentiate instances with the same size ($x \in \{a, b, c, ...\}$).

Demand at retailer j for period t is generated with a Normal distribution: $d_{jt} \sim N(\mu_j, \sigma_j)$, were $\mu_j \in [5, 15]$ and $\sigma_j \in [0, 5]$. The opening costs for depots O_i are generated randomly with a Normal distribution with parameters (μ_i,σ_i) chosen from the set of pairs $\{(1000, 20), (5000, 100), (8000, 300)\}$ while the replenishment cost s_i is chosen from the set $\{100, 500\}$.

Algorithm 3 : Main Algorithm (Overview)

$S, R(S) \leftarrow 0$
$S_{best}, R_{best} \leftarrow 0$
$C_{best} = \infty$; $\tau \leftarrow \emptyset$
for $k_1 = 1$ to m **do**
  $C^* \leftarrow random \cdot C$
  while (Improvement or less than MAX_{it} iterations) **do**
    $S \leftarrow$ SCDP(C^*, τ)
    $R_{(S)} \leftarrow$ RECWA(S)
    if $random < \pi_{LS}$ **then**
      LocalSearch($S, R_{(S)}$)
    end if
    Update($C^*, R_{(S)}$)
    if $p > 1$ **then**
      $Intensification(C^*, S, R_{(S)})$
      Update($C^*, R_{(S)}$)
    end if
    if $C_{best} > C(S, R_{(S)})$ **then**
      $C_{best} = C(S, R_{(S)})$
      $S_{best} \leftarrow S$
      $R_{best} \leftarrow R_{(S)}$
    end if
  end while
  $\tau \leftarrow$ RandomDepotClosure(S)
end for
if $m > 1$ **then**
  $S \leftarrow S_{best}$
  $R_{(S)} \leftarrow R_{best}$
  $ILS(S, R_{(S)})$
end if

The coordinates (X_i, Y_i) for facility $i \in V$ are randomly generated in a square of size 100×100. The function NINT$(\cdot)$ approximates to the closest integer value. Transportation cost c_{ij} =NINT$(100 \cdot \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2})$. Vehicle capacity Q is a random value = $5 \cdot b$ where b is a random integer in the interval $[3, 15]$. The cost of using a vehicle F is selected from the set $\{350, 1000, 5000\}$. Depot capacity W_i is randomly generated in the interval $[D/3, D]$, where $D = \sum_{j \in J} \sum_{t \in H} d_{jt}$. Retailer's capacity W_j are randomly generated in the interval $[g_j, 3 \cdot g_j]$ where $g_j = max_t \ \{d_{jt}\}$. Initial inventories B_j were chosen from the set $\{0, d_{j1}\}$ for retailers and B_i from the set $\{0, 10 \cdot D/n\}$ for depots. Inventory holding costs for a single period $t \in Ho$ at retailers and depots $j \in V$, $q_{j,t,t+1}$ are generated in the interval [0.03,0.50]. The inventory holding costs for k periods as $q_{j,t,t+k} = \sum_{l=t}^{t+k-1} q_{j,l,l+1} + k \cdot \xi_2$, where ξ_2 represents the unitary obsolescence penalty cost. it is generated as $\xi_2 \sim Unif[0.01, 0.02]$.

Further, larger instances for classical subproblems are used to test the validity of the algorithm. 18 LRP instances with capacitated vehicles, 5 capacitated depots and 20, 50 and 100 retailers and single period available in $http://prodhonc.free.fr$ were used to test our hybrid heuristic. They are also solved by Prins et al. [37]. Likewise, 40 "order-up to level" benchmark instances for the IRP with single depot, single vehicle, 5 to 40 customers, three periods and holding cost between $[0.01, 0.05]$ are considered. They are available at http://www-c.eco.unibs.it/∼bertazzi/abls.zip . Our model is slightly adapted to an "order-up to level" policy required on these IRP instances. A set of constraints forcing the replenishment of a retailer j so that the stock raises up to W_j if visited is included.

2.4.2. MIP solver performance and valid inequalities impact analysis

Bounds for the ILRP are computed by solving the MIP presented in section 2.2 using a commercial solver before solving it using the proposed hybrid heuristic. This section is intended to show the impact of the valid inequalities proposed by theorems 1 and 2. Four configurations of the solver are proposed. V.0 corresponds to the MIP presented in section (2.2). V.1 corresponds to V.0 with equations (30). V.2 corresponds to V.0 with equations (31). Finally, V.3 corresponds to the complete strengthened formulation composed by V.0 plus equations (30) and (31). Despite the quality of the final solution, the presolve procedures and the preliminary heuristics of the commercial solver were deactivated here to isolate the effect of adding the valid inequalities. Best found feasible solution (UB) within a time limit of 8200 seconds with default settings and percent gap at 20, 100, 500, 5000 and 8200 seconds are reported in table 2 for a subset of instances to be brief. Besides, we are not able to guarantee the quality of the lower bound provided by the solver resulting from the continuous relaxation of the model given its variations due to the new valid inequalities among the four versions. Then, a fair comparison might be made by comparing the gap of all versions to the same bound value.

It is possible to analyze, for example, on instance 5-5-7-a that V.2. is the fastest to provide (within 100 seconds) a feasible solution that is 69.24% larger than UB. Not adding any valid inequality would result in having within 500s the first feasible solution that is 53.78% larger than UB. V.2 computes the best solution among the compared versions within 5000s and 8200s. V.1 and V.3 are outperformed by V.0 and V.2 in this case. Results show that any version dominates systematically. On one hand, adding both sets of constraints does not seem to speed up the process. On the other, excluding them all (V.0) often provides feasible (but low quality) solutions on short computing times. In any case, the computational burden is significant considering that the instances are smaller than potential real-life instances. A more sophisticated method to dynamically add valid inequalities in a Branch and Cut frame is future research. In subsequent research, we decided to keep both the additional valid inequalities and presolve procedures.

2.4.3. Preliminary tests for the hybrid heuristic

The parameters to calibrate are described next:

- α is used to compute C and to build an initial solution by the SCDP procedure as explained in section 2.3.1. Preliminary tests indicate that α should be between 0.2 and 0.4.

- MAX_{it} represents a limit on the iterations performed in the first phase of the algorithm. The considered levels are 4 and 100. For less than 4 iterations, solution quality is not acceptable. 100 iterations is equivalent to iterating until no improvement is achieved.

Instance	UB	Version	GAP/UB (%) 20s	100s	500s	1000s	5000s	8200s
5-5-5-a	93718.5	V.0	-	94.31	71.06	67.84	62.82	43.94
		V.1	117.56	90.60	35.17	35.17	35.16	35.16
		V.2	83.88	83.88	68.32	68.32	29.75	29.75
		V.3	-	-	79.88	72.63	46.81	39.18
5-5-5-b	62494.5	V.0	-	42.69	39.42	39.42	34.49	16.67
		V.1	-	57.93	53.66	46.36	23.63	16.42
		V.2	74.76	74.76	43.46	38.79	35.03	30.57
		V.3	-	29.38	28.85	28.85	28.85	28.85
5-5-5-c	69760.3	V.0	-	110.10	77.08	70.68	58.11	58.11
		V.1	109.93	89.24	78.69	73.81	65.91	54.89
		V.2	-	-	76.47	76.47	54.83	54.83
		V.3	-	-	81.29	79.97	64.81	60.38
5-5-7-a	77404.1	V.0	-	-	53.78	53.78	48.01	48.01
		V.1	-	-	-	133.90	57.56	57.56
		V.2	-	69.24	69.24	69.24	38.70	38.70
		V.3	-	-	86.30	65.16	60.66	58.35
5-5-7-b	110940	V.0	-	-	147.89	147.89	88.48	88.48
		V.1	-	-	90.25	90.25	90.25	89.10
		V.2	-	-	-	-	128.55	128.55
		V.3	-	-	-	-	-	132.23
5-5-7-c	94150.2	V.0	-	61.95	50.90	50.43	37.01	37.01
		V.1	-	55.09	42.96	42.96	26.05	26.05
		V.2	-	57.39	24.08	24.08	24.08	24.08
		V.3	-	82.69	65.76	65.76	57.51	53.26

Table 2: Impact measure of valid inequalities

- π_{LS}: the probability of performing a local search procedure in the first phase of the algorithm. The considered levels were 40% and 100%.
- N_1 and N_2 represent the number of iterations with and without improvement respectively. They are used in the post-optimization procedure. These are tested considering levels 15 and 30 independently.

The minimum number of instances required for the tuning test N is such that the acceptable standard error (SE) is larger than SD$\cdot\sqrt{1/N}$, where SD stands for the standard deviation of the gap to UB in the sample as proposed by Cobb [15]. An approximate confidence interval for the performance of our algorithm is in the form: Mean Gap $\pm 2 \times SE$. It has about 95% chance of containing the true value for this gap. We consider an acceptable SE to be below 0.25%. In a preliminary test, the maximum estimated standard deviation for a combination of parameters SD$= 1.44$. That is, at least 33 instances are required to build acceptable intervals.

We account for 32 candidate combinations of parameters considering 5 parameters with 2 levels each. The one providing the most stable results and competitive computing times was selected. We ranked for each instance (15 for the ILRP and 18 for the LRP), the best gap and best CPU independently. The non-parametric test of Friedman proved that at least two combinations of parameters in the sample represent populations with different mean ranks (p-value$_{GAP}$ = 0.56, p-value$_{CPU}$ = 0.76). Instead of choosing the parameter combination with minimal GAP or CPU, the combination with the best trade-off between mean rank for gaps and mean rank for CPU is chosen. This approach provides a more stable algorithm, as the chances of obtaining acceptable quality solutions are less variable. Be aware that deeper research on parameter tuning might lead to lower average gaps. For the following analysis: $\alpha = 0.4$, $MAX_{it} = 4$, $\pi_{LS} = 100\%$, and $N_1 = N_2 = 15$.

2.4.4. Results

Table 3 presents, for small ILRP instances, the comparison between 3 heuristics: 1) A time-constrained commercial solver with a time limit of 2.5 hours; 2) Our hybrid heuristic; and 3) a sequential heuristic (H1) that aims to emulate the traditional sequential approach. H1 is equivalent to compute a supply chain design using the commercial solver (fixing location-allocation decisions) and to make inventory-routing decisions through the procedure described in section 2.3.4. In detail, column UB presents the best feasible solution found by the solver within the time limit. Columns two and three present the gap between the solution found in 60 and 500 seconds and UB. Column five (CPU UB) presents the time when UB was found by the solver in seconds. Columns six to eight present the average results of our heuristic in three runs. The average solution cost, the average gap to UB (GAP) and average computation time in seconds (CPU) are reported respectively. Columns nine and ten present the gap to UB and CPU for H1.

On average, our hybrid heuristic outperforms the commercial solver preset as a truncated search by 0.52% with an average computing time of 457s. 6 out of 15 new best solutions are found, and other 4 solutions have a gap to UB inferior to 0.4%. When compared to the solutions computed on a similar average computation time (500s), our method outperforms the solver by 5.66% and improves the solutions of 10 out of 15 instances. Besides, the only interest of H1 is its speed. The traditional approach (H1) provides solutions that are about 2.62% more expensive than UB found by the solver and more than 3% higher than our hybrid heuristic.

5-7-5-c is a difficult instance. The solver was not able to find a feasible solution within the preset time limit. In fact, the first feasible solution (UB) has a cost of 176191.5 computed in 6.7 hours (24180 s). The traditional approach computes a solution of 149459.2 (-15.17% lower) in 1408.8s. Furthermore, the solution found by our hybrid heuristic has a cost 21.1% inferior to UB computed almost 8 times faster.

Similarly, table 4 compares the ILRP instances with 15 clients to the commercial solver with a time limit of 9 hours. The solver was able to find integer solutions for only 3 out of 5 instances. After presolve procedures, instance 5-15-5-c has 49 775 variables and 32 562 constraints. Instance 5-15-5-e has 45 525 variables and 77 488 constraints. Columns four to six present the average cost, gap to UB and average computational time of our heuristic in three runs per instance. The proposed heuristic solved every instance with an average time of 2.33 hours. Furthermore, we improved the average BKS in 37.52%. On the other hand, even if H1 solved every instance in less than 1.4 hours,

	SOLVER				HYBRID HEURISTIC			H1		
Instance	GAP (60s)	GAP (500s)	UB	CPU UB	COST	GAP (%)	CPU (s)	COST	GAP (%)	CPU (s)
5-5-5-a	3.7	0	93718.5	173	93625.3	-0.10	25.2	93625.3	-0.10	4.7
5-5-5-b	10.5	0.88	62494.5	2176	62206.9	-0.46	22.4	62897.7	0.65	4.0
5-5-5-c	23.9	0.34	69760.3	1356	70881.0	1.61	48.0	70964.2	1.73	7.9
5-5-5-d	3.9	0.01	93801.2	592	93451.2	-0.37	28.6	93451.2	-0.37	4.7
5-5-5-e	4.3	0	93851.0	212	94600.6	0.80	14.1	97788.7	4.2	4.3
5-5-7-a	9.7	2.95	77404.1	1093	70966.5	-8.32	320.7	75969.1	-1.85	89.3
5-5-7-b	50.8	25.07	110940.0	4243	107478.5	-3.12	394.9	112299.6	1.23	46.7
5-5-7-c	28.9	28.88	94150.2	8163	94152.9	0.00	328.3	100416.6	6.66	212.0
5-5-7-d	13.8	0.02	87744.2	3951	87744.2	0.00	62.1	91750.1	4.57	8.8
5-5-7-e	-	1.67	69025.9	1819	67275.4	-2.54	176.4	71017.7	2.89	44.8
5-7-5-a	110.1	1.53	68485.2	873	69739.3	1.83	55.0	71522.9	4.44	7.9
5-7-5-b	-	3.09	76339.1	1317	78662.5	3.04	114.8	79122.5	3.65	51.0
5-7-5-c	-	-	-	-	138998.0	-	4964.9	141696	-	1409
5-7-5-d	-	7.52	99988.9	7432	100001.0	0.01	222.3	106584.2	6.60	26.2
5-7-5-e	48.7	0.01	62010.1	1404	62234.0	0.36	81.2	63451.9	2.33	11.5
Average	28.0	5.14	82836.7	2486	86134.5	-0.52	457.3	88819.2	2.62	128.8

Table 3: CPU times and average gap for HH, H1 and solver for random ILRP instances.

solutions are 3.91% higher than the hybrid heuristic.

High robustness is also achieved by our algorithm. The square coefficient of variation for the solution value is always inferior of $1.0e^{-3}$ for every instance tested. Therefore, there is little interest in executing more runs of the algorithm for the same instance.

	SOLVER		HYBRID HEURISTIC			H1		
INSTANCE	UB	CPU UB	COST	GAP (%)	CPU (s)	COST	GAP (%)	CPU (s)
5-15-5-a	156959.0	26726	113434.3	-27.73	1863.5	124376.3	-20.76	101.8
5-15-5-b	233359.1	22055	172743.3	-25.98	2001.3	177696.7	-23.85	443.6
5-15-5-c	-	-	210333.0	-	14301.9	212849.0	-	4076.9
5-15-5-d	403222.5	28844	165939.7	-58.84	1530.7	176477.0	-56.23	330.2
5-15-5-e	-	-	228467.7	-	22661.3	236799.7	-	4880.7
Average	264513.5	25875	178183.6	-37.52	8471.8	185639.7	-33.61	1966.6

Table 4: CPU and average gap for HH, H1 and solver for large ILRP instances.

Table 5 sums up the results on classical LRP instances. Average results for instances with 5 depots, 20 retailers (20R-5D-1P), 50 retailers (50R-5D-1P) and 100 retailers (100R-5D-1P) are presented. Column ($\sharp$) presents the number of instances per data set. The average of best known solution (BKS) in each set are taken from $http://prodhonc.free.fr$. The average cost (cost), gap to BKS (gap) and computation time (CPU) of our hybrid heuristic for three runs and the minimum and maximum gap ($\%_{min}$, $\%_{max}$) to BKS are shown in columns 4-8 respectively. They show an av-

			Hybrid Heuristic [1]					[37][2]		
Instances	♯	BKS	cost	gap	CPU(s)	$\%_{min}$	$\%_{max}$	cost	gap	CPU(s)
20R-5D-1P	4	45087	45154	0.13	0.97	0.10	0.19	45144	0.10	0.17
50R-5D-1P	8	74113	74782	0.68	4.69	0.26	1.23	74701	0.81	2.14
100R-5D-1P	6	199165	201957	1.37	28.7	1.04	1.74	202210	1.55	22.1
Average		109346	110590	0.79	11.9	0.48	1.17	110636	0.90	8.35

1 Intel Xeon with 2.8Ghz processor and 12 GB of RAM

2 Pentium 4 with 2.4Ghz processor and 512MB of RAM

Table 5: Benchmark on Location-Routing problem instances.

erage gap of 0.79% computed in 11.9 s. The average gap to BKS is always between 0.48% and 1.17%. We compare our methodology with the dedicated method of Prins et al. [37] (coded in C++ and executed on a Dell OPTIPLEX GX260 PC, 512MB of RAM, with a Pentium 4 processor clocked at 2.4 GHz and Windows XP) detailed in columns 9-11. Even if it is not dedicated to the LRP, our method is competitive with the algorithm of Prins et al. [37]. We choose the latter for two main reasons: i) it provides very good results and ii) it allows a fair comparison since it uses also the Clarke and Wright algorithm.

Results are also competitive on IRP instances as shown in table 6. The number of retailers is shown in column (♯) and the average optimal solution in column $z*$. Results of our heuristic for average, minimum, and maximum gap, and total computation time and computation time to best solution (CPU$_{best}$) for three runs are presented in columns five to eight, showing an average gap of 2.58% computed in 349.36s. Results for percentage gap to optimal solution of Archetti et al. [4] and Bertazzi et al. [11] are presented in columns nine and ten. Corresponding computation times for Archetti et al. [4] (not available for Bertazzi et al. [11]) are in column 11 (CPU) using an Intel Dual Core 1.86 GHz and 3.2 GB RAM and coded in C++. The approach computes solutions of intermediate quality between Archetti et al. [4] and Bertazzi et al. [11] within computation times that are similar, even when it is not a dedicated method.

			Hybrid Heuristic [1]					Benchmark [2]		
INSTANCE	♯	z*	cost	gap (%)	CPU (s)	$\%_{min}$	$\%_{max}$	Archetti et al. (2011)	Bertazzi et al. (2002)	CPU (s)
5R-1D-3P	5	1418.7	1418.7	0	3.0	0	0	0	2.88	3.0
10R-1D-3P	5	2228.7	2236.2	0.34	7.8	0	0.49	0	0.78	12.8
15R-1D-3P	5	2493.5	2520.4	1.08	28.9	0.03	2.44	0	2.56	41.4
20R-1D-3P	5	3053.0	3160.2	3.51	82.6	1.71	6.61	0.02	3.83	104.2
25R-1D-3P	5	3451.1	3532.9	2.37	191.1	0.38	4.71	0	2.99	258.8
30R-1D-3P	5	3643.2	3731.0	2.41	413.2	0.94	4.81	0.02	3.60	515.0
35R-1D-3P	5	3846.9	4016.9	4.42	812.0	2.29	7.16	0.04	4.46	808.8
40R-1D-3P	5	4125.7	4393.9	6.50	1256.4	5.4	7.7	0.06	6.46	1168.6
Average	5	3032.6	3126.29	2.58	349.36	1.35	4.24	0.02	3.45	364.08

1 Intel Xeon with 2.8Ghz processor and 12 GB of RAM

2 Intel Dual Core 1.86 GHz processor and 3.2 GB RAM

Table 6: Benchmark on Inventory-Routing problem instances

It should, however, be noted that different computers and programming languages were used when comparing algorithms. We have provided this information from benchmark for guidance only since a reliable comparison of computing times is difficult. Even if it were possible to obtain an estimation of the MFlops for each computer (see for example Dongara [16]) to evaluate each speed factor, it is assumed full exploitation of parallelism. Our algorithms do not satisfy this assumption and we cannot guarantee that benchmark algorithms do it. Further, the used languages differ as well. Nonetheless, the proposed hybrid algorithm executes within reasonable time for benchmark subproblems.

2.5. Conclusions

We present the combined Inventory-Location-Routing Problem (ILRP) as an approach to supply chain design considering inventory management and routing cost in order to overcome the fact that traditional approaches decompose decisions and often provide sub-optimal solutions. We consider a discrete and finite planning horizon and a two-echelon supply chain. Assumptions include a homogeneous fleet of vehicles, and deterministic, non-constant demand. Seasonal holding costs and obsolescence penalty costs are additional (but not restrictive) features of our model. Decisions that must be taken simultaneously are: 1) location decisions of depots, 2) inventory decisions at both echelons of the supply chain, 3) allocation decisions of retailers to depots, and 4) multi-period routing decisions.

We propose a hybrid approach to solve a supply chain design problem with estimated distribution costs using exact methods while the remaining routing decisions are computed by heuristic procedures. By alternating between decisions spaces and information sharing, the algorithm manages to optimize globally the components of the problem without oversimplifying it. Results for randomly generated instances show significant cost savings over the traditional approach and efficient computation when compared to commercial solvers. The ILRP reduces to the LRP and the IRP under certain conditions. Our tests show a robust performance over larger benchmark instances for both the LRP and the IRP.

Future research comprises the ILRP with two routing decision levels as in the 2E-LRP and the ILRP considering maritime transportation constraints.

Acknowledgements

This research is partially supported by Champagne-Ardenne Regional Council (France) and Centro de Estudios Interdisciplinarios Básicos y Aplicados - CEIBA (Colombia). We thank two anonymous reviewers for their valuable and constructive suggestions to this paper.

Chapter 2 was published in:
W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya (2013) *Hybrid heuristic for the inventory location-routing problem with deterministic demand*. International Journal of Production Economics 146(1): 359-370.

www.sciencedirect.com/science/article/pii/S0925527313003381

Preliminary results were presented at:
W.J. Guerrero (2013) Heurísticas para el problema combinado de Localización y Ruteo de Inventarios. XVII ELAVIO – Escuela Latino-Iberoamericana de Verano en Investigación Operativa Valencia (España).

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya (2013) Hybrid Heuristic for the Inventory Location-Routing Problem. Seminario PyLO. Universidad de los Andes, Colombia. May 15th (Invited Talk)

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya (2012) Hybrid Heuristic for the Inventory Location-Routing Problem. CLAIO-SBPO Congreso Latino Americano de Investigación Operativa, Sept 24-28, Rio de Janeiro, Brasil. (Invited Talk)

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya.(2012) Intégration d'une gestion de stocks lors de la résolution coopérative du problème de Localisation-Routage . Congrès ROADEF 2012, Angers, France.

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya.(2012) Hybrid Heuristic for the Inventory Location-Routing Problem with Deterministic Demand . 5th International Workshop on freight transportation and Logistics. ODYSSEUS. May 21-25, Mykonos, Greece

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya (2011) Hybrid Heuristic for the Inventory Location-Routing Problem. LOSI SEMINAR. Institut Charles Delaunay. Université de Technologie de Troyes. November.(Invited Talk)

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya.(2011) A Matheuristic for the Inventory Location Routing Problem with Deterministic Demand. MIC 2011 (Metaheuristics International Conference), Udine, Italie.

W.J. Guerrero, N. Velasco, C.A. Amaya , C. Prodhon. (2011) The Inventory Location Routing Problem with Deterministic Demand. POMS Annual Conference, 2011.

2.6. Appendix

2.6.1. Notation for mathematical formulation

Sets

- I Set of candidate depots
- J Set of retailers
- $V = \{I \cup J\}$ Set of facilities
- H Set of periods in the planning horizon
- $H_0 = H \cup \{0\}$ Planning horizon including initial conditions
- $H' = H \cup \{p+1\}$ Planning horizon including final conditions
- K Set of available vehicles
- A Set of arcs connecting facilities

Parameters

- n Number of retailers
- p Number of periods
- m Number of candidate depots
- r Number of available vehicles
- G Graph defining the ILRP. $G = (V, A, C)$
- C Cost matrix where c_{ij} is the traveling cost for arc $(i, j) \in A$ [$]
- W_i Storage capacity at facility $i \in V$ [units]
- O_i Opening cost [$]
- s_i Ordering cost for depot [$]
- Q Vehicle capacity [units]
- F Cost for using a vehicle at least once over H [$]
- B_i Initial inventory [units]
- q_{itl} Unitary holding cost at facility i from period t up to period l [$/unit]
- d_{jt} Demand at retailer $j \in J$ in period $t \in H$ [units]

Decision Variables

- y_i Binary variable. It is equal to 1 iff depot $i \in I$ is opened
- f_{ij} Binary variable. It is equal to 1 iff retailer $j \in J$ is allocated to depot $i \in I$
- x_{ijkt} Binary variable. Equal to 1 iff the arc $(i, j) \in A$ is crossed from i to j by vehicle $k \in K$ on period $t \in H$
- T_i Maximum number of vehicles allocated to depot i

- w^1_{itl} Quantity held in stock at depot i (1st echelon) from period t up to period l [unit]
- w^2_{j0t} Quantity held in stock at retailer i(2nd echelon) from initial period 0 up to period l [unit]
- w^2_{ijtlk} Quantity delivered by depot i with vehicle k held in stock at retailer j(2nd echelon) from period t up to period l [unit]
- z_{ti} Binary variable. It is equal to 1 iff depot i is replenished at period t

2.6.2. Notation for Heuristic Procedure

- $S = (I, F, W)$ Supply chain design
- I set of selected depots to open
- F Depot assignment for each retailer
- W Matrix of dimensions $(m + n) \times (p + 1) \times (p + 1)$ in which each element w_{itl} indicates, for each facility $i \in V$, the quantity of product arriving in period $t \in H_0$ that will remain in stock until period $l \in H'$ [unit]
- $R_{(S)}$ Set of selected routes in the solution
- C^* Matrix with estimated delivery cost from depots to retailers (assignment costs) [$]
- $\hat{x}_{ijt}$ Binary Decision variable indicating whether depot i supplies retailer j at period t
- $\hat{w}^2_{ijtl}$ Decision variable indicating the quantity supplied by depot i , held in stock at retailer j from period t to period l [unit]
- ξ_1, ξ_2 Uniform random variables
- α Parameter to estimate initial distribution costs
- π_{LS} Probability of performing a local search procedure in the first phase of the algorithm
- N_1 Parameter of maximum number of improving iterations
- N_2 Parameter of maximum number of iterations without improvement
- MAX_{it} Parameter to limit on the iterations performed in the first phase of the algorithm
- $\mathscr{L}, \mathscr{L}'$ tabu lists to synchronize retailer visits and diversify initial S

2.7. Résumé en français

La conception de la chaîne logistique est une décision stratégique. Elle consiste à identifier l'ensemble optimal de dépôts ou usines à ouvrir et leurs emplacements afin de minimiser les coûts logistiques. D'autre part, la gestion de la chaîne d'approvisionnement comporte généralement des décisions tactiques et / ou opérationnelles, et elle concerne la coopération entre les établissements afin d'obtenir, transformer, stocker et distribuer des produits, ce qui entraîne également des coûts logistiques [25]. Par conséquent, le défi est de trouver l'équilibre entre les objectifs stratégiques et opérationnels sur un horizon de planification.

La plupart des modèles traitant ce problème considère la distribution faite par des véhicules livrant un seul client au maximum. Toutefois, dans le cas où la taille des commandes à distribuer est beaucoup plus petite que la capacité du véhicule, cette hypothèse n'est plus valable. Les travaux de la littérature préliminaires [43, 42] étudient les effets d'ignorer les décisions de routage lors de la localisation des dépôts. Lorsque les véhicules ne font pas une seule visite, la localisation des dépôts en minimisant la somme des distances entre les dépôts et les détaillants, n'est pas une solution optimale. Un modèle plus approprié est celui représenté par les problèmes de localisation-routage. Ceux-ci proposent d'optimiser les décisions d'emplacement de dépôts simultanément avec les décisions de routage. Des exemples sont décrits par Laporte et al. [20], Prins et al. [36, 37, 35], Belenguer et al. [9], Nguyen et al. [31] et une révision de la littérature est faite par Nagy and Salhi [30]. Néanmoins, tous ces articles traitent la version à une seule période où ils simplifient le problème multi-période, en pondérant la demande des clients afin d'être la même pour chaque période de l'horizon. Récemment, Prodhon and Prins [39], Prodhon [38] et Pirkwieser and Raidl [33] résolvent une version périodique, mais aucune décision de gestion de stocks n'est gérée.

De la même manière, Miranda and Garrido [28] montrent l'impact des décisions en ignorant la gestion de stocks lors de la conception d'une chaîne d'approvisionnement. Ils concluent que l'affectation des détaillants aux dépôts a un impact direct sur le coût de fonctionnement du dépôt car il est en fonction de la demande agrégée des détaillants correspondants. De plus, ils remarquent que les stocks de sécurité dans les dépôts, la fréquence à laquelle les commandes sont faites aux fournisseurs et les coûts de possession peuvent être modifiées largement lorsque la demande agrégée varie.

D'ailleurs, les décisions de gestion de stocks et de routage sont fortement interdépendantes [10]. Considérons un ensemble de véhicules qui part de dépôts pour approvisionner un ensemble de détaillants. Les activités de distribution et les décisions de gestion des stocks aux deux échelons s'influencent mutuellement pour deux raisons: D'abord, l'ensemble des routes de coût minimal pour visiter les détaillants est construit en fonction des quantités à livrer par période, qui sont déterminées par les politiques de gestion de stocks. Deuxiement, les coûts de passation de commande nécessaires pour concevoir ces politiques de gestion de stocks comprennent, entre autres, les frais de transport dépendant du choix de la séquence dans laquelle les détaillants seront servis. Le problème d'optimisation du compromis entre les coûts de possession de stocks et les coûts de distribution est connu comme le problème de tournées avec gestion de stocks (Inventory Routing Problem - IRP) dans Andersson et al. [3], Archetti et al. [4, 5], Bertazzi et al. [11], Oppen et al. [32], Zhao et al. [47]. Le compromis entre les coûts de production, les coûts de possession des stocks au dépôts et les coûts de distribution est étudié dans Armentano et al. [6], Bard and Nananukul [8], Boudia and Prins [12] appelé le problème intégré de production-distribution (IPDP).

La conception d'une chaîne d'approvisionnement qui optimise les coûts logistiques globaux devient plus complexe si la gestion de stocks et de routage sont inclus dans le processus de prise de la décision. Cependant, il est essentiel d'équilibrer les décisions à court terme dans un cadre d'optimisation pour le long terme. En conséquence, la conception de la chaîne logistique a besoin d'avoir une idée des activités dans un niveau opérationnel détaillé afin de décider comment satisfaire la demande future avec un coût minimal.

Cet article étudie la gestion de stocks intégré lors de la résolution du problème de localisation-routage. Nous considérons la décision d'emplacement d'un ensemble de dépôts appartenant à un ensemble de candidats. Ainsi, nous cherchons à minimiser les coûts d'ouverture des dépôts, et les coûts opérationnels tels que le routage et la gestion de stocks. Ce problème a été appelé Inventory-Location-Routing Problem (ILRP). Nous posons l'hypothèse de connaître la demande de manière déterministe. Une révision de la littérature est présentée dans la section 2.1. Le modèle mathématique et des inégalités valides sont présentés dans la section 2.2. La section 2.3 décrit une heuristique hybride et les résultats sont présentés dans la section 2.4. Les conclusions sont données dans la section 2.5.

Nous proposons une approche hybride qui résout un problème de conception de la chaîne logistique en estimant des coûts de distribution par l'intermédiaire de méthodes exactes. Itérativement elle optimise les décisions de routage avec des méthodes heuristiques. En alternant entre les espaces de décision et le partage d'information, l'algorithme arrive à optimiser globalement les composantes du problème sans trop le simplifier. Le tableau 3 présente, pour les instances du ILRP, la comparaison entre trois méthodes heuristiques: 1) Un solveur commercial limité en temps d'exécution à 2.5 heures, 2) Notre heuristique hybride, et 3) une heuristique séquentielle (H1) qui vise à reproduire l'approche traditionnelle. H1 est équivalent à concevoir la chaîne d'approvisionnement en utilisant le solveur commercial afin de fixer les décisions d'emplacement-affectation et d'optimiser la gestion de stocks et de routage avec l'algorithme décrit dans la section 2.3.4.

Les résultats sur des instances générées aléatoirement montrent la supériorité sur l'approche traditionnelle et des temps de calcul plus efficaces par rapport aux solveurs commerciaux. Le ILRP peut être réduit aux problèmes de localisation-routage et de tournées avec gestion de stocks, sous certaines conditions. Nos tests montrent une bonne performance sur des instances de la littérature de plus grande taille pour ces sous-problèmes.

Les recherches futures comprennent le ILRP avec deux niveaux de décisions de routage, comme dans le problème de localisation-routage à deux échelons (2E-LRP) et le ILRP avec des contraintes pour le transport maritime.

References

[1] Ahmadi-Javid, A., Azad, N., 2010. Incorporating location, routing and inventory decisions in supply chain network design. Transportation Research Part E: Logistics and Transportation Review 46 (5), 582–597.

[2] Ambrosino, D., Scutellà, M.-G., 2005. Distribution network design: New problems and related models. European Journal of Operational Research 165 (3), 610–624.

[3] Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A., 2010. Invited review: Industrial aspects and literature survey: Combined inventory management and routing. Computers & Operations Research 37, 1515–1536.

[4] Archetti, C., Bertazzi, L., Hertz, A., Speranza, M., 2011. A hybrid heuristic for an inventory routing problem. INFORMS Journal on Computing, Articles in Advance, 1–16.

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

[6] Armentano, V., Shiguemoto, A., Lokketangen, A., 2011. Tabu search with path relinking for an integrated production distribution problem. Computers & Operations Research 38 (8), 1199–1209.

[7] Balcik, B., Beamon, B., Krejci, C., Muramatsu, K., Ramirez, M., 2010. Coordination in humanitarian relief chains: Practices, challenges and opportunities. International Journal of Production Economics 126, 22–34.

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

[9] Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., Wolfler-Calvo, R., 2011. A branch-and-cut method for the capacitated location-routing problem. Computers & Operations Research 38 (6), 931 – 941.

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

[11] Bertazzi, L., Paletta, G., Speranza, M., 2002. Deterministic order-up-to level policies in an inventory routing problem. Transportation Science 36 (1), 119–132.

[12] Boudia, M., Prins, C., 2009. A memetic algorithm with dynamic population management for an integrated production-distribution problem. European Journal of Operational Research 195 (3), 703–715.

[13] Brandão, J., Mercer, A., 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. European Journal of Operational Research 100 (1), 180–191.

[14] Clarke, G., Wright, J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12 (4), 568–581.

[15] Cobb, G. W., 1998. Introduction to Design and Analysis of Experiments (Textbooks in Mathematical Sciences). Springer.

[16] Dongara, J. J., 2013. Performance of various computers using standard linear equations software. Tech. Rep. CS-89-85, Electrical Engineering and Computer Science Dept., University of Tennessee., Computer Science and Mathematics Division., University of Manchester.
URL `http://www.netlib.org/benchmark/performance.pdf`

[17] Gebennini, E., Gamberini, R., Manzini, R., 2009. An integrated production-distribution model for the dynamic location and allocation problem with safety stock optimization. International Journal of Production Economics 122 (1), 286–304.

[18] Golden, B., Raghavan, S., Wasil, E., 2008. The Vehicle Routing Problem: Latest Advances and New Challenges. Vol. 43 of Operations Research/Computer Science Interfaces. Springer US.

[19] Hansen, P., Mladenovic, N., 2003. Variable neighborhood search. In: Glover, F., Kochenberger, G. (Eds.), Handbook of Metaheuristics. Vol. 57 of International Series in Operations Research & Management Science. Springer New York, pp. 145–184.

[20] Laporte, G., Nobert, Y., Taillefer, S., 1988. Solving a family of multi-depot vehicle routing and location-routing problems. Transportation Science 22 (3), 161–172.

[21] Liu, S., Lee, S., 2003. A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. The International Journal of Advanced Manufacturing Technology 22, 941–950.

[22] Liu, S., Lin, C., 2005. A heuristic method for the combined location routing and inventory problem. The International Journal of Advanced Manufacturing Technology 26, 372–381.

[23] Lourenço, H., Martin, O., Stützle, T., 2003. Iterated local search. In: Glover, F., Kochenberger, G. (Eds.), Handbook of Metaheuristics. Vol. 57 of International Series in Operations Research & Management Science. Springer New York, pp. 320–353.

[24] Ma, H., Davidrajuh, R., 2005. An iterative approach for distribution chain design in agile virtual environment. Industrial Management & Data Systems 105 (6), 815–834.

[25] Melo, M., Nickel, S., da Gama, F. S., 2009. Facility location and supply chain management: A review. European Journal of Operational Research 196 (2), 401–412.

[26] Mendoza, J. E., Medaglia, A. L., Velasco, N., 2009. An evolutionary-based decision support system for vehicle routing: The case of a public utility. Decision Support Systems 46 (3), 730–742.

[27] Mete, H. O., Zabinsky, Z. B., 2010. Stochastic optimization of medical supply location and distribution in disaster management. International Journal of Production Economics 126 (1), 76–84.

[28] Miranda, P., Garrido, R., 2004. Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. Transportation Research Part E: Logistics and Transportation Review 40(3), 183–207.

[29] Miranda, P., Garrido, R., 2009. Inventory service-level optimization within distribution network design problem. International Journal of Production Economics 122 (1), 276–285.

[30] Nagy, G., Salhi, S., 2007. Location-routing: Issues, models and methods. European Journal of Operational Research 177 (2), 649 – 672.

[31] Nguyen, V.-P., Prins, C., Prodhon, C., 2012. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. European Journal of Operational Research 216 (1), 113–126.

[32] Oppen, J., Løkketangen, A., Desrosiers, J., 2010. Solving a rich vehicle routing and inventory problem using column generation. Computers & Operations Research 37, 1308–1317.

[33] Pirkwieser, S., Raidl, G., 2010. Variable neighborhood search coupled with ILP-based very large neighborhood searches for the (periodic) location-routing problem. In: M. Blesa, C., G.Raidl, A.Roli, M.Sampels (Eds.), Hybrid Metaheuristics. Vol. 6373 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 174–189.

[34] Pisinger, D., Ropke, S., 2010. Large neighborhood search. In: Gendreau, M., Potvin, J.-Y. (Eds.), Handbook of Metaheuristics. Vol. 146 of Intl. Series in Operations Research & Management Science. Springer US, pp. 399–419.

[35] Prins, C., Prodhon, C., Ruiz, A., Soriano, P., Wolfler-Calvo, R., 2007. Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. Transportation Science 41 (4), 470–483.

[36] Prins, C., Prodhon, C., Wolfler-Calvo, R., 2006. A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. In: J.Gottlieb, Raidl, G. (Eds.), Evolutionary Computation in Combinatorial Optimization. Vol. 3906 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 183–194.

[37] Prins, C., Prodhon, C., Wolfler-Calvo, R., 2006. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR: A Quarterly Journal of Operations Research 4, 221–238.

[38] Prodhon, C., 2011. A hybrid evolutionary algorithm for the periodic location-routing problem. European Journal of Operational Research 210 (2), 204–212.

[39] Prodhon, C., Prins, C., 2008. A memetic algorithm with population management (MA|PM) for the periodic location-routing problem. In: Blesa, M., Blum, C., Cotta, C., Fernández, A., Gallardo, J., Roli, A., M. Sampels, M. (Eds.), Hybrid Metaheuristics. Vol. 5296 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 43–57.

[40] Raidl, G., Puchinger, J., 2008. Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. In: Blum, C., Aguilera, M., Roli, A., Sampels, M. (Eds.), Hybrid Metaheuristics. Vol. 114 of Studies in Computational Intelligence. Springer Berlin / Heidelberg, pp. 31–62.

[41] Sajjadi, S. R., Cheraghi, S., 2011. Multi-products location-routing problem integrated with inventory under stochastic demand. International Journal of Industrial and Systems Engineering 7 (4), 454–476.

[42] Salhi, S., Rand, G., 1989. The effect of ignoring routes when locating depots. European Journal of Operational Research 39 (2), 150–156.

[43] Shen, Z.-J., Qi, L., 2007. Incorporating inventory and routing costs in strategic location models. European Journal of Operational Research 179 (2), 372–389.

[44] Toth, P., Vigo, D. (Eds.), 2001. The vehicle routing problem. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[45] Wagner, H., Whitin, T., 1958. Dynamic version of the economic lot size model. Management Science 5 (1), 89–96.

[46] Whybark, D. C., 2007. Issues in managing disaster relief inventories. International Journal of Production Economics 108 (1-2), 228–235.

[47] Zhao, Q., Chen, S., Leung, S., Lai, K., 2010. Integration of inventory and transportation decisions in a logistics system. Transportation Research PartE: Logistics and Transportation Review 46 (6), 913–925.

3. Multi Start - Iterated Local Search for Multi-Depot Multi-Vehicle Inventory-Routing

W.J. Guerrero[1,2], C. Prodhon[1], N. Velasco[2], C.A. Amaya[2]

[1] ICD-LOSI (UMR-STMR CNRS 6279), Université de Technologie de Troyes,
12 rue Marie Curie, CS 42060 10004 Troyes Cedex, (France)
[2] PyLO, Depto de Ingeniería Industrial, School of Engineering, Universidad de los Andes,
Cra 1 # 18A- 12, Bogotá, (Colombia)

A heuristic procedure to solve the Inventory-Routing problem with multiple vehicles and multiple depots is presented. Consider a set of depots and a set of retailers, both capable of holding stock. The problem consists on determining their inventory policies and the routes to distribute product from depots to retailers to satisfy the demand faced by the retailers. This paper presents a multi-start iterated local search algorithm tested on benchmark instances for the single depot case. Further, the presented approach has competitive performance and new best solutions are found on large instances for the multi-vehicle single-depot case. For the multi-depot setting, computational experiments are performed by comparing the results of the presented heuristic against a lower bound computed by a cutting plane procedure, proving competitive performance of the presented method.

Keywords: Metaheuristics, inventory-routing, vehicle routing, Vendor-Managed Inventory System (VMI), multi-depot vehicle routing

3.1. Introduction

Recent research on simultaneous optimization of inventory management policies and routing practices, denoted as the Inventory-Routing problem (IRP), is targeted to two objectives: 1) To develop exact and heuristic methodologies capable of finding high quality solutions within reasonable computation time (models assuming deterministic demand [3, 4, 7, 8, 13, 20, 21], and models assuming stochastic demand [15, 18, 24]); and 2) to estimate the potential savings compared to making a sequential optimization heuristic in the forms: inventory first-route second and vice versa [5, 25].

The IRP is commonly defined as the problem of optimizing the lot sizes to replenish a set of retailers, together with the computation of the minimum cost routes to make the corresponding deliveries. A bibliographic review on IRP models and methods is presented by Andersson et al. [2]. A real-life application to the gas distribution industry is presented by Bell et al. [5]. Gaur and Fisher [13] study a periodic version of the IRP on a supermarket chain. Further, since the spreading of the vendor-managed inventory system (VMI), solving the IRP is more interesting from an applied point of view [9].

This problem reduces to subproblems well studied in the literature. On one hand, the periodic vehicle routing problem (PVRP) is a special case in which a single depot is available and quantities to deliver are known. The decisions to optimize in this case are the delivery days per client and the routing per period. The PVRP is NP-hard [12]. This proves the IRP to be NP-hard. Thus, considering that real size instances are potentially composed by hundreds of retailers and a planning horizon of at least a week, industrial needs require high quality solutions computed within controllable computation time. In the meanwhile, exact procedures have been proposed using extensive computational resources [3, 7, 8, 19].

On the other hand, the multi-depot vehicle routing problem (MDVRP) is a special case considering fixed quantities to deliver and a single period horizon. Thus, it is closely related to the IRP [23]. The target of the MDVRP is to replenish a set of retailers from a set of depots while guaranteeing the following: 1) each route starts and ends at the same depot, 2) each retailer is visited exactly once, 3) the total demand satisfied by each route does not exceed the vehicle capacity, and 4) the total routing cost is minimized.

The scope of this paper is to study the natural extension of the IRP on the multi-depot case (MDIRP), by integrating the conditions described for the MDVRP, and to provide a heuristic procedure to solve it. Previous research considering several depots and several product types are studied by Agra et al. [1], Engineer et al. [10], Gaur and Fisher [13], paying special attention to oil distribution on maritime logistics but ignoring inventory holding constraints at depots.

The remainder of the paper is organized as follows. In section 3.2, the mathematical definition of the problem is provided, while in section 3.3 the heuristic is presented. The computational study and conclusions are reported in sections 3.4 and 3.5 respectively.

3.2. Problem Definition

Consider a set of geographically dispersed facilities V. This set is formed by the set of depots I and the set of retailers J ($V = \{I \cup J\}$). The arcs in the set A connect every pair of retailers and depots to retailers. Consider a planning horizon H composed by p periods ($H = \{1, ..., p\}$). Dummy periods are included to model initial and final conditions of stock levels. Let H_0 be defined as $H \cup \{0\}$ and H' be defined as $H \cup \{p+1\}$. The set K is composed by the available fleet of identical vehicles. Retailers have to satisfy a deterministic and non-constant demand, denoted by d_{jt} for each retailer $j \in J$ in period $t \in H$, without backlogging. Every type of facility $j \in V$ might hold stock up to a level W_j and is associated to an initial inventory B_j. Each period, depot $i \in I$ is replenished by a fixed quantity r_i. The cost for holding product at facility $j \in V$ from period $t \in H_0$ up to period $l \in H'$ is denoted as q_{jtl}. Transportation costs are assumed to be constant during the planning horizon. Thus, c_{ij} denotes the cost of sending a vehicle from node i to node j ($i, j \in V$) and a maximum number of vehicles U is to be used, each with capacity of Q units.

The decisions to be optimized are: 1) The assignment of every retailer j to a depot i. The binary decision variable $f_{ij} = 1$ if retailer j will be replenished from depot i persistently on the planning horizon; 2) The quantity held in stock at depot i, in the first echelon of the supply chain, from period t to period l ($l \in H_0, t \in H'$, and $l \geq t$), denoted as w_{1itl}; 3) The quantity held in stock at retailer j, in the second echelon of the supply chain, from period 0 to period l, denoted as w_{2j0l}; 4) The quantity held in stock at retailer $j \in J$, in the second echelon of the supply chain, from period t to period l replenished from depot $i \in I$, is denoted as w_{2ijtl}; and 5) The routing activities per period. Binary decision variables $x_{ijtk} = 1$ iff vehicle k goes from node i to node j at period t. The model is formulated as detailed in the following.

3.2.1. Objective function

Equation (37) presents the objective function as the minimization of inventory and routing costs. Four terms are included into the equation: The first term adds the inventory holding cost at depots while the second term represents the inventory holding cost at retailers for the initial inventories. The third term adds the inventory holding cost at retailers for the quantities replenished from depots; and the final term sums the distribution costs.

$$\min \sum_{i \in I} \sum_{t \in H_0} \sum_{l=t|l>0}^{p+1} q_{itl} w_{1itl} + \sum_{j \in J} \sum_{t \in H'} q_{j0t} w_{2j0t} + \\ \sum_{i \in I} \sum_{j \in J} \sum_{t \in H} \sum_{l=t}^{p+1} q_{jtl} w_{2ijtl} + \sum_{(i,j) \in A} \sum_{k \in K} \sum_{t \in H} c_{ij} x_{ijtk} \tag{37}$$

3.2.2. Demand satisfaction constraints

Equations (38) force retailers to satisfy the demand whether by using its initial inventory or with the inventory previously replenished by a depot. Equations (39) restrain each retailer to be supplied by it assigned depot. While constraints (40) imply that each retailer is assigned to a single depot.

$$\sum_{i \in I} \sum_{l=1}^{t} w_{2ijlt} + w_{2j0t} = d_{jt} \;\; \forall j \in J, \forall t \in H. \tag{38}$$

$$\sum_{l=1}^{t} w_{2ijlt} \leq f_{ij} d_{jt} \;\; \forall i \in I, \;\; \forall j \in J, \forall t \in H. \tag{39}$$

$$\sum_{i \in I} f_{ij} = 1, \;\; \forall j \in J. \tag{40}$$

3.2.3. Inventory Flow coordination constraints

The set of constraints (41) coordinate the flow of products from depots at the first echelon of the supply chain towards the second echelon (retailers). At retailer j, the initial inventory is equal to B_j as stated by equations (42). Analogously, the initial inventory at depot i must be equal to B_i only if there is one or more retailers assigned to the corresponding depot forced by constraints (43) and (44). This means that if a depot has no assigned retailers, the initial inventory should not be positive. Further, each depot $i \in I$ with at least one assigned retailer is forced to have a constant production of r_i units per period as stated by constraints (45).

$$\sum_{l=0}^{t} w_{1ilt} = \sum_{l=t}^{p+1} \sum_{j \in J} w_{2ijtl} \;\; \forall i \in I, \forall t \in H. \tag{41}$$

$$\sum_{t \in H'} w_{2j0t} = B_j, \;\; \forall j \in J. \tag{42}$$

$$\sum_{t \in H'} w_{1i0t} = B_i \cdot y_i, \;\; \forall i \in I. \tag{43}$$

$$f_{ij} \leq y_i, \;\; \forall j \in J, \;\; \forall i \in I. \tag{44}$$

$$\sum_{t \in H'} w_{1ilt} = r_i \cdot y_i, \;\; \forall i \in I, \;\; \forall l \in H. \tag{45}$$

3.2.4. Capacity constraints

Every facility in the supply chain is assumed to have a limited storage capacity. Constraints (46) restrain the inventory held on stock at depots. In particular, if a depot i has no assigned retailers ($y_i = 0$), it holds zero stock. The set of constraints (47) limit the stock levels at retailers.

$$\sum_{r=0}^{t} \sum_{l=t}^{p+1} w_{1irl} \leq W_i \cdot y_i \;\; \forall i \in I, \;\; \forall t \in H. \tag{46}$$

$$\sum_{l=t}^{p+1} \left(w_{2j0l} + \sum_{r=1}^{t} w_{2ijrl} \right) \leq W_j \;\; , \forall i \in I, \;\; \forall j \in J, \;\; \forall t \in H. \tag{47}$$

3.2.5. Distribution constraints

Concerning routing, the set of constraints (48) coordinates replenishment and distribution (i.e. if a retailer is supplied, there must be at least a vehicle making the delivery). Vehicle capacity constraints are stated by equations (49). These force the number of vehicles visiting a subset of retailers S to be larger than the quantity delivered divided by the vehicle capacity Q. Vehicle flux conservation is forced by equations (50). Split deliveries are not considered. At most a single vehicle is allowed to visit a retailer per period as forced by constraints (51). The set of constraints (52) coordinates allocation decisions with routing, e.g. a vehicle departing from depot i visits retailer j if and only if retailer j is allocated to the corresponding depot i. Subtour elimination constraints are stated by equations (53).

$$\sum_{l=t}^{p+1} w_{2ijtl} \leq Q \sum_{u \in V/\{j\}} \sum_{k \in K} x_{ujtk} \;\; \forall i \in I, \;\; \forall j \in J, \;\; \forall t \in H. \tag{48}$$

$$\sum_{j \in S} \sum_{l=t}^{p+1} w_{2ijtl} \leq Q \sum_{j \in S} \sum_{u \in V/S} \sum_{k \in K} x_{ujtk} \;\; \forall i \in I, \;\; \forall S \subset J, \;\; \forall t \in H. \tag{49}$$

$$\sum_{j \in V} x_{ijtk} - \sum_{j \in V} x_{jitk} = 0, \ \forall t \in H, \ \forall i \in J, \ \forall k \in K. \tag{50}$$

$$\sum_{i \in V} \sum_{k \in K} x_{ijtk} \leq 1 \ \forall \ t \in H, \ \forall j \in V. \tag{51}$$

$$\sum_{u \in J} x_{iutk} + \sum_{u \in V \setminus \{j\}} x_{ujtk} \leq 1 + f_{ij} \ \forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall k \in K. \tag{52}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijtk} \leq |S| - 1 \ \forall t \in H, \ \forall S \subseteq J, \ \forall k \in K. \tag{53}$$

3.2.6. Decision variable types

The presented model is based on Mixed-Integer Programming (MIP). Equations (54) to (59) state whether a decision variable is binary, integer or real.

$$y_i \in \{0, 1\} \ \ \forall i \in I. \tag{54}$$

$$f_{ij} \in \{0, 1\} \ \ \forall i \in I, \ \forall j \in J. \tag{55}$$

$$x_{ijtk} \in \{0, 1\}, \ \ i, j \in V, \ \forall t \in H, \ \forall k \in K. \tag{56}$$

$$w_{2ijtl} \in \mathbb{R}^+ \ \forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall l \in H' | l \geq t. \tag{57}$$

$$w_{2j0t} \in \mathbb{R}^+ \ \ \forall j \in J, \ \forall t \in H'. \tag{58}$$

$$w_{1itl} \in \mathbb{R}^+ \ \ \forall i \in I, \ \forall t \in H_0, \ \forall l \in H | l \geq t. \tag{59}$$

3.3. Multi-Start Iterated local search

To solve the MDIRP, a procedure based on an Iterated Local Search (ILS) is proposed. A classical ILS builds solutions successively by creating at each iteration one child-solution using mutation and local search on the incumbent solution. The child solution is accepted according to a predefined criteria [16]. This scheme has been chosen since the evolution of solutions through perturbations followed by local search has been proven to be efficient on vehicle routing problems [22]. Additionally, to achieve diversification it is proposed to re-start the procedure from a new random initial solution. This technique has proven to overcome local optimality [17]. Therefore, the proposed algorithm fits within the definition of a Multi-Start Iterated Local Search (MS-ILS).

To code a solution, it is proposed to store information about inventory policies at every facility using variables w_{2ijtl}, w_{2j0t}, and w_{1itl} $\forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall l \in H' | l \geq t$

with the same definition as provided in section 3.2. As for depot-retailer allocation, the integer vector f'_j indicates the depot allocated to retailer $j \in J$. Routing decisions will be coded as permutations of retailers. One sequence per vehicle each period.

The MS-ILS is composed by three main components: a heuristic to build an initial solution, a local search operator and a perturbation operator, all three described next.

3.3.1. Initial solution

To find an initial solution, a sequential constructive heuristic is proposed based on mixed-integer programming (MIP). The complexity of the presented MDIRP in equations (37)-(59) is significantly reduced by computing a lower bound on routing costs instead of computing feasible routing. A trivial estimation on the routing cost for each depot is:

$$z^{LB} = \sum_{i \in I} \sum_{j \in J} \sum_{t \in H} \cdot z_{ijt} \cdot \hat{c}_{ij} \tag{60}$$

Where a new binary variable is included, z_{ijt}, equal to 1 if at least one unit of product is sent from depot i to retailer j at period t. The parameter $\hat{c}_{ij}$ represents an estimation of the cost of traveling from depot i to retailer j. Coordination of this decision variable and inventory decisions is induced by adding the following set of constraints:

$$\sum_{l=t}^{p+1} w_{2ijtl} \leq M_j \cdot z_{ijt} \;\; \forall i \in I, \;\; \forall j \in J, \;\; \forall t \in H. \tag{61}$$

Where the value of M_j is the minimum between Q and W_j. Then, the fourth term in the objective function is replaced by z^{LB} defined in equation (60), the set of constraints (61) is also included together with constraint (62) which guarantees that inventory decisions respect the total capacity of vehicles per period. Further, constraints (48) to (53) and (56) are excluded in the process of obtaining an initial solution.

$$\sum_{i \in I} \sum_{j \in J} \sum_{l=t}^{p+1} w_{2ijtl} \leq Q \cdot U \;\; \forall t \in H. \tag{62}$$

By solving the simplified MIP using commercial solvers, inventory and allocation decisions are fixed. Routing decisions are constructed using direct deliveries (one retailer per route) each period.

3.3.2. Local Search Operator

The proposed local search operator combines several neighborhoods. They might be grouped into three main types: neighborhoods on the routing construction, on the inventory decisions, and on the allocation decisions. They are explored following the systematical change of neighborhoods as proposed in a variable neighborhood descent (VND) subroutine [14]. This procedure makes a hierarchical order of neighborhoods. Following this order, it fully explores a given neighborhood and applies a best-improvement strategy. If an improving movement is found, it is applied and the search is reinitialized from the first neighborhood. If no improving movement is found, the search is changed to the next neighborhood. The definition and priority of the neighborhoods is described next.

- *Local Search on Routing Decisions*

The local search operator on routing decisions evaluates independently each period and each depot. The considered moves are: EXCHANGE, SWAP, 2-Opt, 3-Opt. The last neighborhood is the only one performed exclusively on single routes. These are standard in most heuristics for the capacitated vehicle routing problem.

Note that limiting the search on movements for a single depot and single period case allows to perform fast evaluations since inventory and allocation decisions are fixed. When relocating a retailer to a different depot, a significantly larger number of operations is required to keep the solution feasible. Those are: 1) re-optimizing inventory policies at depots, and 2) repairing routing in every period since retailers require to be consistently visited by the same depot on the planning horizon.

- *Local Search on Inventory Decisions*

Two neighborhoods are considered: a) For each retailer, it is potentially possible to delete a visit. This requires to increase the stock level in the previous scheduled visit. For example, imagine retailer j has scheduled visits only at periods 2 and 5 to deliver $\sum_{l=2}^{p+1} w_{2ij2l}$ and $\sum_{l=5}^{p+1} w_{2ij5l}$ units of product respectively. If visit at period 5 is deleted, the new quantity delivered at period 2 ($\sum_{l=2}^{p+1} w'_{2ij2l}$) is increased by $\sum_{l=5}^{p+1} w_{2ij5l}$. That is:

$$w'_{2ij2l} = w_{2ij2l} + w_{2ij5l}, \ \forall l \in \{H' | l \geq 5\} . \tag{63}$$

$$w'_{2ij5l} = 0, \ \forall l \in H'. \tag{64}$$

The change in the cost of the incumbent solution includes the increased holding cost (in the example it is $\sum_{l=5}^{p+1} w_{2ij5l} \cdot (q_{j2l} - q_{j5l})$); and the reduction in distribution costs due to the removal of client j from its route in period 5.

b) For each retailer, it is possible to decrease inventory levels. Contrary to the previous case, stock levels are reduced without modifying the scheduled visits. To illustrate further, suppose that a retailer j has scheduled visits only at periods 2 and 5 to deliver $\sum_{l=2}^{p+1} w_{2ij2l}$ and $\sum_{l=5}^{p+1} w_{2ij5l}$ units of product respectively. Assume $\sum_{l=5}^{p+1} w_{2ij2l}$ to be positive, meaning that a vehicle departing from depot i in period 2, supplies product at retailer j to be stocked up to a period greater or equal to 5. Now assume that this quantity $\sum_{l=5}^{p+1} w_{2ij2l}$ plus the total quantity delivered at j in period 5 ($\sum_{l=5}^{p+1} w_{2ij5l}$) is inferior to the vehicle capacity Q. Then, it is possible to transfer the fraction of the quantity delivered in period 2 that is stocked until period $l(l \geq 5)$ to be delivered in period 5 without falling into stock-out. Routing cost remain unchanged, while inventory holding costs at retailers are decreased. The new values for the decision variables are:

$$w'_{2ij5l} = w_{2ij5l} + w_{2ij2l}, \ \forall l \in \{H' | l \geq 5\} . \tag{65}$$

$$w'_{2ij2l} = 0, \ \forall l \in H'. \tag{66}$$

The total reduction in retailers' inventory holding costs for this example are computed as: $\sum_{l=5}^{p+1} w_{2ij2l} \cdot (q_{j5l} - q_{j2l}))$. Note that inventory policies at depots must be recomputed in the same way.

- *Local Search on Allocation Decisions*

A solution might be improved by modifying allocation decisions. Two neighborhoods are explored. First, by selecting a retailer and relocating it to a different depot. Second by selecting two retailers allocated to different depots and swapping them in their depot allocation. Keep in mind that routing decisions must be re-evaluated in both cases. Here, a best insertion heuristic is applied. As for inventory at depots, since their aggregated demand varies, so does their inventory policies.

3.3.3. Perturbation operator

It is proposed to make perturbations of the incumbent solution affecting simultaneously the routing, stock management and allocation decisions. The target is to avoid local optima. For MS-ILS, a single retailer is reallocated to a different depot. Routing and inventory decisions at the depots have to be repaired. Routing is repaired by a best-insertion heuristic. Further, on a random retailer, a new visit is scheduled on a random period. Note that stock levels at a retailers are forced to be reduced this way. This perturbation is performed for a random number of retailers.

3.3.4. General Heuristic Procedure

This algorithm combines the operators presented in sections 3.3.1 to 3.3.3 to solve the MDIRP. The pseudo-code of the MS-ILS is presented in algorithm 4. $N1$ restarts are executed. Each iteration an initial solution S_0 is obtained by solving the MIP defined in section 3.3.1. This operator is denoted as `Initial_Sol` in line 3. To obtain a different solution per iteration, two strategies are implemented: 1) $\hat{c}_{ij}$ is randomly perturbed. That is: $\hat{c}_{ij} = \delta \cdot c_{ij}$ for each depot $i \in I$ and retailer $j \in J$, where δ is defined as a random variable. These values provided the best results in tuning tests. 2) To force the replenishment to a set of retailers from a random depot at a random period.

At line 4, the local search operator (`LS`$(\cdot)$) is applied on the initial solution S_0. The inner loop in lines 6-15 is performed until $N2$ consecutive iterations without improvement on the incumbent solution S_0 are perceived. This inner loop starts by applying the perturbation operator at line 7 (operator explained in section 3.3.3) and the local search procedure at line 8 on solution S_0. The incumbent solution S_0 is replaced by S' in line 10 if the cost of solution S' is better than the cost of solution S_0 ($f(S') < f(S_0)$). The cost function $f(\cdot)$ is computed with equation (37).

The best solution of the search is stored in S^* at line 16 with procedure `Save_Best`. Note that the local search operator is never invoked less than $N1 \cdot (N2 + 1)$ times.

3.4. Computational Study

For all results presented by the proposed MS-ILS heuristic, tests were run on an Intel Xeon with a 2.80Ghz processor and 12 GB RAM. Xpress-IVE 7.0, 64-bits was used as MIP solver. The algorithm is coded in C.

Algorithm 4 MS-ILS for MD-IRP

$S^* \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** $N1$ **do**
  $S_0 \leftarrow$ Initial_Sol;
  $S_0 \leftarrow$ LS(S_0);
  $j := 0$;
  repeat
    $S' \leftarrow$ Perturbation (S_0);
    $S' \leftarrow$ LS(S');
    if $f(S_0) > f(S')$ **then**
      $S_0 \leftarrow S'$;
      $j := 0$;
    else
      $j := j + 1$;
    end if
  until $j = N2$
  $S^* \leftarrow$ Save_Best(S_0, S^*);
end for
Return S^*;

Tests are performed using the benchmark IRP instances of Archetti et al. [3, 4], Bertazzi et al. [6] for the single vehicle, single depot case to which optimal solutions are known. The benchmark instances for the multi-vehicle case adapted by Coelho et al. [7], Coelho and Laporte [8] are also included in our tests. These are available at: [http://www.leandro-coelho.com/instances/inventory-routing].

These sets of instances are adapted to the multi-depot version of the problem, adding one and two depots. The coordinates for the new depots are randomly generated on a 500×500 square, as in the original instances. The initial inventories are randomly generated as once or twice the initial inventories at the original depot. Depot storage capacities where generated to be the maximum between the corresponding initial inventory and the storage capacity of the original depot randomly multiplied by one or two. Finally, holding costs for the new depots are randomly chosen to be once or twice the holding cost at the original depot. In total, 390 instances were solved with two depots and 390 instances with three depots. 780 instances in total for the multi-depot multi-vehicle case.

For preliminary tuning tests, a random sample of 30 instances with different sizes (with single and multiple depots; and single and multiple vehicles as well) were selected from the benchmark, all of them with known optimal solutions. Three parameters are required to be tuned for the proposed MS-ILS. First, δ is required when building the initial solutions. Second, $N1$ and $N2$ representing number of iterations for the outer and inner loop of the algorithm respectively. The range for δ was tested to be in the set: $\{[0, 2], [0.5, 1.5], [0.9, 1.1]\}$. Also, the pair ($N1$, $N2$) was chosen to evaluate no less than 62,500 solutions. It will be shown that this value provides an interesting trade-off between solution quality and processing time. The tested values for these parameters where: $(N1, N2) \in \{(250, 250), (100, 650), (650, 100)\}$.

In total, nine combinations of parameters were compared and for each combination, the MS-ILS was executed 10 times. The average results of these preliminary tests for gap, standard deviation of the gap (σ) and average computational time (avg cpu) in seconds are shown in table 7.

Table 7: Tuning tests results

	$\delta \in$ **[0.9, 1.1]**			$\delta \in$ **[0.5, 1.5]**			$\delta \in$ **[0, 2]**		
(N1 , N2)	**avg gap**	σ	**avg cpu**	**avg gap**	σ	**avg cpu**	**avg gap**	σ	**avg cpu**
(100, 650)	0.42	0.34	35.17s	0.69	0.52	160.85s	1.12	0.56	186.09s
(250, 250)	0.43	0.31	55.42s	1.29	0.34	525.05s	1.63	1.52	1162.77s
(650, 100)	0.44	0.24	145.76s	1.16	0.25	1032.45s	1.15	0.28	1093.52s

Four parameter combinations are candidates to be chosen from those depicted in table 7. The three combinations with $\delta \in$ [0.9, 1.1] and a fourth one with (N1,N2) = (100, 650) and $\delta \in [0.5, 1.5]$. The remaining combinations do not seem to be good candidates since solution quality and the computational times are not competitive in this preliminary test for computational time and solution quality. These results are analyzed with a non-parametric test. The performance of each candidate parameter combination was ranked from 1 to 4 for each tested instance. The combination of parameters with the lowest average gap is ranked with 1 and the one with the largest average gap is ranked with 4. The average rank of each parameter combination is shown in table 8.

Table 8: Average ranks for each candidate parameter combination

Parameter Combination ($N1$, $N2$)	δ	Average rank
(100, 650)	[0.9, 1.1]	2.37
(250, 250)	[0.9, 1.1]	2.25
(650, 100)	[0.9, 1.1]	2.38
(100, 650)	[0.5, 1.5]	2.90
Average		2.48

The conclusion of the non-parametric test of Friedman [11] is that not enough statistical proof is provided to state there is a difference between the performance of the candidate combinations of parameters (the p-value of the test is 0.2). Therefore, the MS-ILS results are proven to be robust enough with respect to variations of the parameters and no significant improvement of the average solution quality are expected by selecting a different combination of parameters among the tested ones. Therefore, for further tests, the chosen combination of parameters is δ to be in the range [0.9, 1.1] and parameters $N1$ and $N2$ to be 100 and 650 respectively since the preliminary test showed these values to provide the best performance in solution quality and computational times. Potential improvement of the algorithm performance by making finer tuning of parameters is acknowledged. Nonetheless, the exposed results seem reasonably good for industrial applications.

Table 9 summarizes the results on single-depot, single vehicle IRP benchmark instances. They feature three or six periods (p = 3 or 6). Instances with three decision periods are formed with n retailers ranging from 5 to 50, while for those with six periods, n ranges from 5 to 30. Finally, they are also classified according to inventory holding costs as "high" or "low". Each instance type is composed by 5 instances. Solutions obtained by the presented MS-ILS in section 3.3 are compared to the results presented by Archetti et al. [3] with a hybrid heuristic dedicated to the single depot, single vehicle IRP, denoted as HAIR. Table 9 presents average percentage gap to the optimal solution and computational times in seconds for both methods.

The presented heuristic has competitive performance. On average, the presented MS-ILS solves instances with a gap to optimal solutions of 0.50% within 115s. In contrast, [3] reports results with an average gap of 0.05% computed on an average of 458s. From an applied point of view, MS-ILS responds to industrial needs since the model could provide a three-day planning (or weekly planning) in less than 5 minutes for problems with 50 retailers. Instances with "low" holding cost and six periods are the most difficult to solve for the MS-ILS. The average percentage gap to optimal solutions is 1.76% computed within 166s with this configuration. On the contrary, the instances providing the best average results are those with three periods and "high" inventory holding cost. This type of instances are solved within 80 seconds with an average gap to optimality of 0.02%.

Further, the heuristic HAIR proposed by Archetti et al. [3] is designed to be hybrid since an improvement operator is performed on a given solution by solving a series of MIP problems to optimality. In fact, the MS-ILS is also hybrid since initial solutions are computed using a MIP solver.

Also, comparing computational times does not lead to accurate conclusions since the computers, programming languages, and MIP solvers differ. The computations in Archetti et al. [3] are performed on a Intel Dual Core 1.86 GHz and 3.2 GB RAM and coded in C++. The MS-ILS was executed on an Intel Xeon with a 2.80Ghz processor and 12 GB RAM using Xpress-IVE. Nevertheless, an important conclusion is that the growth of the computational times with respect to the size of the instance of MS-ILS is lower than the one presented by Archetti et al. [3].

Note that two factors explain that the MS-ILS has lower growth in computational times: 1) The MS-ILS stopping criteria is fixed for every instance size. The number of iterations have been fixed by fixing $N1 = 100$ and $N2 = 650$. On the other hand, the heuristic HAIR is designed to stop until $200 \cdot n \cdot p$ iterations without improvement are performed; n being the number of retailers and p the number of periods. That is, the number of iterations increases polynomially with the number of periods and retailers of the instance. 2) The heuristic HAIR solves a series of MIP problems per iteration working as a local search based on exact methods. Instead, The MS-ILS is constructed with polynomially bounded perturbation and local search operators. Only initial solutions are constructed with a general purpose solver. Further, fig. 4 presents the average computational time provided by the HAIR heuristic and the MS-ILS executed on the instance sets from 5-50 retailers.

Benchmark instance sets for the single depot, multi-vehicle case range from two up to five vehicles ($U = \{2, 3, 4, 5\}$), 5 up to 50 retailers ($n = \{5, 10, 15, 20, 25, 30\}$), three or six periods (p), and inventory holding costs set as "high" or "low" are studied. Each instance type set is composed by five instances. Coelho and Laporte [8] propose a Branch-and-Cut algorithm (BCA) for the single and multi-vehicle single-

Table 9: MS-ILS: results on single vehicle benchmark instances

		MS-ILS[1]				HAIR [2] [3]			
		High holding cost		Low holding cost		High holding cost		Low holding cost	
	Instance set	p = 3	p = 6	p = 3	p = 6	p = 3	p = 6	p = 3	p = 6
U = 1	n = 5	0% (7s)	0.09% (25s)	0% (7s)	0.01% (24s)	0% (2s)	0% (18s)	0% (3s)	0% (16s)
	n = 10	0.07% (14s)	0.24% (61s)	0.06% (16s)	0.28% (61s)	0% (10s)	0.11% (86s)	0% (9s)	0.01% (65s)
	n = 15	0% (26s)	0.62% (117s)	0.02% (31s)	1.25% (115s)	0% (32s)	0.06% (226s)	0% (27s)	0.11% (180s)
	n = 20	0.04% (40s)	0.92% (189s)	0.01% (48s)	2.33% (191s)	0% (67s)	0.21% (401s)	0% (65s)	0.26% (456s)
	n = 25	0.03% (54s)	1.49% (276s)	0.03% (58s)	3.53% (267s)	0% (121s)	0.28% (1027s)	0% (113s)	0.13% (1031s)
	n = 30	0% (95s)	1.51% (346s)	0.07% (73s)	3.16% (339s)	0% (254s)	0.33% (1922s)	0% (260s)	0.21% (2534s)
	n = 35	0.02% (103s)		0.12% (117s)		0% (399s)		0% (415s)	
	n = 40	0.01% (117s)		0.05% (125s)		0% (579s)		0% (521s)	
	n = 45	0% (164s)		0.08% (157s)		0% (806s)		0% (778s)	
	n = 50	0.01% (180s)		0.06% (226s)		0%(1146s)		0%(1087s)	
	Average	0.02% (80s)	0.81% (169s)	0.05% (86s)	1.76% (166s)	0% (342s)	0.17% (613s)	0% (328s)	0.12% (714s)

Note 1: Coded in C and executed on a Intel-Xeon with a 2.80Ghz processor and 12 GB RAM. Xpress-IVE 7.0, 64-bits as MIP solver.
Note 2: Coded in C++ and executed on a Intel Dual Core 1.86 GHz processor and 3.2 GB RAM. Cplex as MIP solver.

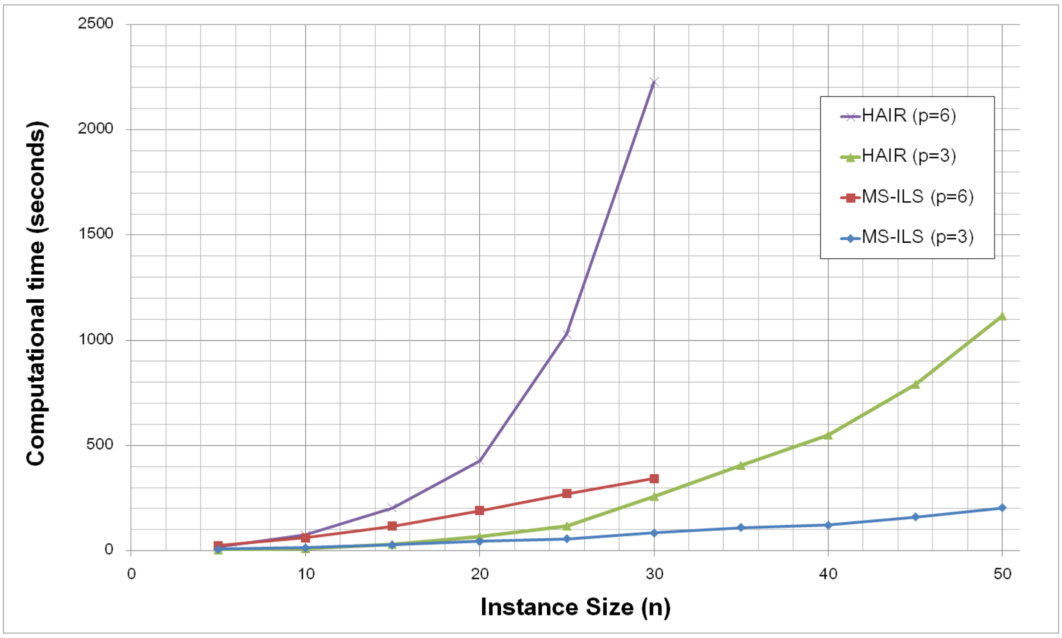

Figure 4: Computational times for HAIR and MS-ILS

depot Inventory-routing Problem. For the single vehicle case, results are provided by BCA with proven optimality on an average of 18.7s, and a maximum of 128s for an instance with 30 nodes, 6 periods and "low" inventory cost. For the multi-vehicle case, benchmark results from the BCA are known for lower and upper bounds after 12h of execution. Comparative results of MS-ILS and BCA for the single-depot multi-vehicle case are shown in table 10. The multi-vehicle case is much more combinatorial than the single vehicle case since the decision of "packing" retailers into vehicles per period is also included.

Columns `UB` present the average percentage gap between the MS-ILS and the upper bound provided by BCA. Similarly, columns `LB` present the average percentage gap between the MS-ILS and the lower bound provided by BCA. Columns `cpu` present the corresponding average computation time in seconds in the corresponding workstation. Coelho and Laporte [8] coded in C++ using CPLEX 12.3 with six threads on a grid of Intel Xeon processors running at 2.66 GHz with up to 48 GB of RAM installed per node. Again, scaling these results is a difficult task since the former algorithm is parallelized on several workstations and threads. Conclusions comparing computational times might be misleading. On the other hand, the method proposed by Coelho and Laporte [8] is a non-polynomial algorithm intended to prove optimality while the MS-ILS provides high quality solutions solving a MIP with polynomial number of constraints and iterating by performing polynomially bounded operations.

Note that the gap between the upper bound provided by MS-ILS and the upper bound provided by BCA is always inferior to 1% for instances with 3 periods and inferior to 4% for instances with 6 periods. The most difficult instances are those with larger planning horizons ($p = 6$) and low inventory holding costs. Note as well that the MS-ILS computational time is not highly sensitive to the number of available vehicles and feasible solutions are improved by around a mean of 1.05% for $p = 3$ and 1.12% for $p = 6$ from previous best known solutions. Once more, the MS-ILS is proven to have a lower

Table 10: MS-ILS: Results on single-depot multi vehicle benchmark instances

Instance		**MS-ILS**													**BCA** [8]			
		High						Low							High		Low	
Set		p = 3			p = 6			p = 3			p = 6				p = 3	p = 6	p = 3	p = 6
		UB	LB	cpu	UB	LB	cpu	UB	LB	cpu	UB	LB	cpu		cpu	cpu	cpu	cpu
U = 2	n = 5	0.03%	0.03%	(7s)	0.17%	0.17%	(28s)	0.01%	0.01%	(7s)	0.13%	0.13%	(29s)		3s	9s	4s	9s
	n = 10	0.25%	0.25%	(17s)	0.75%	0.75%	(66s)	0.04%	0.04%	(16s)	0.87%	0.87%	(66s)		6s	58s	8s	607s
	n = 15	0.06%	0.06%	(27s)	1.13%	1.13%	(127s)	0.02%	0.02%	(27s)	1.91%	1.91%	(122s)		12s	351s	12s	555s
	n = 20	0.12%	0.12%	(43s)	1.13%	1.13%	(184s)	0.07%	0.07%	(46s)	2.90%	2.90%	(192s)		24s	4036s	24s	8642s
	n = 25	0.07%	0.07%	(58s)	2.00%	2.00%	(296s)	0.06%	0.06%	(61s)	3.91%	4.17%	(305s)		31s	10160s	32s	19002s
	n = 30	0.05%	0.05%	(78s)	2.04%	2.73%	(331s)	0.01%	0.01%	(86s)	3.93%	5.73%	(355s)		70s	28789s	62s	36835s
	n = 35	0.04%	0.04%	(101s)				0%	0%	(97s)					66s		56s	
	n = 40	0.15%	0.15%	(121s)				0.04%	0.04%	(128s)					479s		525s	
	n = 45	0.07%	0.07%	(158s)				0.08%	0.08%	(141s)					1595s		3868s	
	n = 50	0.15%	0.15%	(163s)				0.02%	0.16%	(177s)					4432s		10797s	
U = 3	n = 5	0.92%	0.92%	(8s)	0.72%	0.72%	(39s)	0.35%	0.35%	(7s)	0.98%	0.98%	(39s)		3s	9s	5s	57s
	n = 10	0.04%	0.04%	(18s)	0.89%	1.63%	(71s)	0%	0%	(17s)	0.80%	1.84%	(67s)		13s	58s	17s	14579s
	n = 15	0%	0%	(27s)	1.49%	1.98%	(114s)	0.02%	0.02%	(30s)	2.28%	2.65%	(131s)		26s	351s	31s	18761s
	n = 20	0.35%	0.35%	(41s)	2.00%	6.13%	(203s)	0.24%	0.24%	(42s)	2.90%	11.0%	(196s)		217s	4036s	221s	42752s
	n = 25	0.23%	0.23%	(60s)	1.95%	6.97%	(313s)	0.05%	0.05%	(55s)	2.88%	13.2%	(296s)		1014s	10160s	574s	43047s
	n = 30	0.07%	0.07%	(85s)	1.07%	7.79%	(345s)	0%	0%	(81s)	2.27%	17.1%	(309s)		1623s	28789s	1286s	43080s
	n = 35	0.32%	0.32%	(90s)				0.15%	0.15%	(104s)					2696s		1936s	
	n = 40	0.38%	0.38%	(106s)				0.16%	0.16%	(116s)					6312s		9092s	
	n = 45	0.16%	0.56%	(140s)				0.01%	0.88%	(138s)					32821s		31805s	
	n = 50	**-1.46%**	3.07%	(155s)				**-5.80%**	7.57%	(160s)					42991s		42930s	
U = 4	n = 5	0.57%	0.57%	(7s)	1.15%	1.15%	(47s)	0.53%	0.53%	(8s)	0.23%	0.23%	(44s)		5s	51s	4s	78s
	n = 10	0.16%	0.16%	(19s)	1.17%	4.67%	(71s)	0.03%	0.03%	(21s)	1.43%	6.27%	(67s)		54s	25467s	41s	30232s
	n = 15	0%	0%	(29s)	1.60%	5.05%	(131s)	0.12%	0.12%	(30s)	2.30%	8.68%	(112s)		89s	39945s	119s	42293s
	n = 20	0.05%	0.05%	(46s)	1.33%	11.9%	(195s)	0.03%	0.03%	(46s)	2.08%	20.0%	(193s)		7779s	37441s	5544s	43098s
	n = 25	0.16%	0.16%	(69s)	**-22.2%**	13.5%	(299s)	0.44%	0.44%	(57s)	0.56%	24.0%	(273s)		9781s	34402s	4666s	42537s
	n = 30	**-0.13%**	0.99%	(93s)	**-1.55%**	12.7%	(330s)	**-0.58%**	2.51%	(93s)	**-18.1%**	31.2%	(313s)		27419s	43080s	29715s	42685s
	n = 35	0.07%	1.86%	(98s)				**-0.97%**	5.01%	(106s)					27955s		31756s	
	n = 40	0.29%	2.53%	(112s)				**-0.81%**	8.15%	(114s)					39508s		43010s	
	n = 45	**-1.08%**	3.37%	(145s)				**-3.66%**	9.84%	(154s)					34789s		34722s	
	n = 50	**-4.28%**	6.06%	(179s)				**-14.3%**	17.9%	(201s)					43032s		42999s	
U = 5	n = 5	0.05%	0.05%	(7s)	1.70%	1.70%	(49s)	0.27%	0.27%	(7s)	2.04%	2.04%	(45s)		5s	131s	5s	208s
	n = 10	0.26%	0.26%	(18s)	1.34%	7.47%	(85s)	0.35%	0.35%	(19s)	1.25%	10.9%	(74s)		75s	37589s	94s	36498s
	n = 15	0%	0%	(35s)	0.89%	8.69%	(138s)	0.06%	0.06%	(38s)	1.25%	16.0%	(130s)		1008s	42801s	1196s	37435s
	n = 20	0.15%	0.22%	(46s)	0.39%	16.8%	(217s)	0.21%	3.16%	(45s)	**-0.13%**	28.9%	(192s)		16268s	42965s	14619s	42446s
	n = 25	**-0.24%**	1.54%	(57s)	**-12.1%**	18.2%	(327s)	**-0.26%**	4.40%	(68s)	**-14.8%**	34.5%	(302s)		26255s	41064s	26720s	42282s
	n = 30	**-1.09%**	3.59%	(87s)	**-4.77%**	16.5%	(388s)	**-1.30%**	10.0%	(95s)	**-47.9%**	44.3%	(346s)		38874s	43043s	39794s	30445s
	n = 35	**-2.06%**	4.70%	(103s)				**-7.70%**	5.65%	(110s)					43013s		43010s	
	n = 40	**-1.34%**	4.65%	(116s)				**-8.11%**	15.0%	(129s)					37707s		34767s	
	n = 45	**-3.81%**	5.41%	(154s)				**-11.0%**	16.7%	(132s)					43103s		43046s	
	n = 50	**-8.00%**	8.74%	(183s)				**-18.0%**	27.0%	(189s)					43058s		43043s	
Average		**-0.46%**	1.30%	(78s)	**-0.65%**	6.31%	(183s)	**-1.73%**	3.43%	(80s)	**-1.83%**	12.1%	(175s)		13355s	22304s	5200s	25757s

growth in computational times than the exact method of Coelho and Laporte [8]. As a matter of fact, MS-ILS is able to improve the feasible solutions provided by Coelho and Laporte [8] on 124 instances out of the 640 instances analyzed in table 10. Fig. 5 and 6 exhibit the growth of computational times for the BCL and MS-ILS on instances with $p = 3$ and $p = 6$ respectively with different number of vehicles. The difference between the growth of computational times among both methods is important. Further, BCL is highly sensitive to the number of vehicles (U) while the proposed MS-ILS presents no significant variation in computational times when the number of available vehicles is increased. Logarithmic scale is used in these figures to depict a more clear difference between both methods on instances with two to five vehicles.

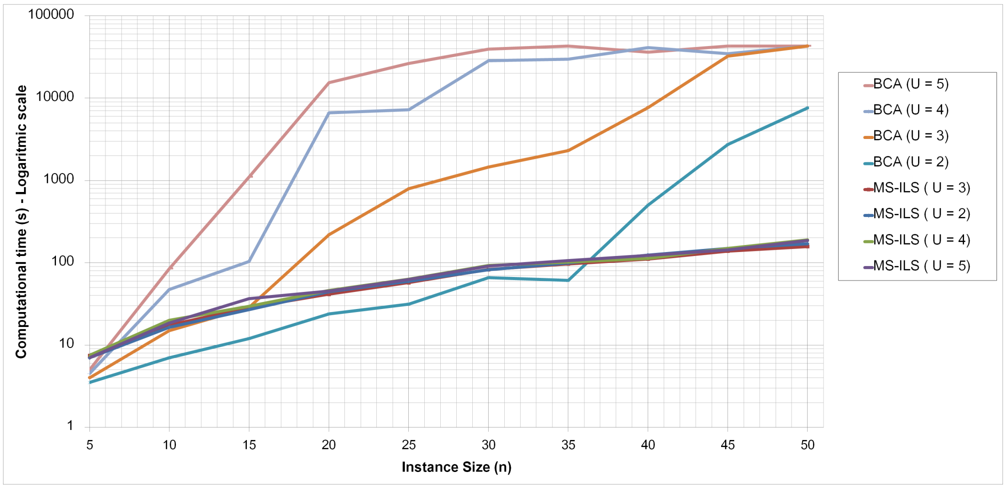

Figure 5: Computational times for BCA and MS-ILS on logarithmic scale on p=3 instances

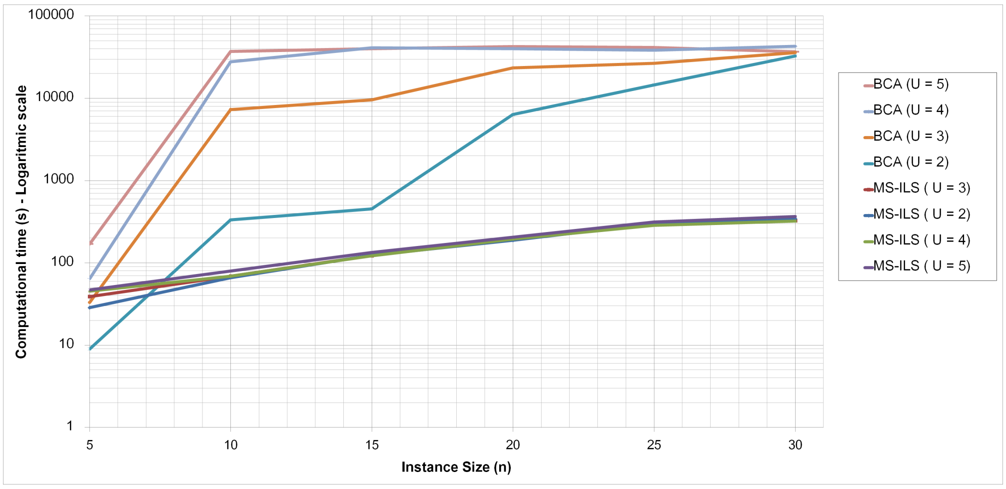

Figure 6: Computational times for BCA and MS-ILS on logarithmic scale on p=6 instances

To compute a valid lower bound on instances with multiple depots, a cutting plane method (CPM) is implemented, truncated by execution time. The standard sub-tour elimination constraints 53 are added dynamically. The maximum execution time for CPM is 1 hour per instance. Table 11 presents the results of CPM in terms of percentage of instances solved to proven optimality within one hour. CPM could find solutions up to three candidate depots, 3 period planning horizon and 45 retailers. Nonetheless, when the planning horizon is composed by 6 periods, relevant results for instances with up to 15 retailers could be computed within this time limit.

On average, 36.3% of all instances were solved to proven optimality. In detail, CPM found 70% of optimal solutions for instances with two depots, single vehicle, "high" inventory cost and up to 50 retailers. Similarly, up to 48% of the optimal solutions are known for instances with two depots, three periods, single vehicle, "low" inventory cost and up to 50 retailers. For the single vehicle, two depot case and "high" inventory cost, instances with up to 15 retailers and those with 25 retailers were always solved to optimality; 3 out of 5 of the instances with 20, 30 and 40 retailers, two depots and single vehicle have known optimal solutions. 40 % of the optimal solutions with 45 retailers are known while none of the instances with 50 retailers were solved to optimality. The quality of the lower bounds decreases as the number of vehicles increases and solutions with more than three vehicles were not acceptably tight lower bounds. Considering three vehicles, three periods and two depots, only 16% of the instances were solved to optimality, with the "high" and "low" inventory cost configuration. Therefore we limit the analysis to three vehicles. Current research is to develop improved methods to compute better lower bounds.

Results in table 12 show, in column `LB`, the percentage gap between the solution provided by MS-ILS and the best lower bound provided by CPM for instances with two depots. If the optimal solutions in the corresponding set of instances are known, the gap is highlighted in bold font. Column `cpu` presents the computational time of the MS-ILS in seconds. Instances with 5 retailers could be solved to optimality by CPM and the average gap between MS-ILS and the optimal solutions for this instance set is 1.17%. On average, MS-ILS provides solutions 9.98% larger than the lower bound. On the multi-depot setting, the quality of the lower bound could be inferior than the one for the single depot instances. The average computation time is 360 seconds. Once more, note that the computational time is not highly sensitive to the number of vehicles available. On the contrary, the length of the planning horizon does increases the computation time of the algorithm significantly. Instances with six periods are significantly harder to solve than those with three periods for CPM and MS-ILS. Keep in mind that MS-ILS is capable of solving instances with more than 15 retailers and six periods, but since CPM can not provide acceptable enough lower bounds, we are not able to make significant conclusions from that data.

Similarly, table 13 presents the results for the case where the set I is composed by three depots. Once more, the comparison is provided against the lower bound found by CPM and gaps in bold fonts present the gaps to optimal solutions when they are known. In this case, the average gap is 13.8% larger than the computed lower bound. The average computation time is 626 seconds. As far as we know, the quality of the lower bound in this case is not as good as in the previous case since fewer instances were solved to proven optimality. Nonetheless, the gaps and processing times remain reasonable for industrial applications. One important conclusion is that the instances become considerably harder with longer planning horizons.

Table 11: Percentage of multi-depot IRP instances solved with proven optimality by a cutting plane method within 1h.

		Two depots (High)		Two depots (Low)		Three depots (High)		Three depots (Low)	
	N	p=3	p=6	p=3	p=6	p=3	p=6	p=3	p=6
U = 1	5	100%	100%	100%	100%	100%	100%	100%	100%
	10	100%	60%	100%	80%	100%	80%	80%	60%
	15	100%	0%	60%	0%	100%	20%	100%	0%
	20	60%		60%		60%		80%	
	25	100%		60%		80%		100%	
	30	60%		0%		0%		40%	
	35	80%		20%		60%		60%	
	40	60%		60%		60%		20%	
	45	40%		20%		60%		60%	
	50	0%		0%		0%		0%	
	avg.	**70%**	**53%**	**48%**	**60%**	**62%**	**67%**	**64%**	**53%**
U = 2	5	100%	100%	60%	100%	100%	100%	100%	100%
	10	100%	0%	60%	0%	100%	0%	80%	0%
	15	60%	0%	80%	0%	80%	0%	20%	0%
	20	0%		0%		0%		0%	
	25	0%		20%		0%		0%	
	30	0%		0%		0%		0%	
	35	0%		20%		0%		0%	
	40	0%		20%		0%		0%	
	≥ 45	0%		0%		0%		0%	
	avg.	**26%**	**33%**	**26%**	**33%**	**28%**	**33%**	**20%**	**33%**
U = 3	5	100%	60%	100%	100%	100%	40%	100%	80%
	10	60%	0%	40%	0%	40%	0%	20%	0%
	15	0%	0%	20%	0%	20%	0%	0%	0%
	20	0%		0%		0%		20%	
	≥ 25	0%		0%		0%		0%	
	avg.	**16%**	**20%**	**16%**	**33%**	**16%**	**13%**	**14%**	**27%**
	Average	**37%**	**36%**	**30%**	**42%**	**35%**	**38%**	**33%**	**38%**

3.5. Conclusions

A formulation for the Inventory-Routing Problem is proposed considering multiple sources of production, denoted as depots. It is a combination of two well-known problems in the literature: the multi-depot vehicle routing problem (MDVRP) and the single depot Inventory-Routing problem (IRP). Both known to be NP-hard.

The paper presents a heuristic algorithm providing high quality solutions and running with lower computational resources than existing algorithms. A multi-start iterated local search (MS-ILS) procedure is proposed. Tests over benchmark instances for the single and multiple vehicle case show competitive results. In fact, the proposed MS-ILS finds 124 new best solutions for large benchmark instances of the IRP with multiple vehicles and single depot. On average, the previous best known solutions are improved by a mean of 1.05% for instances with three periods and 1.12% for instances with six periods.

On the multi-depot case, vehicles are forced to start and end at the same depot per period. Also, each retailer must be replenished from a single depot over the complete

Table 12: MS-ILS: Results on two-depot multi vehicle MDIRP instances

	Instance Set	High				Low			
		p=3		p=6		p=3		p=6	
		LB	cpu	LB	cpu	LB	cpu	LB	cpu
U = 1	n= 5	**0.35**	13.51	**1.34**	43.64	**0**	10.63	**0.43**	40.08
	n= 10	**3.00**	23.96	0.87	203.27	**0.02**	27.12	0.82	221.43
	n= 15	**2.59**	39.07	4.15	460.44	2.64	42.63	7.72	1208.91
	n= 20	0.81	72.16			3.41	82.09		
	n= 25	**0.02**	166.06			0.91	141.30		
	n= 30	0.31	239.60			0.82	186.86		
	n= 35	0.36	288.45			2.81	254.75		
	n= 40	0.64	336.55			1.97	287.57		
	n= 45	1.11	418.52			2.01	415.12		
	n= 50	2.56	549.54			2.48	546.31		
	avg.	1.18	214.74	2.12	235.78	1.71	199.44	2.99	490.14
U = 2	n= 5	**1.44**	10.75	**0.34**	56.39	0	10.43	**0.41**	53.32
	n= 10	**0.79**	43.89	8.91	466.56	2.70	38.84	18.61	427.96
	n= 15	1.42	83.28	8.96	969.01	7.07	71.96	23.32	1504.50
	n= 20	7.10	119.03			13.23	110.71		
	n= 25	7.83	185.66			27.20	199.52		
	n= 30	5.78	271.56			22.26	288.69		
	n= 35	6.66	388.86			27.34	432.45		
	n= 40	7.71	489.96			31.76	525.42		
	n= 45	8.78	559.76			23.36	505.02		
	n= 50	6.73	562.70			31.15	578.65		
	avg.	5.42	271.55	6.07	497.32	18.61	276.17	14.11	661.93
U = 3	n= 5	**6.33**	14.37	2.33	73.25	**0.08**	11.96	**0.98**	68.60
	n= 10	3.65	55.51	19.18	348.57	8.28	47.10	19.01	358.26
	n= 15	18.72	84.63	31.07	846.87	16.95	61.10	30.56	1066.35
	n= 20	10.33	124.40			23.86	112.99		
	n= 25	12.91	199.75			38.09	162.10		
	n= 30	7.42	310.63			33.13	273.47		
	n= 35	8.28	488.44			23.70	340.97		
	n= 40	9.17	588.50			31.63	462.36		
	n= 45	13.21	610.89			26.41	457.03		
	n= 50	7.19	613.92			32.31	517.26		
	avg.	9.72	309.10	17.53	422.90	23.44	244.63	16.85	497.74
Average		5.44	265.13	8.57	385.33	14.59	240.08	11.32	549.93

planning horizon. In real applications, this constraint makes products to be easily trackable. Further, only one route per vehicle is allowed each period. The complexity of the multi-depot problem is significantly increased since the allocation problem of retailers to depots is included. Thus, new instances are proposed and a lower bound is computed via a basic cutting plane method. On average, the gap between this lower bound and the MS-ILS solutions is 9.98% for instances with two depots and 13.8% for instances with three depots.

Table 13: MS-ILS: Results on three-depot multi vehicle MDIRP instances

Instance		High				Low			
Set		p=3		p=6		p=3		p=6	
		LB	cpu	LB	cpu	LB	cpu	LB	cpu
U = 1	n= 5	**0.48**	12.72	**0.91**	45.02	**0.44**	12.02	**1.16**	43.27
	n= 10	**2.41**	29.20	1.70	298.12	0.92	28.94	4.14	330.25
	n= 15	**1.83**	79.08	8.92	915.81	**2.43**	61.25	10.55	753.54
	n= 20	1.95	145.65			2.29	133.68		
	n= 25	1.83	202.10			**3.79**	202.23		
	n= 30	4.67	370.34			3.69	389.95		
	n= 35	2.48	404.71			4.49	486.35		
	n= 40	2.66	597.97			6.59	791.98		
	n= 45	5.59	1308.29			5.47	832.38		
	n= 50	4.76	1777.17			14.90	1979.80		
	avg.	2.87	492.72	3.84	419.65	4.50	491.86	5.28	375.69
U = 2	n= 5	**0.78**	7.45	**2.26**	61.81	**0**	8.49	**0.02**	61.10
	n= 10	**2.22**	80.74	12.14	759.87	3.68	64.60	25.47	792.91
	n= 15	8.03	182.85	27.25	2471.43	5.26	154.25	39.11	2420.51
	n= 20	27.97	268.95			12.11	283.24		
	n= 25	24.81	465.78			25.01	447.11		
	n= 30	5.07	632.72			39.49	656.32		
	n= 35	12.06	872.34			20.18	886.41		
	n= 40	28.98	1120.18			31.15	1290.41		
	n= 45	17.20	1565.55			27.07	1726.29		
	n= 50	10.16	1897.56			27.24	1829.20		
	avg.	13.73	709.41	13.88	1097.70	19.12	734.63	21.53	1091.51
U = 3	n= 5	**2.89**	16.08	9.34	73.73	**0.09**	15.11	0.80	68.02
	n= 10	9.39	63.98	22.51	537.34	6.69	52.69	21.40	769.22
	n= 15	12.56	148.07	29.38	1653.24	22.80	103.62	43.03	1148.63
	n= 20	14.32	214.00			37.88	229.04		
	n= 25	24.81	387.16			37.81	359.32		
	n= 30	19.36	549.21			34.57	467.54		
	n= 35	15.74	698.63			47.08	585.69		
	n= 40	17.29	986.63			41.90	834.07		
	n= 45	10.25	1307.38			19.50	1272.24		
	n= 50	21.02	1517.65			24.29	1577.21		
	avg.	14.76	588.88	20.41	754.77	27.26	549.65	21.74	661.96
Average		10.45	597.00	12.71	757.37	16.96	592.05	16.19	709.72

Future research is focused on improving the provided lower bounds and extending the method on the maritime logistics setting. In maritime transportation, ships do not have fixed depots and routes might start and end in different ports. Also, the planning horizon needs to be longer since port operations and sailing times are longer, increasing this way the importance of inventory decisions. Finally, the problem needs to be solved for the case when the ships composing the fleet differ in capacity, cost, or travel times [2]. Recent work studies a multi-depot, multi-product case in the oil industry where distribution is performed by ships and loading/unloading costs are relevant. Nevertheless, inventory holding costs and inventory management at depots are often ignored [1, 10]. Indeed, another future research is to study the effect of relaxing the constraints limiting the allocation of retailers to a single depot.

Acknowledgment

This research is partially supported by Champagne-Ardenne Regional Council (France) and Centro de Estudios Interdisciplinarios Básicos y Aplicados - CEIBA (Colombia). The authors thank John Leonardo Vargas for helping with the implementation of the cutting plane method for MDIRP described in section 3.4.

Chapter 3 was submitted as:

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya (2013) *Multi Start - Iterated Local Search for Multi-Depot Multi-Vehicle Inventory-Routing*. European Journal of Operational Research. IN SUBMISSION

Preliminary results were presented at:

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya (2013) Multi-start Iterated Local Search for the Multi-depot Inventory-Routing Problem. Seminario PyLO. Universidad de los Andes, Colombia. August 28th (Invited Talk)

3.6. Résumé en français

Ce chapitre traite le problème connu comme le problème de tournées avec gestion de stocks (Inventory-routing, IRP). Des articles récents sur le sujet ont montré deux motivations: 1) développer des méthodes exactes et heuristiques capables de trouver des solutions de haute qualité dans un temps de calcul raisonnable (pour les modèles posant l'hypothèse que la demande soit déterministe, voir [3, 4, 7, 8, 20, 21], et pour des articles présentant des modèles stochastiques, des exemples sont trouvés chez [15, 18, 24]); et 2) pour estimer les économies potentielles en comparant les méthodes d'optimisation globale avec les méthodes d'optimisation séquentielle telles que: stock d'abord - routage ensuite (inventory first- routing second) et vice versa [5, 25].

L'IRP est communément défini comme le problème consistant à optimiser la taille des lots pour approvisionner un ensemble de détaillants, ainsi que les tournées des véhicules pour faire les livraisons correspondantes. Une révision de la littérature sur les modèles et méthodes pour l'IRP est présentée par Andersson et al [2]; et une application réelle de l'industrie de distribution de gaz est présentée par Bell et al. [5]. Gaur and Fisher [13] étudient une version périodique de l'IRP sur une chaîne de supermarchés. En outre, depuis la diffusion du système de gestion des stocks hébergé (Vendor managed Inventory System VMI), la résolution de l'IRP a gagné en intérêt d'un point de vue appliqué [9].

Le problème périodique de tournées de véhicules, qui est intégré dans le problème considéré, est bien connu pour être NP-difficile. Cela prouve que l'IRP est aussi NP-difficile. Bien que les besoins industriels considèrent des instances de taille importante, potentiellement composées par des centaines de détaillants et d'un horizon de planification d'au moins une semaine, il n'est guère réaliste d'envisager le recours à des méthodes exactes. Des exemples sont néanmoins proposés mais ils utilisent des ressources informatiques assez importantes [3, 7, 8].

D'autre part, le problème de tournées de véhicules à plusieurs dépôts (Multi depot vehicle routing problem MDVRP) est fortement lié à l'IRP [23] puisque les deux étudient des conditions réelles des opérations quotidiennes de la chaîne logistique. L'objectif du MDVRP est d'approvisionner un ensemble de détaillants à partir d'un ensemble de dépôts tout en garantissant que: 1) chaque tournée commence et termine au même dépôt, 2) tous les détaillants soient visités une seule fois, 3) la demande totale satisfaite par chaque tournée ne dépasse jamais la capacité du véhicule, et 4) le coût de routage total est minimisé. Le but de l'article est d'étudier le cas général de l'IRP avec plusieurs dépôts (Multi-Depot Inventory Routing Problem MDIRP), en intégrant les conditions décrites pour le MDVRP, et de fournir une méthode heuristique pour le résoudre. Le document est organisé comme suit. Dans la section 3.2, la définition mathématique du problème est fournie, tandis que dans la section 3.3 la méthode heuristique est présentée basée sur un algorithme de recherche locale itérée. L'analyse numérique et les conclusions sont présentées dans les sections 3.4 et 3.5.

D'abord, considérons un ensemble d'etablissements géographiquement dispersées V. Cet ensemble est composé par l'ensemble de dépôts I et l'ensemble des détaillants J ($V = \{I \cup J\}$). L'ensemble des arcs A relie chaque paire de détaillants et de dépôts/détaillants. Considérons un horizon de planification H composé par p périodes ($H = \{1, ..., p\}$). Des périodes factices sont incluses afin de modéliser les conditions initiales et finales des niveaux de stock. Alors, H_0 est défini comme $H \cup \{0\}$ et H' est défini comme $H \cup \{p+1\}$. L' ensemble K est composé par la flotte disponible de véhicules qui sont identiques. Les détaillants doivent satisfaire une demande déterministe et non constante, denotée par d_{jt} pour chaque détaillant $j \in J$ dans la période $t \in H$. Chaque établissement $j \in V$ peut stocker jusqu'à W_j produits et ils sont associés à un niveau de stock initial B_j . A chaque période, le dépôt $i \in I$ est approvisionné avec une quantité fixe r_i. Le coût de possession du produit dans l'établissement $j \in V$ de la période $t \in H_0$ jusqu'à la période $l \in H$ est noté q_{jtl}. Les coûts de transport sont supposés constants au cours de l'horizon de planification. Ainsi, c_{ij} représente le coût d'un véhicule allant du noeud i au noeud j $(i, j \in V)$ et U est le nombre maximum de véhicules à utiliser, chacun avec une capacité de Q unités de produit.

Les décisions qui doivent être optimisées sont: 1) L'affectation de chaque détaillant j à un dépôt i. La variable de décision binaire $f_{ij} = 1$ si le détaillant j sera affecté au dépôt i et cette décision est fixe pour l'horizon de planification; 2) La quantité stockée au dépôt i, dans le premier échelon de la chaîne d'approvisionnement, depuis la période t jusqu'à la période l ($l \in H_0, t \in H$ et $l \geq t$), denoté w_{1itl}; 3) La quantité tenue en stock chez le détaillant j, au deuxième échelon de la chaîne d'approvisionnement, depuis la période 0 jusqu'à la période l , denoté w_{2j0l}; 4) La quantité tenue en stock chez le détaillant $j \in J$, au deuxième échelon de la chaîne d'approvisionnement, depuis la période t jusqu'à la période l approvisionné à partir du dépôt $i \in I$, denoté w_{2ijtl} et 5) les activités de routage par période. Les variables de décision binaires $x_{ijtk} = 1$ si le véhicule k va du noeud i au noeud j dans la période t.

Pour résoudre le problème, une méthode basée sur une recherche locale itérée (ILS) est proposée. Une ILS classique développe des solutions successivement en construisant à chaque itération une solution enfant à l'aide des opérateurs de mutation et de recherche locale sur la meilleure solution trouvée. La solution enfant est accepté selon un critère prédéfini [16]. Cette méthode a été choisie car l'évolution des solutions à travers des perturbations suivies par la recherche locale a été prouvée efficace sur les problèmes de tournées de véhicules [22]. En outre, pour assurer la diversification des solutions trouvées, il est proposé de reprendre la procédure à partir d'une nouvelle solution initiale aléatoire. Cette technique est mise en place pour échapper des minimums locaux [17]. Par conséquent, l'algorithme proposé correspond à la définition d'une recherche locale itérée à plusieurs points de départ ou "multi-start iterated local search" (MS-ILS).

Cet algorithme combine les opérateurs présentés dans les sections 3.3.1 à 3.3.3 pour résoudre le problème. Le pseudo-code de la méthode est présenté dans l'algorithme 4. $N1$ points de départ sont exécutés. A chaque itération, une solution initiale S_0 est obtenue en résolvant le MIP défini dans la section 3.3.1. Cet opérateur est noté `Initial_Sol` à la ligne 3. À la ligne 4, l'opérateur de recherche local (`LS`$(\cdot)$) est appliqué sur la solution initiale S_0. La boucle interne dans les lignes 6-15 est effectuée jusqu'à $N2$ itérations consécutives sans amélioration par rapport à la solution S_0. Cette boucle intérieure commence par appliquer l'opérateur de perturbation à la ligne 7 (opérateur expliqué dans la section 3.3.3) et l'opérateur de recherche locale à la ligne 8. La solution S_0 est remplacée par S' à la ligne 10 si le coût de la solution S' est meilleur que le coût de la solution S_0 $(f(S') < f(S_0))$. La fonction de coût $f(\cdot)$ est calculée avec l'équation (37). La meilleure solution trouvée est sauvée dans S^* à la ligne 16 par la procedure `Save_Best`. Notez que l'opérateur de recherche locale n'est jamais invoqué moins de $N1 \cdot (N2 + 1)$ fois.

Des tests sur les instances de référence dans la littérature pour le cas mono et multi véhicule montrent des résultats compétitifs en trouvant 124 nouvelles meilleures solutions pour les grandes instances avec plusieurs véhicules. En moyenne, les meilleures solutions connues auparavant sont améliorées de 1.05% pour les cas à trois périodes et 1.12% pour les instances à six périodes. Pour le cas multi-dépôt, de nouvelles instances sont proposées et des bornes inférieures sont calculées par une méthode de branchements et coupes. En moyenne, l'écart entre cette borne inférieure et nos solutions est de 9.98% pour les instances avec deux dépôts et de 13.8% pour les instances à trois dépôts.

La recherche future se concentre sur l'amélioration des bornes inférieures fournies et une modification de la méthode pour intégrer les contraintes de la logistique maritime. Dans le transport maritime, les navires n'ont pas des dépôts fixes et les routes peuvent commencer et terminer dans des ports différents. En outre, l'horizon de planification doit être plus long car les opérations portuaires et les temps de navigation sont plus longs, augmentant ainsi l'importance des décisions de stocks. Aussi, le problème doit être résolu pour le cas où les navires composant la flotte ont des différences de capacité, de coût, de temps de voyage, etc. [2]. Ainsi, des travaux récents étudient le problème dans la version à plusieurs dépôts et plusieurs produits. L'industrie pétrolière est un exemple où la distribution est effectuée par les navires et les coûts de chargement et déchargement sont importants. Néanmoins, les coûts de possession de stocks et les décisions de la gestion de stocks aux dépôts sont souvent ignorées [1, 10]. De plus, l'étude de l'effet de la relaxation de la contrainte qui limite l'affectation des détaillants à un seul dépôt peut être aussi intéressante.

References

[1] Agra, A., Christiansen, M., Delgado, A., Simonetti, L., 2013. Hybrid heuristics for a short sea inventory routing problem. European Journal of Operational Research, IN PRESS.

[2] Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A., 2010. Invited review: Industrial aspects and literature survey: Combined inventory management and routing. Computers & Operations Research 37, 1515–1536.

[3] Archetti, C., Bertazzi, L., Hertz, A., Speranza, M., 2011. A hybrid heuristic for an inventory routing problem. INFORMS Journal on Computing, Articles in Advance, 1–16.

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

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

[6] Bertazzi, L., Paletta, G., Speranza, M., 2002. Deterministic order-up-to level policies in an inventory routing problem. Transportation Science 36 (1), 119–132.

[7] Coelho, L., Cordeau, J.-F., Laporte, G., 2012. Consistency in multi-vehicle inventory-routing. Transportation Research Part C: Emerging Technologies 24 (0), 270 – 287.

[8] Coelho, L., Laporte, G., 2013. The exact solution of several classes of inventory-routing problems. Computers & Operations Research 40 (2), 558 – 565.

[9] Disney, S., Potter, A., Gardner, B., 2003. The impact of vendor managed inventory on transport operations. Transportation Research Part E: Logistics and Transportation Review 39 (5), 363–380.

[10] Engineer, F., Furman, K., Nemhauser, G., Savelsbergh, M., Song, J.-H., 2012. A branch-price-and-cut algorithm for single-product maritime inventory routing. Operations Research 60 (1), 106–122.

[11] Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32 (200), 675–701.

[12] Gaudioso, M., Paletta, G., 1992. A heuristic for the periodic vehicle routing problem. Transportation Science 26 (2), 86–92.

[13] Gaur, V., Fisher, M., 2004. A periodic inventory routing problem at a supermarket chain. Operations Research 52 (6), 813–822.

[14] Hansen, P., Mladenović, N., 2001. Variable neighborhood search: Principles and applications. European Journal of Operational Research 130 (3), 449 – 467.

[15] Hvattum, L., Løkketangen, A., Laporte, G., 2009. Scenario tree-based heuristics for stochastic inventory-routing problems. INFORMS Journal on Computing 21 (2), 268–285.

[16] Lourenço, H., Martin, O., Stützle, T., 2010. Iterated local search: Framework and applications. In: Gendreau, M., Potvin, J.-Y. (Eds.), Handbook of Metaheuristics. Vol. 146 of International Series in Operations Research & Management Science. Springer US, pp. 363–397.

[17] Martí, R., 2003. Multi-Start Methods. In: Glover, F., Kochenberger, G. (Eds.), Handbook of Metaheuristics. Vol. 57 of International Series in Operations Research & Management Science. Springer US, pp. 355–368.

[18] Mete, H., Zabinsky, Z., 2010. Stochastic optimization of medical supply location and distribution in disaster management. International Journal of Production Economics 126 (1), 76–84.

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

[20] Moin, N., Salhi, S., Aziz, N., 2011. An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. International Journal of Production Economics 133 (1), 334–343.

[21] Oppen, J., Løkketangen, A., Desrosiers, J., 2010. Solving a rich vehicle routing and inventory problem using column generation. Computers & Operations Research 37 (7), 1308–1317.

[22] Prins, C., 2009. A GRASP× evolutionary local search hybrid for the vehicle routing problem. In: Pereira, F., Tavares, J. (Eds.), Bio-inspired Algorithms for the Vehicle Routing Problem. Vol. 161 of Studies in Computational Intelligence. Springer Berlin Heidelberg, pp. 35–53.

[23] Renaud, J., Laporte, G., Boctor, F., 1996. A tabu search heuristic for the multi-depot vehicle routing problem. Computers & Operations Research 23 (3), 229–235.

[24] Solyali, O., Cordeau, J.-F., Laporte, G., 2012. Robust inventory routing under demand uncertainty. Transportation Science 46 (3), 327–340.

[25] Zhao, Q., Chen, S., Leung, S., Lai, K., 2010. Integration of inventory and transportation decisions in a logistics system. Transportation Research Part E: Logistics and Transportation Review 46 (6), 913–925.

Part II. Second Hybrid Approach: Set-Covering formulation based heuristic

4. A Relax-and-Price Heuristic for the Inventory-Location-Routing Problem

W.J. Guerrero[1,2], C. Prodhon[1], N. Velasco[2,3], C.A. Amaya[2]

[1] ICD-LOSI, UMR CNRS STMR, Université de Technologie de Troyes
12 Rue Marie Curie, CS 42060 10004 Troyes CEDEX, France.
[2] PyLO, Depto de Ingeniería Industrial, School of Engineering,
Universidad de los Andes, Cra 1 # 18A- 12, Bogotá, Colombia
[3] Invited Professor, ICD-LOSI, UMR CNRS,
Université de Technologie de Troyes, France.

This paper considers the problem of designing a supply chain assuming routing decisions. The objective is to select a subset of depots to open, the inventory policies for a 2-echelon system, and the set of routes to perform distribution from the upper echelon to the next by a homogeneous fleet of vehicles over a finite planning horizon considering deterministic demand. To solve the problem, a partition is proposed using a Dantzig-Wolf formulation on the routing variables. A hybridization between column generation, Lagrangian relaxation and local search is presented. Results demonstrate the capability of the algorithm to compute high quality solutions and empirically estimate the improvement in the cost function of the proposed model at up to 9% compared to the sequential approach. Furthermore, the suggested pricing problem is a new variant of the shortest path problem with applications in urban transportation and telecommunications.

Keywords: Inventory-Location-Routing Problem, Lagrangian Relaxation, Column Generation, Supply chain design, vehicle routing problem.

4.1. Introduction

Most of supply chain design problems (SCDP), being a strategic level decision, consider the link between facilities at different levels but not the links between those at a common level. In the case of SCDP models, in order to identify the optimal subset of plants and their location such that logistic costs are minimal, the model graph is restricted to be incomplete by forbidding links between facilities at the same level (e.g. edges connecting two retailers are not allowed). A specific hierarchical order of echelons must be respected [16] and routing decisions are neglected. Nonetheless, when vehicles have enough capacity to deliver more than one retailer per route, this assumption is not suitable.

A variant of traditional SCDP is here presented. The Inventory-Location-Routing problem (ILRP) extends the SCDP by taking into account routing decisions, inventory management policies and their mutual interactions over a multi-period planning horizon. An integrated approach is proposed, given the fact that decomposing the problems of locating facilities, designing inventory policies and finding the optimal set of routes to visit the clients is often suboptimal [6]. This is due to the fact that the search space is truncated by the lack of information sharing between steps when optimizing independently (or sequentially) the location, inventory and routing decisions.

Manzini [15] recently presented an example of this top-down approach proposing a series of mathematical models integrated under a single framework to provide solutions to supply chain design and management problems. This technique is also known as hierarchical optimization since location decisions are optimized at the highest level. Inventory policies are optimized for the given location-allocation scheme afterwards. Finally, routing is solved using a cluster-first route-second approach at the very last step.

More to the point, if the location decision is based on the minimization of the sum of distances (or maximum) between depots and retailers, when vehicles are not performing single-visit tours, optimality is not guaranteed. This statement is verified by Salhi and Rand [23] by testing the effects of ignoring routing decisions when locating depots. Hence, Location-Routing problems (LRP) propose to simultaneously optimize location and routing decisions. Examples are presented by Prins et al. [20, 19], Belenguer et al. [5].

Considering deterministic demand, Ambrosino and Scutellà [2] proposed a linear programming model combining simultaneously depot location, vehicle routing and inventory control policies on a multi-period setting but provided feasible solutions on 12 single-period instances (LRP) with up to 13 depots, 95 retailers, showing that commercial MIP solvers are not able to prove optimality within 25h for most instances exposing empirically the difficulty of the problem.

On the stochastic demand setting, most of the research addressing this issue assume constant demand (Wilson model) and propose an EOQ-like cost in the objective function to include the inventory management component. Papers presented by Ahmadi-Javid and Azad [1], Reza Sajjadi [22], Shen and Qi [24] seek to solve a non-linear LRP. Their proposal is to include in the objective function the annual expected cost of ordering plus holding stock for a random demand which is the non-linear term. On the contrary, the research presented by Liu and Lee [13], Liu and Lin [14] proposed to fix the lot sizes to be equal to expected quantities to deliver in the supply chain for a single period and optimize location-routing decisions from that point of depart. This last approach loses a global perspective as it optimizes sequentially the components of the problem.

An interesting application of the aggregated decision making under incertitude is exposed recently by Mete and Zabinsky [17]. Their research aims to locate emergency stock of medicines and the routes to perform distribution in case of a catastrophe. Their solution method is also hierarchical. After generating several scenarios for the demand of each hospital and the availability of the highways, they use stochastic programming to decide the optimal emergency inventory levels at each opened depot. Based on this information, they use MIP to solve the allocation of hospitals (clients) to depots. On a final step, a set covering problem is solved using MIP to select the routes to distribute product on a single period setting.

This paper deals with the problem of integrating location, inventory, and routing decisions. Consider a 2-echelon supply chain, assuming deterministic demand, and the routing resolution to be made for a discrete and finite planning horizon. In the open literature it is often discussed how strategic level decisions, such as location, should not be integrated with tactical/operational planning. Nonetheless, examples are provided next of situations where integrating routing and inventory management decisions when making location planning is beneficial.

In the first place, the model presented is suitable when location decisions is not made for the long run. It is the case for companies that strategically decide to lease depots and pay rent, signing a rental contract for specific periods of time. The benefit for them is the flexibility to change locations periodically as needed. It is also the case on humanitarian logistics. When a catastrophe happens, emergency response teams set in place facilities to distribute water, medicines and other relief inventories [3]. Often this location requires an investment and it is not supposed to be used to satisfy permanent needs. On the contrary, it is expected to be temporary until the situation is normalized. Finally, it is also the case for military logistics. On the battlefield, bases must be located to store ammunition, supplies or to provide medical support. Military tactics often require these bases to be strongly protected and its location to be changed to minimize the risk of been attacked.

In the second place, the model is also suitable for companies requiring to make better approximations of their operational costs on the long run when locating facilities. It is the case for supply chain designs allowing different frequencies of replenishment at retailers and distribution to be performed by vehicles capable of visiting more than one retailer per route. Industries concerned by integrating these decisions usually face high inventory and distribution costs on the long run when compared to the fixed cost of locating depots and making the assumption of performing single period routing is not realistic enough. Still, depot opening costs should be scaled on the modeled horizon to be in balance with the operational costs.

A Dantzig-Wolfe formulation is proposed on the routing variables, allowing taking apart subtour elimination constraints. Nonetheless, this formulation still contains an exponential set of constraints to force a link between routing and inventory management decisions. These constraints are tackled with Lagrangian relaxation. By doing so, a decomposition in subproblems is possible.

Previous research in vehicle routing problems (VRP) and other combinatorial problems integrating column generation and Lagrangian relaxation techniques show improved computation times and results. Afterall, there is an important link between these two techniques. Geoffrion [8] states that the Lagrangian dual is equivalent to the dual of the continuous relaxation of the Dantzig-Wolfe reformulation. Both examples of such hybridizations are presented by Kallehauge et al. [12] who were able to solve with proven optimality two Homberger VRP instances with time windows (VRPTW) with 400 and 1000 customers, the largest to be solved to date.

Also, Nishi et al. [18] presented their integrated approach using column generation and Lagrangian relaxation for a flow shop scheduling problem. The execution time is reduced with their technique on about 25% for instances with 50 jobs and 3 stages when compared to a pure column generation framework. Large scale problems were solved faster with the hybrid version of the algorithm than with the pure benchmark one. Their conclusions show a high sensitivity between Lagrangian multipliers and the performance of the column generation. In this sense, lots of unnecessary columns are generated if Lagrangian multipliers are far from the optimum.

This research is on a location-routing problem that includes inventory management decisions. The mathematical formulation has two dependent constraint sets with an exponential nature. One set is tackled through Lagrangian relaxation while the VRP constraints are handled with a column generation technique. A heuristic procedure is proposed based on these techniques in order to have better control on computational times.

Section 4.2 presents the problem definition. Section 4.3 is dedicated to the partition principle and the heuristic methodology. The computational experiments are detailed in section 4.4. Analysis and conclusions are given in section 4.5.

4.2. Problem Definition

The considered problem is to decide the location of depots from which a set of routes will depart in order to serve a set of retailers facing deterministic demand over a finite planning horizon. In other words, the issue is to design a two-echelon supply chain comprising the depot location decision, the assignment of each retailer to an open depot, the lot sizes over the planning horizon for each facility (depots and retailers) and the routes per period to perform the distribution activities (dedicated routes to depots, non-dedicated to retailers). The costs include the opening costs, the delivery costs and the inventory costs, including an obsolescence penalty cost.

Let J be a set of n retailers, I the set of m candidate depots, and H the set of p periods in the planning horizon. Retailers face a deterministic non-constant demand $d_{jt}, \forall j \in J, \forall t \in H$. The ILRP is defined on a weighted and directed graph $G = (V, A, C)$. $V = \{J \cup I\}$ is the set of nodes in the graph. C is the cost matrix c_{ij} associated to the traveling cost from node i to node j in the set of arcs A. Figures 7 and 8 present the associated graphs for a traditional SCDP and for the ILRP in a small example with three candidate depots and four retailers. Note that the set of arcs A in G includes the arcs linking every pair of retailers for the ILRP in figure 8 whereas the graph for the SCDP forbids the links between two retailers (figure 7).

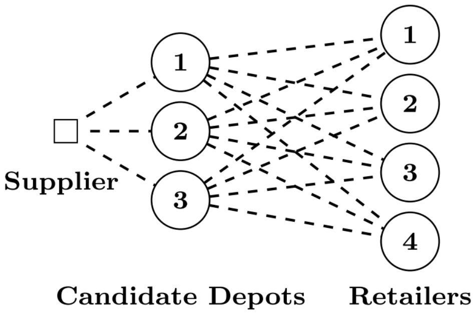

Figure 7: SCDP graph

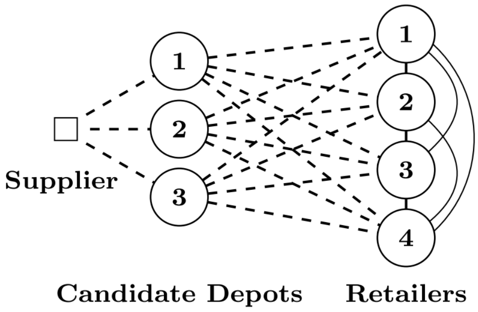

Figure 8: ILRP graph

Each node $j \in V$ is associated to a storage capacity W_j. Each depot $i \in I$ is associated to an opening cost o_i and an ordering cost s_i (dedicated route from the factory).

To tackle inventory management, let $\tilde{G}_i$ be defined as an auxiliary bipartite graph to model inventory decisions for each particular facility $i \in V$. The sets of source nodes $H_0 = \{0\} \cup H$ and destination nodes $H' = H \cup \{p + 1\}$ are included to model initial and final conditions on stock levels. Figure 9 presents an example of $\tilde{G}_i$ considering a three period planning horizon ($p = 3$).

A solution to the inventory management plan at facility i is represented by the flow of product in $\tilde{G}_i$ from every source node $t \in H_0$ to every destination node $l \in H'$. Thus, the flow from t to l is interpreted as stock kept at facility i from period t until period l. $\tilde{G}_i$ is a bipartite graph to permit splitting of the inflow demanded at destination nodes in contrast to the graph proposed by Wagner and Whitin [25].

The flow leaving node 0 must be equal to the initial inventory at i, denoted by B_i, and the flow incoming to node $p+1$ represents the residual stock at the end of the horizon. The edge connecting a source node t to a destination node l is associated to a cost q_{jtl} denoting the holding plus obsolescence penalty cost for one unit of product kept at facility $j \in V$ from period t until period l.

In addition, $\tilde{G}_i$ is incomplete because the edges $\{(t,l),\ \forall t \in H_0, \forall l \in H' | t > l\}$ are not included to forbid backlogging. To illustrate the graph with an example, consider that the flow from delivery node 1 to consumption node 3 to be the quantity kept on stock at j from period 1 to 3 at cost q_{j13}.

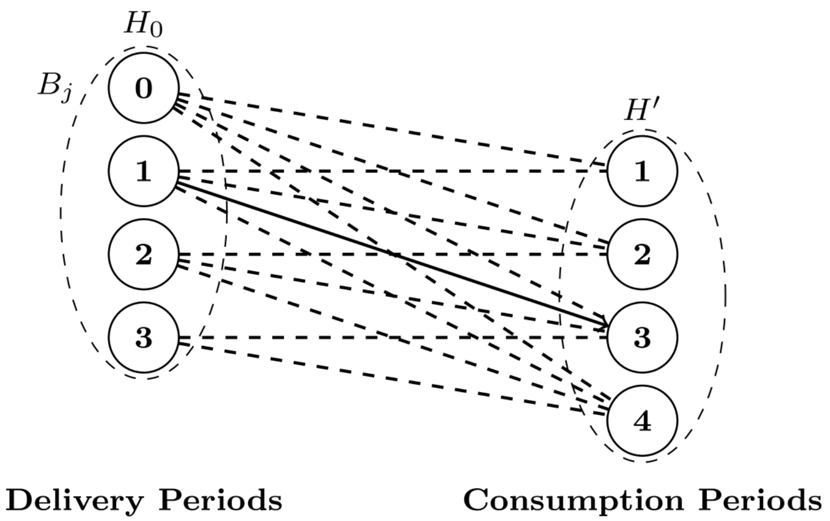

Figure 9: Auxiliary graph $\tilde{G}_i$ for inventory features at facility $i \in V$

To proceed with the routing aspect, consider an unlimited fleet of vehicles with capacity v_{cap} and b as the cost of using a vehicle at least once in the planning horizon. Consider Ω as the set of all feasible routes (sequences of retailers $j \in J \subseteq J$). For each route $r \in \Omega$ and retailer $j \in J$, the associated parameters a_{jr} is equal to 1 iff route r visits retailer j and $\hat{c}_{ir}$ is the cost of the tour r plus the best insertion cost of depot i into r.

Let be the decision variables $y_i = 1$ iff depot $i \in I$ is opened. $f_{ij} = 1$ iff retailer $j \in J$ is assigned to depot $i \in I$, $\theta_{rti} = 1$ iff route $r \in \Omega$ is assigned to depot i for period $t \in H$, and r_i be the maximum number of vehicles used from depot $i \in I$ over H. Inventory decisions at echelon e are denoted by the variable w_e. The quantity replenished from depot i to retailer j on period t to satisfy the demand on period l is denoted by w_{2ijtl}. The quantity of product used from initial stock at retailer j to satisfy demand in period $t \in H'$ is denoted by w_{2j0t}. At the first echelon, $z_{li} = 1$ iff depot $i \in I$ is replenished in period $l \in H$, 0 otherwise. The quantity to replenish in depot $i \in I$ that is delivered in period $t \in H$ to satisfy the demand in period $l \in H'$ is w_{1itl}.

Therefore, the ILRP objective function can be stated as:

$$\min \sum_{i \in I}\left(o_i y_i + br_i + \sum_{l \in H} s_i z_{li}\right) + \sum_{i \in I}\sum_{t \in H_0}\sum_{l=t|l>0}^{p+1} q_{itl} w_{1itl} + \sum_{j \in J}\sum_{t \in H'} q_{j0t} w_{2j0t} + \sum_{i \in I}\sum_{j \in J}\sum_{t \in H}\sum_{l=t}^{p+1} q_{jtl} w_{2ijtl} + \sum_{i \in I}\sum_{t \in H}\sum_{r \in \Omega} \hat{c}_{ir}\theta_{rti} \tag{67}$$

The objective function (67) sums, in this order, the opening costs, the costs of using a vehicle at least once, and ordering costs for every depot in the first term. Holding costs at depots are added in the second term while third and fourth terms add holding costs at retailers. The last term in the objective function sums the distribution costs (the sum of the costs of the selected routes).

Let Ψ be a set of subsets of retailers (all feasible combinations of retailers) indexed to k. Each subset $k \in \Psi$ is associated to a set $S_k \subseteq J$, where the parameter β_{rk} is binary indicating whether route $r \in \Omega$ visits any customer $j \in S_k$. Then, the ILRP is subject to the following constraints:

$$\sum_{i \in I}\sum_{l=1}^{t} w_{2ijlt} + w_{2j0t} = d_{jt} \;\; \forall j \in J, \forall t \in H \tag{68}$$

$$\sum_{l=1}^{t} w_{2ijlt} \leq f_{ij} d_{jt} \;\; \forall i \in I, \;\; \forall j \in J, \forall t \in H \tag{69}$$

$$\sum_{i \in I} f_{ij} = 1, \;\; \forall j \in J \tag{70}$$

$$f_{ij} \leq y_i, \;\; \forall j \in J, \;\; \forall i \in I \tag{71}$$

$$\sum_{l=0}^{t} w_{1ilt} = \sum_{l=t}^{p+1}\sum_{j \in J} w_{2ijtl} \;\; \forall i \in I, \forall t \in H \tag{72}$$

$$\sum_{t \in H'} w_{2j0t} = B_j, \;\; \forall j \in J \tag{73}$$

$$\sum_{t \in H'} w_{1i0t} = B_i \cdot y_i, \;\; \forall i \in I \tag{74}$$

$$\sum_{r=0}^{t}\sum_{l=t}^{p+1} w_{1irl} \leq W_i \cdot y_i \;\; \forall i \in I, \;\; \forall t \in H \tag{75}$$

$$\sum_{l=t}^{p+1}\left(w_{2j0l} + \sum_{r=1}^{t} w_{2ijrl}\right) \leq W_j \;\; , \forall i \in I, \;\; \forall j \in J, \;\; \forall t \in H \tag{76}$$

$$\sum_{l=t}^{P+1} w_{1itl} \leq W_i \cdot z_{ti} \;\; \forall i \in I, \;\; \forall t \in H \tag{77}$$

$$\sum_{l=t}^{p+1} \sum_{j \in S_k} w_{2ijtl} \leq v_{cap} \sum_{r \in \Omega} \beta_{rk} \theta_{rti} \;\; \forall i \in I, \;\; \forall t \in H, S_k \subseteq J, \;\; \forall k \in \Psi. \tag{78}$$

$$\sum_{r \in \Omega} \theta_{rti} \leq r_i \;\; \forall t \in H, \;\; \forall i \in I \tag{79}$$

$$\sum_{i \in I} \sum_{r \in \Omega} \theta_{rti} a_{jr} \leq 1 \;\; \forall t \in H, \;\; \forall j \in J \tag{80}$$

$$\sum_{r \in \Omega} \theta_{rti} a_{jr} \leq f_{ij} \;\; \forall t \in H, \;\; \forall i \in I, \;\; \forall j \in J \tag{81}$$

$$y_i \in \{0, 1\} \;\; \forall i \in I \tag{82}$$

$$z_{it} \in \{0, 1\} \;\; \forall i \in I, \;\; \forall t \in H \tag{83}$$

$$f_{ij} \in \{0, 1\} \;\; \forall i \in I, \;\; \forall j \in J \tag{84}$$

$$\theta_{rti} \in \{0, 1\}, \;\; r \in \Omega, \;\; \forall t \in H, \;\; \forall i \in I \tag{85}$$

$$r_i \in \mathbb{N} \;\; \forall i \in I \tag{86}$$

$$w_{2ijtl} \in \mathbb{R}^+ \;\; \forall i \in I, \;\; \forall j \in J, \;\; \forall t \in H, \;\; \forall l \in H' | l \geq t \tag{87}$$

$$w_{2j0t} \in \mathbb{R}^+ \;\; \forall j \in J, \;\; \forall t \in H' \tag{88}$$

$$w_{1itl} \in \mathbb{R}^+ \;\; \forall i \in I, \;\; \forall t \in H_0, \;\; \forall l \in H | l \geq t \tag{89}$$

Constraints (68) force to satisfy the demand. Each retailer must be assigned and replenished from a single opened depot as stated by constraints (69)-(71). Inventory flow conservation is forced by constraints (72). The sum over the horizon of the quantity kept on stock from period one up to period $p + 1$ is equal to the initial stock as stated by constraints (73)-(74). Capacity for depots and retailers is guaranteed by (75) and (76). Ordering decisions at depots are forced by constraints (77). If a retailer is replenished on period t, it must be visited accordingly by any route departing from the assigned depot. Constraints (78) force this statement along with the limited vehicle capacity. To explain further, these constraints state that the total quantity delivered to a predefined cluster of retailers $k \in \Psi$ at period t, must be at the most v_{cap} times the number of routes that visit that cluster. Next, equations (79) link the cost of using vehicles with the routing decisions. Equations (80) state that each retailer is visited once per period at the most. This constraint could be reinforced by equations (81). Finally, constraints (82)-(89) state the nature of the decision variables.

Further, theorem 1 presents a set of valid inequalities for the presented problem.

Theorem 3. *The set of constraints (90) are valid inequalities for the ILRP.*

$$\sum_{r\in\Omega}\sum_{l=1}^{t}\sum_{i\in I}\theta_{rli}a_{jr} \geq \left\lceil \frac{\sum_{l=1}^{t} d_{jl} - B_j}{v_{cap}} \right\rceil, \; \forall\, j \in J, \, \forall\, t \in H \tag{90}$$

proof. Consider the quantity $\sum_{l=1}^{t} d_{jl} - B_j$ as the total demand faced by retailer j up to period $t \in H$ that can not be satisfied with the initial inventory at j (B_j). In order to satisfy constraints (2), every retailer $j \in J$ must be visited at least $\left\lceil \frac{\sum_{l=1}^{t} d_{jl}-B_j}{v_{cap}} \right\rceil$ times up to period t (right hand). Then, the number of vehicles visiting retailer j up to period t from any depot is defined by $\sum_{r\in\Omega}\sum_{l=1}^{t}\sum_{i\in I}\theta_{rli}a_{jr}$ (left hand) and must be larger the right size as stated by (90). □

4.3. Solution Method

The problem combines two well known NP-hard problems: the SCDP and the VRP. A problem partition is proposed without performing hierarchical or sequential optimization. The suggested pattern combines exact and heuristic procedures leading to a heuristic defined as a matheuristic [21] that will be called as a Relax-and-Price algorithm. In this context, the relaxation of the set of constraints (78) in a Lagrangian fashion allows to decompose the problem into two, which are solvable using column generation. Based on this idea, a heuristic procedure is developed. The partition principle to tackle the problem is explained in section 4.3.1 followed by the solution algorithm and its components detailed in sections 4.3.2-4.3.5.

4.3.1. Partition Principle

The ILRP formulation presented in section 4.2, has an exponential number of θ variables in addition to the exponential number of constraints in equations (78). In practice, this makes the model very hard to solve even for small instances.

The set of constraints (78) link the distribution activities (routing) with the flow of stock through the supply chain, and force to respect the limited vehicle capacity. If these constraints are relaxed, ignoring the reinforcement provided by constraints (81), a relaxed ILRP (RILRP) is obtained. RILRP can be optimized by solving independently a SCDP and a VRP. This follows from the structure, where there is no constraint linking variables θ and w_2 other than equations (78). The RILRP objective function is expressed as:

$$
\min \sum_{i \in I} \left(o_i y_i + br_i + \sum_{l \in H} s_i z_{li} \right) + \sum_{i \in I} \sum_{t \in H_0} \sum_{l=t|l>0}^{p+1} q_{itl} w_{1itl} +
\sum_{j \in J} \sum_{t \in H'} q_{j0t} w_{2j0t} + \sum_{i \in I} \sum_{t \in H} \sum_{l=t}^{p+1} \left(\sum_{j \in J} w_{2ijtl} q_{jtl} + \sum_{k \in \Psi} \sum_{j \in S_k} w_{2ijtl} \mu_{itk} \right) +
\sum_{i \in I} \sum_{t \in H} \sum_{r \in \Omega} \theta_{rti} \left(\hat{c}_{ir} - \sum_{k \in \Psi} v_{cap} \beta_{kr} \mu_{itk} \right) \tag{91}
$$

Subject to: (68)-(77),(79),(80), (82)-(89).

μ_{itk} are the Lagrangian multipliers associated to the set of constraints (78). As explained before, given that variables θ and w_2 are independent in RILRP, the problem can be decomposed into two MIPs. Subproblem 1 handles inventory-location decisions, denoted as ILP1; and subproblem 2 makes the routing decisions, denoted as VRP2. ILP1 and VRP2 are formulated as follows:

$$
ILP1 : \min \sum_{i \in I} \left(o_i y_i + \sum_{l \in H} s_i z_{li} \right) + \sum_{i \in I} \sum_{t \in H_0} \sum_{l=t|l>0}^{p+1} q_{itl} w_{1itl} +
\sum_{j \in J} \sum_{t \in H'} q_{j0t} w_{2j0t} + \sum_{i \in I} \sum_{t \in H} \sum_{l=t}^{p+1} \left(\sum_{j \in J} w_{2ijtl} q_{jtl} + \sum_{k \in \Psi} \sum_{j \in S_k} w_{2ijtl} \mu_{itk} \right) \tag{92}
$$

subject to: (68)-(77),(82)-(84),(87)-(89).

$$
VRP2 : \min \sum_{i \in I} br_i + \sum_{i \in I} \sum_{t \in H} \sum_{r \in \Omega} \theta_{rti} \left(\hat{c}_{ir} - \sum_{k \in \Psi} v_{cap} \beta_{kr} \mu_{itk} \right) \tag{93}
$$

subject to: (79),(80),(85) and (86).

The distinctive challenge of the problem arises from two main differences between VRP formulations based on column generation and the presented one: 1) In the ILRP, a periodic routing problem is tackled where quantities to deliver are decision variables and therefore, the master problem is not modeled as a set covering problem where the solution is forced to visit retailers with fixed frequencies; and 2) In the ILRP, the vehicle capacity constraint is not considered in the subproblem of generating columns with negative reduced cost [7]. The violation of these constraints is penalized in the master problem objective function.

Two issues arise from this decomposition. The first is how to compute the objective functions (92) and (93), given that Ψ is a set that grows exponentially with the number of retailers $|J|$. Then, estimating the Lagrangian multiplier μ_{itk} for every subset $k \in \Psi$ is a complex task. The second issue to solve is how to handle the set Ω for subproblem VRP2.

To solve these issues, axiom 1 is formulated.

Axiom 1. *Since the set Ω is the set of feasible permutations of a subset of retailers and* *the set Ψ is the set of the corresponding combinations of subsets of retailers, therefore, $|\Psi| \leq |\Omega|$.*

The proposed decomposition method is based on column generation. Thus, the ILRP can be restrained to consider only a subset of feasible routes $\Omega' \subseteq \Omega$. Given this subset of permutations of retailers Ω', the corresponding subset of combinations $\Psi' \subseteq \Psi$ is computed. This restrained version will be denoted as Rv-ILRP. Further, it would be easy to show using axiom 1, that limiting the set of constraints (78) to the set Ψ' for Rv-ILRP provides a feasible solution if each retailer is visited at least by a single route in Ω'. Then, it is proven that the set of constraints (78) corresponding to the set $\Psi \setminus \Psi'$ is dominated by the set of constraints (78) corresponding to the set Ψ'.

Similarly, by restraining the RILRP to consider only the sets Ψ' and Ω' instead of Ψ and Ω respectively, a restrained-version RILRP (Rv-RILRP) is obtained and the optimal solution is a lower bound on Rv-ILRP. Moreover, considering that $|\Psi'| \leq |\Omega'|$, the computation of equations (92) and (93) becomes easier.

It is proposed to start by solving Rv-RILRP with elementary routes on the pool of routes Ω'. Interesting columns (routes) are going to be dynamically added into Ω'. The corresponding combination of retailers is going to be added into Ψ' to maintain the principle exposed previously. In the following section, this principle is used to develop the proposed Relax-and-Price heuristic.

4.3.2. Relax-and-Price Algorithm

Algorithm 5 presents the general procedure proposed to find solutions to the ILRP. For notation, consider that S^* represents the best found solution, S the incumbent solution, and $\hat{S}$ an unfeasible solution. The operator C($\cdot$) returns the cost of a solution, C($\emptyset$) returns a very large number, and C($\vec{0}$) returns zero. At line 1 and 2, the algorithm is initialized with subset $\Omega' \subset \Omega$ of dedicated routes to retailers. Subset $\Psi' \subset \Psi$ is also generated considering the combinations of retailers for every route in Ω'. The Lagrangian multipliers μ_{itk} are initialized to zero.

For a fixed number of iterations, in lines 3 to 16, the algorithm starts by adding to the set Ω' new columns in line 5. The procedure GENERATE_COLUMNS iterates by solving the LP relaxation of VRP2, to obtain the optimal values of dual variables associated to constraints (79), (80) and (90). These are required in the pricing procedure to compute new columns to be added. More details are given in section 4.3.3. GENERATE_COLUMNS iterates until no further columns with negative reduced cost are found.

In line 7, once new promising routes are included into Ω' along with their associated decisions variables θ, Rv-RILRP described in section 4.3.1 is solved using a MIP solver providing an unfeasible solution $\hat{S}$. In line 8, $\hat{S}$ is repaired by the procedure CORRECTION_PROCEDURE to guarantee the feasibility in the solution S. This procedure will be described in section 4.3.4 while section 4.3.5 details the local search operator that improves S at line 9. Lines 10-12 store the best found solution.

The subgradient method is proposed to update Lagrangian multipliers in line 13 through equations (94) to (96) as in Beasley [4]. The method computes the gap between the best feasible solution C(S^*) and the current lower bound C($\hat{S}$) to set a step size $\delta^{(p)}$ at the iteration p by equation (95). The direction of the multipliers $\vec{\mu}$ is also

Algorithm 5 Main Algorithm (Overview)

1: $S^* \leftarrow \emptyset$; $\vec{\mu} \leftarrow \vec{0}$; //Initialization
2: $\Omega' \leftarrow$ ELEMENTARY_ROUTES;
3: **for** $(i = 0;\ i < N_1;\ i++)$ **do**
4: $\quad \hat{S} \leftarrow \vec{0}$;
5: $\quad \Omega' \leftarrow \Omega' \cup$ GENERATE_COLUMNS($\vec{\mu}$) //Solve the pricing problem
6: $\quad$ **while** (C($\hat{S}$) $<$ C(S^*) or N_2 iterations) **do**
7: $\quad\quad \hat{S} \leftarrow$ SOLVE_RV_RILRP($\vec{\mu}$); //Get unfeasible solution
8: $\quad\quad S \leftarrow$ CORRECTION_PROCEDURE($\hat{S}$); //Repair solution
9: $\quad\quad S \leftarrow$ LOCALSEARCH(S);
10: $\quad\quad$ **if** (C(S) $<$ C(S^*)) **then**
11: $\quad\quad\quad S^* \leftarrow S$;
12: $\quad\quad$ **end if**
13: $\quad\quad \vec{\mu} \leftarrow$ SUBGRADIENTMETHOD($\vec{\mu}, \hat{S}, S^*$); //Upgrade multipliers
14: $\quad$ **end while**
15: $\quad$ ADD_RANDOM_CUTS;
16: **end for**

corrected with equation (96) which penalizes the relaxed constraints (14). The new $\vec{\mu}$ coefficients are computed by equation (94).

$$\mu_{itk}{}^{(p)} = \max\{0, \mu_{itk}{}^{(p-1)} + \delta^{(p)}_{i,t,k}\nu^{(p)}_{i,t,k}\}\ \forall i \in I,\ \forall t \in H,\ \forall k \in \Psi' \tag{94}$$

$$\delta^{(p)} = \frac{(C(S^*) - C(\hat{S}))}{\|\nu^{(p)}\|} \tag{95}$$

$$\nu^{(p)}_{i,t,k} = \sum_{l=t}^{p+1}\sum_{j\in S_k} w_{2ijtl} - v_{cap}\sum_{r\in\Omega'}\beta_{rk}\theta_{rti} \tag{96}$$

The cost of $\hat{S}$ works as a lower bound to Rv-ILRP, while the cost of the best found solution S^* is the best upper bound. Lines 6 to 14 are repeated until $C(\hat{S})$ is larger than $C(S^*)$, in which case it is required to add more routes to Rv-RILRP. Finally, in line 15, random cuts are included in the MIP formulation of Rv-RILRP. Two random retailers $R_1, R_2 \in J$ are forced to be assigned to their corresponding closest depot D_{R_1} and $D_{R_2} \in I$ ($f_{R_1,D_{R_1}} = 1, f_{R_2,D_{R_2}} = 1$). These cuts are deleted after the Lagrangian multipliers are updated $\vec{\mu}$ if no improvement is found in line 13. By perturbing $\vec{\mu}$ for some iterations, some diversification is induced in the search.

4.3.3. Pricing Problem

In order to identify interesting routes to be dynamically included into Ω', the following dual problem to the continuous relaxation of VRP2 is formulated. λ_{1it}, λ_{2jt} and λ_{3jt} are defined as the dual variables associated to constraints (79),(80), and (90) respectively.

$$\max \sum_{j \in J} \sum_{t \in H} (\bar{b}_{jt} \lambda_{3jt} - \lambda_{2jt}) \tag{97}$$

Subject to:

$$\sum_{t \in H} \lambda_{1it} \leq b, \;\; \forall \; i \in I \tag{98}$$

$$\sum_{j \in J} \left[a_{jr} \left(-\lambda_{2jt} + \sum_{l=t}^{p} \lambda_{3jl} \right) \right] - \lambda_{1it} \leq \bar{c}_{rti}, \; \forall r \in \Omega, \; \forall \, t \in H, \; \forall \, i \in I \tag{99}$$

$$\lambda_{1it} \geq 0, \lambda_{2jt} \geq 0, \lambda_{3jt} \geq 0, \forall i \in I, \forall j \in J, \forall t \in H \tag{100}$$

Where $\bar{b}_{jt}$ is equal to the right-hand size of equations (90) and $\bar{c}_{rti}$ is the cost coefficient of θ_{rti} in equation (93): $\left(\hat{c}_{ir} - \sum_{k \in \Psi'} v_{cap} \beta_{kr} \mu_{itk}\right)$. Ideally, it is required to add as many columns with negative reduced cost as possible to solve to optimality the master problem (VRP2). New columns can be generated by solving the associated pricing problem.

The presented formulation of the pricing problem is interpreted as a generalized elementary shortest path problem (GESPP) for each period $t \in H$ and depot $i \in I$. Further explanation is provided next:

It is natural to state that $a_{jr} = \sum_{u \in J \cup \{i\}} x_{ujr}$ into equation (99). Therefore, the reduced cost of a route $r \in \Omega$ for period t assigned to depot i is equal to:

$$\sum_{u \in J \cup \{i\}} \sum_{v \in J \cup \{i\}} x_{uvr} \cdot \left[c_{uv} + \lambda_{2jt} - \sum_{l=t}^{p} \lambda_{3vl} \right] + \lambda_{1it} - \sum_{k \in \Psi'} v_{cap} \beta_{kr} \mu_{itk} \tag{101}$$

Now, recall that vehicle capacity constraints (78) were relaxed and therefore, the pricing problem has no resource constraints. The only difference between the shortest path problem objective function and equation (101) is the last term. It is interpreted as a profit obtained if the selected path visits the predefined clusters $k \in \Psi'$.

The GESPP is studied by Guerrero et al. [10]. Its purpose is to find the minimum reduced cost path for a fixed depot i and period t. Consider the set of retailers J to be aggregated in non-disjoint clusters where each cluster $k \in \Psi'$ is associated with a profit p_k to the cost function equal to the corresponding coefficient $v_{cap} \mu_{itk}$. The objective is to find the minimum cost path from a dummy node $\{0\}$ to a sink dummy node $\{n+1\}$ while visiting a subset of retailers. Both dummy nodes $\{0, n+1\}$ represent the depot i. The profit could be interpreted as a marginal decrease in the Lagrangian term in the objective function of Rv-RILRP. Additionally, let the GESPP be defined over a complete weighted graph with $\tilde{c}_{ij}$ the cost of connecting node j after node i in the path. $\tilde{c}_{ij}$ can be negative and the graph may contain negative cycles.

Figure 10 presents the graph for an example of the GESPP considering nine retailers. Nodes 0 and 10 are the source and sink of the problem to represent the depot. For the sake of simplicity, consider four clusters only (C_1 to C_4). $C_1 = \{1, 2, 4, 5\}$, $C_2 = \{2, 3\}$, $C_3 = \{7, 8\}$, and $C_4 = \{5, 6, 8, 9\}$. Each cluster k is associated to a profit p_k. Then, the

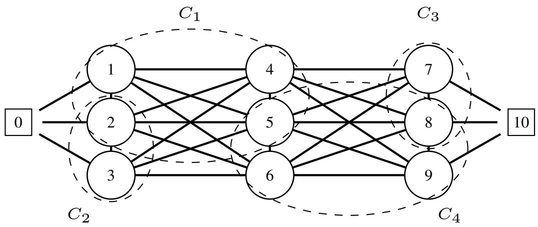

Figure 10: Graph for the GESPP - an example

path $\{0-2-4-7-10\}$ would have a cost equal to $\tilde{c}_{0,2}+\tilde{c}_{2,4}+\tilde{c}_{4,7}+\tilde{c}_{7,10}-p_1-p_2-p_3$. Since any retailer from cluster C_4 is visited, the profit p_4 is not included. Also, p_1 is added only once despite the fact that two retailers from cluster C_1 are visited (retailer 2 and 4).

To illustrate the mathematical formulation of GESPP, let x_{ij} be a binary decision variable indicating whether the arc (i, j) belongs to the shortest path; let y_k be a binary variable equal to 1 iff cluster $k \in \Psi'$ is visited at least once. Let $\tilde{c}_{ij}$ be the cost of using arc (i, j) and p_k the profit for visiting cluster k. Let the GESPP be formulated as follows:

$$\text{GESPP:} \quad min \sum_{i \in J \cup \{0\}} \sum_{j \in J \cup \{n+1\}} c_{ij} x_{ij} - \sum_{k \in \Psi} y_k p_k \tag{102}$$

Subject to:

$$\sum_{i \in J} x_{0,i} = 1 \tag{103}$$

$$\sum_{i \in J} x_{i,n+1} = 1 \tag{104}$$

$$\sum_{i \in J \cup \{0\}} x_{ij} - \sum_{i \in J \cup \{n+1\}} x_{ji} = 0 \,, \forall j \in J \tag{105}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \,, \forall\, S \subseteq J \tag{106}$$

$$\sum_{i \in S_k} \sum_{j \in J \setminus \{S_k\}} x_{ij} \geq y_k \ \ , \forall\, k \in \Psi \tag{107}$$

$$x_{ij} \in \{0,1\} \ \forall\, i \in J \cup \{0\}, \forall\, j \in J \cup \{n+1\}, y_k \in \{0,1\} \ \forall\, k \in \Psi \tag{108}$$

Equation (102) presents the objective function. It is the minimization of the total path length value after subtracting the corresponding cluster profits. Constraints (103) and (104) force the path to start and end at nodes 0 and $n + 1$ respectively. Equations (105) force flow conservation while equations (106) are traditional subtour elimination constraints. Additionally, the set of constrains (107) states that the cluster profits are obtained if the path visits any retailer belonging to the corresponding cluster.

To solve the GESPP, Guerrero et al. [10] presents a truncated labeling heuristic algorithm also used in this Relax-and-Price procedure. The computation of a path requires each retailer j to be associated with a set of labels representing the non-dominated paths from node 0 up to j. Each label keeps track of the visited clusters and visited retailers. This way, the algorithm enumerates the paths from 0 up to every node in the graph keeping only non-dominated labels. To speed up the search, a limit of K labels per retailer is imposed, making the procedure to be heuristic. The algorithm stops when all the existing labels have being extended to unvisited retailers. A local search procedure is performed as post-optimization with the following traditional neighborhoods:

- Exchange: Modifies the position of a retailer in the path.
- Swap: Interchanges the position of two retailers in the path.
- 2-Opt: Erases two arcs in the path and reconnects it with two different arcs such that the path is still feasible.
- 3-Opt: Erases three arcs and reconnects the path with three different arcs in the best possible way.
- Insert: Insert an unvisited retailer into the path.

4.3.4. Repairing Operator

As the constraints linking distribution and replenishment (78) were relaxed on Rv-RILRP, three cases might make a solution infeasible: 1) Retailers that are visited without being replenished; 2) Replenished retailers without scheduled visit on a particular period (no vehicle visits the retailer); 3) Overloaded routes. In the first case, the visit is simply removed. In the second case, a best-insertion procedure is performed. In the third case, the route is divided into two different routes. The point of division of the original route is computed as the point where the insertion of the depot is performed at the minimum cost. This procedure is repeated until no further routes violate the vehicle capacity constraint.

4.3.5. Local Search

Once a feasible solution is found, a local search in the form of a variable neighborhood descent (VND) [11] is performed to intensify the search. Several neighborhoods are explored to improve routing, inventory costs and location-allocation decisions, using a first improvement movement strategy, in the following order:

- *Routing neighborhoods:* These neighborhoods are limited to evaluate changes for scheduled visits to retailers sharing the same depot and period.
  - Move: The visit of a retailer is shifted from its current position to a different position.
  - Swap: The positions of two different retailers is exchanged.
  - 2-Opt: Two arcs are removed from the solution and new arcs are included to assure feasibility. The removed arcs might belong or not to the same route.

- *Inventory Neighborhoods:* These neighborhoods are limited to evaluate changes for a single retailer at a time.
  - Deplete Stock: Consider a particular retailer j, and two consecutive replenishments at periods t_0 and t_1 with quantities Q_{t_0} and Q_{t_1} respectively. The stock level is reduced by reducing the quantity delivered Q_{t_0} subject that the quantity Q_{t_1} is increased accordingly to guarantee demand fulfillment.
  - Remove Visit: Once more, consider a particular retailer j, and two consecutive replenishments at periods t_0 and t_1 with quantities Q_{t_0} and Q_{t_1} respectively. The replenishment in period t_1 is removed if Q_{t_0} can be sufficiently raised (by at least Q_{t_1} units) to satisfy future demand. Routing costs decrease while inventory holding costs increase.
- *Location-Allocation Neighborhoods:* These neighborhoods consider the scheduled visits to remain unchanged. Routing and depot inventory policies are revisited.
  - Depot Reallocation: A retailer is re-allocated to a different depot.
  - Depot Allocation Swap: Two retailers are exchanged in their depot allocation.

4.4. Computational Study

The algorithm is coded in C and the MIP model is solved with Xpress-IVE 7.0. Tests are performed on an Intel Xeon with 2.80Ghz processor and 12 GB of RAM. 20 ILRP random instances were generated with the following size: $m : \{5\}$ depots, $n : \{5, 7\}$ retailers, $p : \{5, 7\}$ periods. They are labeled as $m - n - p - x$ and x is used to itemize.

Demand at retailer j for period t is $d_{jt} \sim N(\mu_j, \sigma_j)$, were $\mu_j \in [5, 15]$ and $\sigma_j \in [0, 5]$. The opening costs o_i is generated with a Normal distribution with parameters (μ_i,σ_i) chosen from the set of pairs $\{(1000, 20)$, $(5000, 100)$, $(8000, 300)\}$. s_i is chosen from the set $\{100, 500\}$. The coordinates (X_i, Y_i) for facility $i \in V$ are randomly generated in a square of size 100×100. Transportation cost c_{ij} is equal to the closest integer of a hundred times the euclidean distance from i to j. v_{cap} is a random integer in the interval $[15, 75]$. The cost b is selected from the set $\{350, 1000, 5000\}$. Depot capacity W_i is randomly generated in the interval $[D/3, D]$, were $D = \sum_{j \in J} \sum_{t \in H} d_{jt}$. Retailer's capacity W_j are randomly generated in the interval $[g_j, 3 \cdot g_j]$ were $g_j = max_t\ d_{jt}$. Initial inventories B_j were chosen from the set $\{0, d_{j1}\}$ for retailers and B_i from the set $\{0, 10 \cdot D/n\}$ for depots. Inventory holding costs for a single period $t \in Ho$ at retailers and depots $j \in V$, $q_{j,t,t+1}$ is generated in the interval [0.03,0.50]. The inventory holding costs for k periods as $q_{j,t,t+k} = \sum_{l=t}^{t+k-1} q_{j,l,l+1} + k \cdot \xi_2$, were $\xi_2 \sim Unif[0.01, 0.02]$ represents the obsolescence penalty cost per period.

Results of the presented heuristic are compared to the three alternative methodologies listed below:

SOLVER: A feasible solution is obtained from solving a MIP model using a commercial solver imposing a time limit of 2.5 hours for instances with 5 retailers and 9 hours for instances with 15 retailers. This model is presented at Guerrero et al. [9].

H1: A constructive heuristic method with two phases following an intuitive idea is explored. This approach is based on hierarchical optimization where location decisions are fixed before optimizing inventory-routing decisions. On the first phase, a SCDP is solved using a commercial solver which fixes the location of depots and decides an initial allocation of retailers to depots together with the inventory decisions. On a second stage, an iterative local search to tackle inventory-routing decisions is performed.

H2: A cooperative heuristic is also compared. $H2$ has two main components that share information. The first component solves the SCDP using a commercial solver just as in $H1$ but uses information of the estimated distribution costs computed by the second component. For a fixed number of iterations, the second components destroys and repairs inventory-routing decisions to obtain complete solutions based on the supply chain design fixed by the first component. By alternating between solution spaces, both components optimize simultaneously the different decisions. An iterative local search procedure is performed as post-optimization to intensify on allocation decisions. This heuristic is proposed by Guerrero et al. [9].

To make a fair comparison between the proposed relax-and-price algorithm and $H2$, both are run in the same workstation, both using the same MIP solver. Additionally, a limit of 450 evaluations of the objective function or calls to the local search operator is imposed on both heuristics. Tunning test showed that the best results for the relax-and-price algorithm are obtained when $N_1 = 10$ and $N_2 = 45$.

Table 14 presents the results of comparing the proposed methodology versus the three heuristics described before. The instance labels are presented in column one, column two presents the value for the best know solutions (BKS) out of the four methods. So far, any instance has being solved to proven optimality. Columns three to ten present the solution quality of the corresponding heuristic computed as $gap = 100 \cdot (Cost - BKS)/BKS$ and the computation time in seconds (CPU). If the computation time is highlighted in bold font, the corresponding method is the fastest among all compared algorithms. Columns three and four present the gap between the best feasible solutions found by the solver at the first 60 seconds (gap^0) and after the imposed time limit (gap^1). For instances with $n = \{5, 7\}$, the time limit is 2.5 hours while for instances with $n = \{15\}$ the time limit is 9 hours. This comparison is provided since it is common for commercial solvers to find good solutions quickly and spend important computation time trying to prove optimality.

Columns five to ten present the average performance of the presented relax-and-price algorithm, $H1$ and $H2$ for three executions. Results show an average gap to BKS of 0.47% computed in 887 seconds while $H2$ computes solutions with a gap of 0.58% within more than 2000s. Furthermore, $H1$ provides consistently good solutions. 8/20 BKS are found by the relax-and-price algorithm and 17/20 are always below 1%.

The MIP solver is not capable of finding feasible solutions for two out of the five instances with 15 retailers (5-15-5-c & 5-15-5-e) and 5 periods within 9 hours. Similarly, the solver is incapable of finding a feasible solution for one instance with seven retailers and 5 periods (5-7-5-c) within 2.5 hours. Thus, the average gap to BKS is 25% within the first 60s and 13.7% within the time limit.

On the other hand, $H1$ is the fastest heuristic but provides solutions with a gap of 4.14%. The largest instances with 15 retailers are computed with a gap up to 9%. Hierarchical optimization in this case, where location decisions are fixed before optimizing

Instance	BKS	*SOLVER*		*relax-and-price*		*H1*		*H2*	
		gap^0	gap^1	gap	CPU	gap	CPU	gap	CPU
5-5-5-a	93625.3	3.84	0.1	0	32.4	0	**4.7**	0	25.2
5-5-5-b	62206.9	10.98	0.46	0	30.1	1.12	**4**	0	22.4
5-5-5-c	69760.3	23.9	0	0.14	34.6	1.73	**7.9**	1.61	48
5-5-5-d	93451.2	4.25	0.37	0.8	32.4	0	**4.7**	0	28.6
5-5-5-e	93851	4.3	0	0.25	35.3	4.2	**4.3**	0.8	14.1
5-5-7-a	70966.5	19.65	9.07	0	103.4	7.05	**89.3**	0	320.7
5-5-7-b	107478.5	55.65	3.22	3.22	90.1	4.49	**46.7**	0	394.9
5-5-7-c	94150.2	28.88	0	0	**128.4**	6.66	212	0	328.3
5-5-7-d	87744.2	13.79	0	0	106.8	4.57	**8.8**	0	62.1
5-5-7-e	67275.4	-	2.6	0	108.6	5.57	**44.8**	0	176.4
5-7-5-a	68485.2	110.09	0	0.73	150.6	4.44	**7.9**	1.83	55
5-7-5-b	76339.1	-	0	1.31	90.4	3.65	**51**	3.04	114.8
5-7-5-c	138998	-	-	0.07	**237.1**	2.49	1409	0	4964.9
5-7-5-d	99988.9	-	0	0.51	166.7	6.6	**26.2**	0.01	222.3
5-7-5-e	62010.1	48.66	0	1.16	97.5	2.33	**11.5**	0.36	81.2
5-15-5-a	113434.3	-	38.37	0	1502.8	9.65	**101.8**	0	1863.5
5-15-5-b	172743.3	-	35.09	0.58	1524.1	2.87	**443.6**	0	2001.3
5-15-5-c	210333	-	-	0.48	10240.1	1.2	**4076.9**	0	14301.9
5-15-5-d	165939.7	-	142.99	0.05	1463.5	6.35	**330.2**	0	1530.7
5-15-5-e	219634.4	-	-	0	1573.6	7.82	**4880.7**	4.02	22661.3
Average	102599.8	29.45	13.66	0.47	887.43	4.14	588.3	0.58	2460.88

Table 14: Comparative performance on ILRP instances.

routing decisions does not provide the best average solution quality.

Fig. 11 summarizes the average computational time for the Relax-and-Price, H1 and H2. Instances are classified according to its size, computed as $n \cdot m \cdot p$. As shown, the computational times required by H2 grow significantly faster than H1 and the Relax-and-Price procedure. Again, H1 is the method with the smallest average computational times. Further, it is natural to see that all compared methodologies have non-polynomial computational complexity since they are all hybrid methods solving combinatorial sub-problems with exact methodologies.

Further, the impact of removing each component is evaluated. Three components are removable without affecting the stability of algorithm 5: the local search procedure discussed in section 4.3.5 performed in line 9; the inclusion of random cuts performed in line 15; and the local search for the pricing problem discussed in section 4.3.3 when solving the GESPP at line 5. Table 15 presents the results for average gap to BKS and computational time in seconds for the different variants of algorithm 5 when removing a particular component.

The operator with the largest impact in solution quality is the local search procedure. As a matter of fact, the solution cost is increased by about 42% when removing it. The convergence of the method is also disturbed since the average computational time is increased to 974s. This is explained by the fact that lagrangian multipliers are updated with slower convergence if the gap between the best feasible solution and the lower bound is larger.

Removed Component	avg gap (%)	cpu[s]
Local Search	42.35	974.36
Random Cuts	12.87	1106.42
Local Search on GESPP	24.46	1095.24

Table 15: Impact on solution quality and cpu when removing components from algorithm 1.

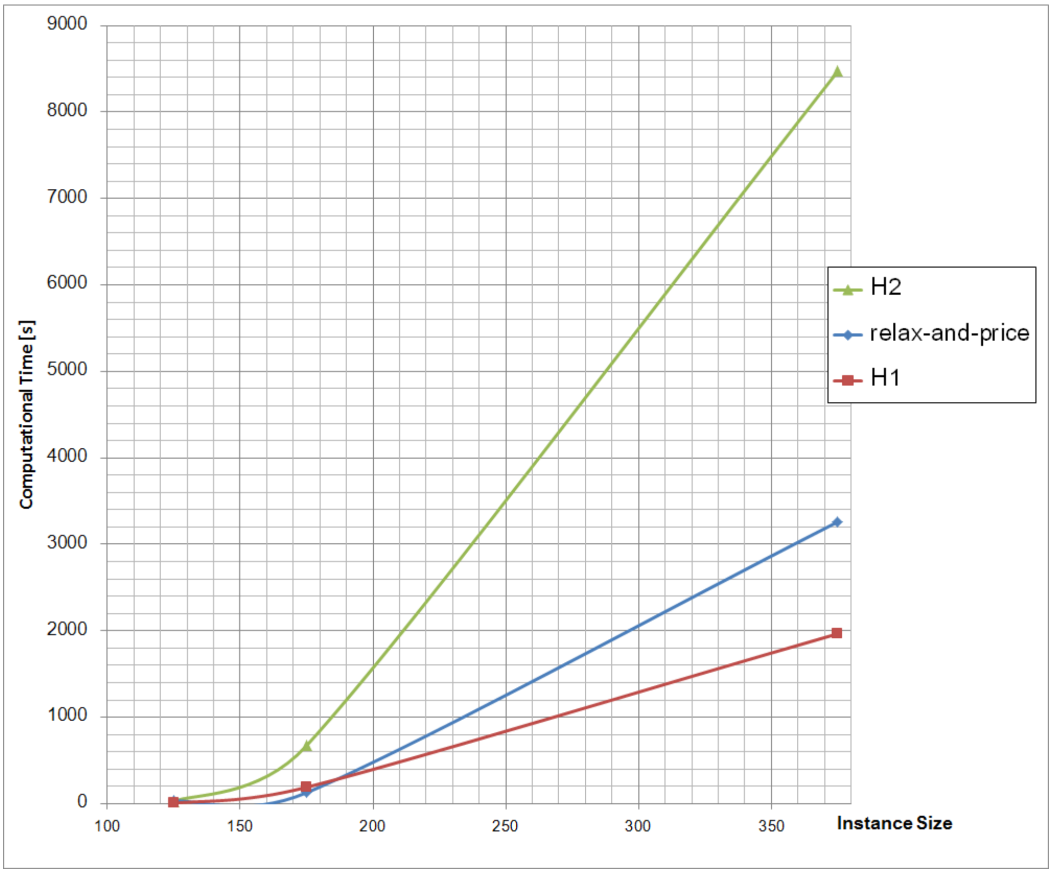

Figure 11: Growth in Computational times for the different solution approaches

Also, the random cuts are usefull to help the procedure to converge faster. By adding these cuts, the procedure SOLVE_RV_RILRP is solved faster since some decision variables are fixed. Nonetheless, the solution quality is decreased on average about 12% while the computation time is significantly larger. Finally, the local search procedure that is applied when solving the pricing problem also has an important impact on solution quality and computational time. The quality of the columns in the pool Ω' is naturally decreased, resulting on solutions 24% more expensive than the best known solutions computed within 1095 seconds on average.

4.5. Conclusions

A Lagrangian relaxation based heuristic methodology to solve the Inventory-Location-Routing Problem is proposed, including the ideas of column generation for vehicle routing. The target is to simultaneously optimize a supply chain design considering inventory and routing decisions without decomposing the problem into sub problems in order to optimize globally.

The presented methodology combines a non-traditional column generation approach with a Lagrangian relaxation within a framework that is denoted as relax-and-price. The challenge is to coordinate the generation of new columns in the pricing problem and the update of Lagrangian multipliers. In addition, to compute interesting routes, a shortest path problem with cluster profits, the generalized elementary shortest path problem, is solved through a two- phase algorithm.

Results for randomly generated instances show important cost savings over the traditional approach and efficient computation if compared to commercial solvers and other benchmark heuristics. The impact of each component of the algorithm is evaluated. Future research focuses on a multi-objective version of the problem and applying stochastic programming techniques in order to include the uncertainty associated to demand.

Acknowledgement

This research is supported by the Champagne-Ardenne Regional Council (France), Centro de Estudios Interdisciplinarios Básicos y Aplicados - CEIBA (Colombia) and ICD-LOSI, Université de Technologie de Troyes. We thank two anonymous reviewers that made valuable suggestions to help improve the manuscript.

Chapter 4 was submitted as:

W.J. Guerrero, N. Velasco, C. Prodhon, C.A. Amaya (2013) *A Relax-and-Price Heuristic for the Inventory-Location-Routing Problem.* International Transactions in Operations Research. IN SUBMISSION

Preliminary results were presented at:
W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya (2013) A Relax-and-Price Algorithm for the Inventory-Location-Routing Problem. TRISTAN VIII Conference, San Pedro de Atacama, Chile.

W.J. Guerrero, C. Prodhon, N. Velasco, C.A. Amaya (2012) Lagrangian Relaxation – Column Generation Approach for the Inventory Location-Routing Problem. Fourth International Workshop on Model Based Metaheuristics, Matheuristics. Sept 17-20. Angra dos Reis, Brasil.

4.6. Résumé en français

La plupart des problèmes de conception de la chaîne d'approvisionnement (Supply chain design problem SCDP), considèrent un lien entre les établissements d'échelons différents (décisions stratégiques), mais pas les liens entre ceux au même niveau (décisions tactiques et opérationnelles). Dans le cas des modèles pour le SCDP, le but est d'identifier le sous-ensemble optimal des dépôts et leur localisation tel que le coût logistique soit minimisé. Le graphe utilisé interdit les liens entre établissements du même niveau. Un ordre hiérarchique précis d'échelons doit être respecté [16] et les décisions de routage sont négligés. Néanmoins, lorsque les véhicules ont une capacité suffisante pour fournir plus d'un détaillant par tournée, cette hypothèse n'est pas valable.

Une variante du SCDP traditionnel est ici présentée. Le problème de Localisation-Routage avec gestion de stock (Inventory-Location-Routing Problem ILRP) étend le SCDP en tenant compte des décisions de routage, l'optimisation de la gestion de stocks et leurs interactions sur un horizon de planification à plusieurs périodes. Une approche intégrée est proposée, compte tenu du fait que la décomposition du problème dans les étapes de localisation des dépôts, de l'optimisation de gestion de stocks, et la recherche de l'ensemble optimal de tournées est une approche qui peut fournir des solutions globalement sous-optimales [6].

Manzini [15] a récemment présenté un exemple de cette approche dite "top-down" en proposant une série de modèles mathématiques intégrés dans un cadre unique pour fournir des solutions au problème de conception de la chaîne d'approvisionnement et aux problèmes de gestion de stocks et tournées de vehicules. Cette technique est également connue comme une méthode d'optimisation hiérarchique puisque les décisions de localisation sont optimisées au plus haut niveau. La gestion de stocks est optimisée en considérant fixes les décisions de localisation et l'affectation des détaillants aux dépôts. Finalement, le problème de routage est résolu en utilisant une méthode de "route-first, cluster second" dans la dernière étape.

Une application intéressante du problème d'optimisation intégré avec incertitude est exposée récemment par Mete and Zabinsky [17]. Leur recherche vise à faire la gestion de stock d'urgence de médicaments et les routes pour effectuer la distribution en cas de catastrophe. La méthode de solution proposée est également hiérarchique. Après avoir généré plusieurs scénarios pour la demande de chaque hôpital et la disponibilité des routes, ils utilisent la programmation stochastique pour décider les niveaux de stocks d'urgence optimals à chaque dépôt ouvert. Sur la base de cette information, ils utilisent un modèle d'optimisation linéaire en nombres entiers pour résoudre l'affectation des hôpitaux (clients) aux dépôts. Sur une dernière étape, un problème de recouvrement est résolu afin de sélectionner les routes pour distribuer un produit sur un horizon de planification à une seule période.

Plus précisément dans les méthodologies traditionnelles, si la décision de localisation est basée sur la minimisation de la somme (ou de la valeur maximale) des distances entre les dépôts et les détaillants, lorsque les véhicules font des livraisons à plusieurs clients par tournée, l'optimalité de la solution n'est pas garantie. Cette affirmation est vérifiée par Salhi and Rand [23] qui testent les effets d'ignorer les décisions de routage lors de la localisation des dépôts. Par conséquent, les problèmes Localisation-Routage (LRP) proposent d'optimiser simultanément l'emplacement et les décisions de routage.

Des exemples sont présentés par Belenguer et al. [5], Prins et al. [19, 20].

Ce chapitre traite la question de l'intégration de la localisation, la gestion de stocks et les décisions de routage avec une formulation du type "set covering" ou recouvrement d'ensembles. Considérons une chaîne d'approvisionnement à deux échelons. Nous posons l'hypothèse que la demande soit déterministe et que la résolution du problème de routage doive être faite pour un horizon de planification discret et fini.

Dans la littérature, il est souvent discuté comment les décisions au niveau stratégique, comme l'emplacement, ne devraient pas être intégrés à la planification tactique / opérationnel. Néanmoins, des exemples sont fournis montrant des situations où l'intégration des décisions de routage et les décisions de gestion des stocks lors de la planification de l'emplacement est bénéfique.

Premièrement, le modèle présenté est pertinent lorsque la décision de localisation des dépôts n'est pas faite pour le long terme. C'est le cas pour les entreprises qui décident stratégiquement d'avoir ses dépôts en location au lieu d'être propriétaires. L'avantage est alors la possibilité de changer de lieu périodiquement au besoin. Le modèle est également pertinent dans le cas de la logistique humanitaire. Quand une catastrophe se produit, les équipes d'intervention d'urgence sont mises en place. Des installations de distribution d'eau, de médicaments et d'autres stocks de secours sont requis [3]. Souvent, ces ressources ne sont pas censées être utilisées pour répondre aux besoins permanents. Au contraire, ils doivent être mis en place d'une façon temporaire jusqu'à ce que la situation se soit normalisée. Enfin, il est également pertinent pour le cas de la logistique militaire. Sur le champ de bataille, les bases doivent être situées afin de stocker des munitions ou pour fournir un soutien médical. Les tactiques militaires ont souvent besoin que ces bases soient fortement protégées et leurs emplacements modifiés afin de minimiser le risque d'être attaqué.

Deuxièmement, le modèle est également pertinent pour les entreprises qui ont besoin d'avoir des meilleures approximations de leurs frais de fonctionnement sur le long terme lors de la localisation des établissements. C'est le cas pour les systèmes d'approvisionnement permettant différentes fréquences de réapprovisionnement chez les détaillants et aussi quand la distribution est effectuée par des véhicules capables de visiter plus d'un détaillant par tournée. Les industries concernées par l'intégration de ces décisions sont souvent confrontées à des coûts de distribution et de possession de stock sur le long terme du même ordre que le coût fixe de localiser les dépôts. Alors, les modèles qui posent l'hypothèse de modeliser les décisions de routage limités à une seule période ne soit pas réalistes. Les coûts d'ouverture des dépôts doivent être mis à l'échelle de l'horizon modélisé pour être en équilibre avec les coûts opérationnels.

Le problème a été traité dans la littérature. En considérant la demande déterministe, Ambrosino and Scutellà [2] proposent un modèle de programmation linéaire qui combine simultanément la décision d'emplacement des dépôts, l'optimisation des tournées de véhicules et la gestion de stocks. Ils trouvent des solutions réalisables pour 12 instances à une seule période (LRP) avec 13 dépôts et 95 détaillants, montrant que les solveurs linéaires ne peuvent pas prouver l'optimalité après 25h de calcul pour la plupart des instances. Cela montre empiriquement la difficulté du problème.

En considérant la demande stochastique, la plupart des recherches abordant cette question considèrent la moyenne de la demande (modèle de Wilson) et elles proposent d'ajouter un coût relatif à une quantité économique EOQ (Economic order quantity) dans la fonction objectif pour optimiser la gestion des stocks. Les papiers présentés par Ahmadi-Javid and Azad [1], Reza Sajjadi [22], Shen and Qi [24] cherchent ainsi à résoudre un LRP non-linéaire. Leur proposition est d'inclure dans la fonction objectif le coût annuel attendu de possession de stocks pour une demande aléatoire. Cela est le terme non - linéaire du modèle. Au contraire, les papiers présentés par Liu and Lee [13], Liu and Lin [14] proposent de fixer la taille des lots pour être égale aux quantités attendues de la demande et d'optimiser les décisions de routage pour une seule période. Cela devient un problème de localisation-routage. Cette dernière approche perd la perspective globale car elle cherche à optimiser de façon séquentielle les composantes du problème.

Le modèle présenté dans ce chapitre montre une formulation permettant une décomposition de type Dantzig-Wolfe sur les variables de routage, ce qui permet de supprimer les contraintes d'élimination de sous-tours. Néanmoins, cette formulation contient encore un ensemble exponentiel de contraintes: ceux qui garantissent que les tournées des véhicules choisies peuvent livrer les quantités pour approvisionner les détaillants. Ces contraintes sont abordées avec la méthode de relaxation lagrangienne. Cela permet une décomposition en sous-problèmes.

Des recherches antérieures sur des méthodes qui intégrent la génération de colonnes et des techniques de relaxation de Lagrange montrent des résultats compétitifs pour des problèmes de tournées de véhicules (vehicle routing problem VRP) et d'autres problèmes combinatoires. En effet, il existe un lien important entre ces deux techniques. Geoffrion [8] déclare que le problème dual de Lagrange est équivalent au problème dual de la relaxation continue de la reformulation de Dantzig-Wolfe. Deux exemples de ces hybridations sont présentés par Kallehauge et al. [12] qui ont réussi à résoudre avec optimalité prouvée deux instances VRP de Homberger avec fenêtres de temps (VRPTW) avec 400 et 1000 clients, les plus grandes à être résolues à ce jour.

Aussi, Nishi et al. [18] ont présenté leur approche intégrée en utilisant la génération de colonnes et relaxation lagrangienne pour un problème d'ordonnancement de type flow shop. Le temps d'exécution est réduit avec leur technique d'environ 25%, sur des instances avec 50 tâches et 3 étapes par rapport à une méthode de génération de colonnes pure. Des problèmes de grande taille ont été résolus plus rapidement avec la version hybride de l'algorithme qu'avec celui du benchmark pur. Leurs conclusions montrent une forte sensibilité de la performance de la génération de colonnes par rapport aux multiplicateurs de Lagrange. En ce sens, beaucoup de colonnes inutiles sont générées si les multiplicateurs de Lagrange sont loin des valeurs optimales.

Cette recherche se développe sur un problème plus complexe. La formulation mathématique a deux ensembles de contraintes dépendantes avec une nature exponentielle. Un des ensembles est abordé par la méthode de relaxation lagrangienne tandis que les contraintes du VRP sont traitées avec la technique de génération de colonnes. Une procédure heuristique est proposée à partir de ces techniques afin d'avoir un meilleur contrôle sur les temps de calcul.

Dans la section 4.2, le problème a été présente avec la formulation mathématique. La section 4.3 est consacrée au principe de décomposition et la méthode heuristique. Les tests informatiques sont détaillés dans la section 4.4. L'analyse et conclusions sont montrés dans la section 4.5.

En détail, pour resoudre le problème, nous devons considérer que la formulation mathématique présentée du ILRP dans la section 4.2, a un nombre exponentiel de variables θ, celles qui modelisent le routage, et en plus, un nombre exponentiel de contraintes dans les équations (78). Ces contraintes harmonisent les décisions de routage avec celles de la gestion de stocks. Dans la pratique, cela rend le modèle très difficile à résoudre, même pour les petites instances.

L'ensemble des contraintes (78) relient les activités de distribution (routage) avec l'approvisionnement de stock dans la chaîne, et force la solution à respecter la capacité limitée du véhicule. Si ces contraintes sont relaxées, ignorant le renforcement apporté par les contraintes (81), une version relaxée du ILRP (RIRLP) est obtenue. RILRP peut être optimisé en résolvant indépendamment un SCDP et un VRP. Cela est résultat de la structure du modèle, où il n'y a pas de contrainte reliant les variables θ et w_2 autres que les équations (78).

Puis, le problème peut être décomposé en deux modèles basés sur la programmation en nombres entiers mixte. Le premier sous-problème cherchant à gérer les décisions de stocks et leur localisation est désigné comme ILP1, tandis que le sous-problème 2 (VRP2) est conçu pour prendre les décisions de routage. Les fonctions objectifs de ILP1 et VRP2 sont présentées par les équations (92) et (93).

Cette décomposition permet de séparer le routage du problème de conception de la supply chain. Pour résoudre le problème de tournées de véhicules, la méthode proposée est basée sur la génération de colonnes. Considérons l'ensemble Ω ayant toutes les tournées possibles et l'ensemble Ψ ayant toutes les combinaisons possibles de détaillants. Ainsi, le ILRP peut être restreint à ne considérer qu'un sous-ensemble des routes possibles $\Omega' \subseteq \Omega$. Compte tenu de ce sous-ensemble de permutations de détaillants Ω', le sous-ensemble correspondant de combinaisons $\Psi' \subseteq \Psi$ est calculé. Cette version restreinte sera notée comme Rv-ILRP. En outre, il serait facile de montrer en utilisant l'axiome 1, que la limitation de l'ensemble des contraintes (78) à l'ensemble Ψ' pour Rv-ILRP fournit une solution réalisable si chacun des détaillants est visité au moins par une tournée dans l'ensemble des tournées Ω'. Ensuite, il est prouvé que l'ensemble des contraintes (78) correspondant à l'ensemble $\Psi \setminus \Psi'$ est dominé par l'ensemble des contraintes (78) correspondant à l'ensemble Ψ'.

De même, en limitant le RILRP pour ne considérer que les ensembles Ψ' et Ω' au lieu de Ψ et Ω, respectivement, une version restreinte du RILRP (Rv-RILRP) est obtenue et la solution optimale est une borne inférieure du Rv-ILRP. Étant donné que $|\Psi'| \leq |\Omega'|$, le calcul des équations (92) et (93) devient plus facile.

Il est proposé de commencer par résoudre le Rv-RILRP avec des tournées élémentaires sur l'ensemble de routes Ω'. Les colonnes intéressantes (routes) vont être ajoutées dynamiquement dans Ω'. La combinaison correspondante de détaillants va être ajoutée dans Ψ' afin de maintenir le principe exposé. Cette idée est utilisée pour développer l'algorithme heuristique de "Relax-and-Price".

En conclusion, une méthode heuristique basée sur la relaxation lagrangienne pour résoudre le problème de Localisation-Routage avec contraintes de stockage (ILRP) est proposé, comprenant les idées de la génération de colonnes pour le routage des véhicules. L'objectif est d'optimiser simultanément la conception de la supply chain, la gestion des niveaux d'inventaire et les décisions de routage sans décomposer le problème en sous-problèmes afin d'optimiser globalement.

La méthodologie présentée combine une approche non traditionnelle de génération de colonnes avec une relaxation lagrangienne dans un cadre qui est désigné comme “Relax-and-Price”. Le défi est de coordonner la génération de nouvelles colonnes dans le problème de “pricing” et la mise à jour des multiplicateurs de Lagrange. Pour calculer des tournées intéressantes, un problème de plus court chemin avec profits, appelé problème du plus court chemin élémentaire généralisé, est résolu par un algorithme à deux phases.

Les résultats pour les instances générées aléatoirement se montrent competitifs par rapport à l’approche traditionnelle et des temps de calcul réduits si on les compare aux solveurs commerciaux et aux heuristiques de référence comme celle présentée dans le chapitre 2. L’impact de chaque composant de l’algorithme est évalué. Les futures recherches se concentrent sur une version multi-objective du problème et l’application des techniques de programmation stochastique afin d’inclure l’incertitude associée à la demande.

References

[1] Ahmadi-Javid, A., Azad, N., 2010. Incorporating location, routing and inventory decisions in supply chain network design. Transportation Research Part E: Logistics and Transportation Review 46 (5), 582 – 597.

[2] Ambrosino, D., Scutellà, M.-G., 2005. Distribution network design: New problems and related models. European Journal of Operational Research 165 (3), 610 – 624.

[3] Balcik, B., Beamon, B., Krejci, C., Muramatsu, K., Ramirez, M., 2010. Coordination in humanitarian relief chains: Practices, challenges and opportunities. International Journal of Production Economics 126, 22–34.

[4] Beasley, J., 1993. Lagrangean relaxation. In: C.R.Reeves (Ed.), Modern heuristic techniques for combinatorial problems. Operations Research Proceedings. Blackwell Scientific Publications, pp. 243–303.

[5] Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., Wolfler-Calvo, R., 2011. A branch-and-cut method for the capacitated location-routing problem. Computers & Operations Research 38 (6), 931 – 941.

[6] Daskin, M., Snyder, L., Berger, R., 2005. Facility location in supply chain design. In: Langevin, A., Riopel, D. (Eds.), Logistics Systems: Design and Optimization. Springer US, pp. 39–65.

[7] Feillet, D., 2010. A tutorial on column generation and branch-and-price for vehicle routing problems. 4OR: A Quarterly Journal of Operations Research 8, 407–424.

[8] Geoffrion, A., 1974. Lagrangian relaxation for integer programming. Mathematical Programming Study 2, 82–114.

[9] Guerrero, W., Prodhon, C., Velasco, N., Amaya, C., 2013. Hybrid heuristic for the inventory location-routing problem with deterministic demand. International Journal of Production Economics 146, 359–370.

[10] Guerrero, W., Velasco, N., Prodhon, C., Amaya, C., 2013. On the generalized elementary shortest path problem: A heuristic approach. Electronic Notes in Discrete Mathematics 41, 503 – 510.

[11] Hansen, P., Mladenovic, N., 2003. Variable neighborhood search. In: Glover, F., Kochenberger, G. (Eds.), Handbook of Metaheuristics. Vol. 57 of International Series in Operations Research & Management Science. Springer New York, pp. 145–184.

[12] Kallehauge, B., Larsen, J., Madsen, O., 2006. Lagrangian duality applied to the vehicle routing problem with time windows. Computers & Operations Research 33 (5), 1464–1487.

[13] Liu, S., Lee, S., 2003. A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. The International Journal of Advanced Manufacturing Technology 22, 941–950.

[14] Liu, S., Lin, C., 2005. A heuristic method for the combined location routing and inventory problem. The International Journal of Advanced Manufacturing Technology 26, 372–381.

[15] Manzini, R., 2012. A top-down approach and a decision support system for the design and management of logistic networks. Transportation Research Part E: Logistics and Transportation Review 48 (6), 1185–1204.

[16] Melo, M., Nickel, S., da Gama, F. S., 2009. Facility location and supply chain management: A review. European Journal of Operational Research 196 (2), 401–412.

[17] Mete, H., Zabinsky, Z., 2010. Stochastic optimization of medical supply location and distribution in disaster management. International Journal of Production Economics 126 (1), 76 – 84.

[18] Nishi, T., Isoya, Y., Inuiguchi, M., 2011. An integrated column generation and lagrangian relaxation for solving flowshop problems to minimize the total weighted tardiness. International Journal of Innovative Computing, Information and Control 7 (11), 6453–6471.

[19] Prins, C., Prodhon, C., Ruiz, A., Soriano, P., Wolfler-Calvo, R., 2007. Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. Transportation Science 41 (4), 470–483.

[20] Prins, C., Prodhon, C., Wolfer-Calvo, R., 2006. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR: A Quarterly Journal of Operations Research 4, 221–238.

[21] Raidl, G., Puchinger, J., 2008. Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. In: Blum, C., Aguilera, M., Roli, A., Sampels, M. (Eds.), Hybrid Metaheuristics. Vol. 114 of Studies in Computational Intelligence. Springer Berlin / Heidelberg, pp. 31–62.

[22] Reza Sajjadi, S., H. C. S., 2011. Multi-products location-routing problem integrated with inventory under stochastic demand. International Journal of Industrial and Systems Engineering 7 (4), 454–476.

[23] Salhi, S., Rand, G., 1989. The effect of ignoring routes when locating depots. European Journal of Operational Research 39 (2), 150 – 156.

[24] Shen, Z.-J., Qi, L., 2007. Incorporating inventory and routing costs in strategic location models. European Journal of Operational Research 179 (2), 372 – 389.

[25] Wagner, H., Whitin, T., 1958. Dynamic version of the economic lot size model. Management Science 5 (1), 89–96.

5. Relax-and-Price Decomposition for an Inventory Routing Problem

W.J. Guerrero[1,2], C. Prodhon[1], N. Velasco[2], C.A. Amaya[2]

[1] ICD-LOSI (UMR-STMR CNRS 6279), Université de Technologie de Troyes,
12 rue Marie Curie, CS 42060 10004 Troyes Cedex, (France)
[2] PyLO, Depto de Ingeniería Industrial, School of Engineering, Universidad de los Andes,
Cra 1 # 18A- 12, Bogotá, (Colombia)

This paper presents a decomposition method for combinatorial optimization problems having both, a large set of constraints, and a large set of variables. It is based on column generation principles and Lagrangian relaxation. Interest arises from a new formulation for the multi-vehicle Inventory-Routing problem (MIRP). First, the theoretical basis for the methodology are exposed. Second, the method is applied to find near-optimal solutions for the MIRP. Third, computational results for benchmark instances are provided. Results show the capability of the heuristic to find near-optimal solutions for this NP-hard problem.

Keywords: Decomposition method, Column generation, Lagrangian relaxation, vehicle routing, Logistics

5.1. Introduction

A decomposition principle for mixed-integer programming (MIP) problems with a special structure is presented. Consider a combinatorial optimization problem with a large number of variables and a large set of constraints. Two main optimization techniques are combined by the presented methodology: 1) Column generation, where the objective is to handle a large set of variables, and 2) Lagrangian relaxation, that approximates a problem with "difficult" constraints by a simpler problem. An algorithm capable of putting together both techniques in order to track this problem structure is presented. The studied application is the multi vehicle Inventory Routing Problem (MIRP), an extension of the vehicle routing problem (VRP) that includes inventory management decisions. Since the MIRP involves routing decisions on a multi-period planning horizon, it is NP-hard [4].

Consider a traditional MIP in the form of a minimization problem. The Dantzig-Wolfe decomposition reformulates a MIP by substituting the original variables with other representing extreme points of a substructure of the problem [14]. The resulting formulation of the problem is often called the Master-Problem (MP).

Column Generation is based on the Dantzig-Wolfe decomposition. By considering a subset of variables in the MP, a restrained version is proposed as the Restrained Master Problem (RMP). The target of limiting the number of variables in RMP, is naturally to solve it faster than the original MP. The method iteratively aims to find the excluded variables that could potentially improve the objective function if they were included. The most common method to do this is to search for variables in the LP relaxation of RMP with negative reduced cost. Solving this minimization sub-problem is called the pricing problem. When no new variables with negative reduced cost exist, the incumbent solution for MP is optimal. The reader is refered to [15, 26, 34] for more on this method.

On the other hand, Lagrangian relaxation works as follows: A set of complicating constraints is dualized into the objective function. A penalty coefficient is introduced for each violated constraint. Since some constraints in the original problem were relaxed, the resulting problem provides a lower bound if solved to optimality. The relaxed problem is supposed to be significantly easier to solve than the original problem. Iteratively, the method updates the coefficients searching to maximize the obtained lower bound. The subgradient method, the bundle method, and the volume algorithm are some of the most common procedures to perform this update [21, 22, 25].

Thus, Column generation and Lagrangian Relaxation are closely related methodologies. They are both intended to obtain tighter lower bounds on a MIP in its minimization form than the one obtained from the LP relaxation. Moreover, one could prove that it is equivalent to solve the dual of the LP relaxation of the Dantzig-Wolfe reformulation to solve the Lagrangian dual of the original problem.

Inspired by this fact, Huisman et al.[24] proposed an algorithm for a Capacitated Lot-Sizing problem, also applied for an integrated vehicle and crew scheduling problem, for a Plant Location problem, and for a Cutting Stock problem. Meanwhile, van den Akker et al.[33] study a Single-Machine common due date problem using a combination of both methods. Finally, Nishi et al.[29] present a Flowshop scheduling problem solved by an algorithm based on the integration of both methodologies as well.

The presented work stands for optimization problems with a special structure. To explain how it is different from the previous examples [24, 29, 33], consider a MIP original problem (OP) to which the corresponding Dantzig-Wolfe decomposition results in the following combinatorial optimization problem (P):

$$\text{(P): minimize } \sum_{i \in I} c_i x_i \tag{109}$$

Subject to:

$$\sum_{i \in I} a_{ij} x_i \geq b_j \quad \forall j \in J \cup \Psi \tag{110}$$

$$x_i \in \{0,1\}, \quad \forall i \in I \tag{111}$$

Where $\{x_i, \forall i \in I\}$ is a large set of decision variables representing the extreme points of (OP). Let Ψ be a large set of constraints of the reformulated problem and J a small-size set of linking constraints. By large, for example in a Dantzig-Wolfe decomposition, consider an exponential number of variables $|I|$ and Ψ is also an exponentially large set of constraints requiring significant computational resources.

Previous research by Huisman et al.[24], Nishi et al.[29], and van der Akker et al.[33] propose an optimization through the duality between OP and P since exploring both formulations simultaneously might provide better results. Nonetheless, these methods consider that P has a polynomial set of constraints. In the meanwhile, the optimization problem tackled in this paper is defined with an exponential set of constraints Ψ. Further, since extensive computational resources are required to perform parallel computing of (P) and (OP), the presented methodology is based exclusively on (P).

The structure of the paper is as follows: in section 5.2, the relax-and-price decomposition method to solve (P) in the general form to near-optimality is developed. In section 5.3, a new formulation for a version of the IRP with multiple vehicles is presented. The corresponding computational experiments of applying the presented algorithm for this MIRP formulation are detailed at section 5.4. To conclude, section presents 5.5 the discussion and final remarks of the paper.

5.2. Relax-and-Price Method

The purpose of the paper is to present a decomposition method for solving an Integer Programming (IP) model with a special structure to near-optimality. In fact, exact methods for IP models are traditionally based on a tree search scheme in which a linear relaxation of the problem is solved to optimality at each node. The classical Branch-and-Price adds dynamically new columns at each node while classical Branch-and-Cut-and-Price adds both new valid inequalities and columns at each node of the search [20]. Specific implementations of both methods might underestimate the computational burden of solving the LP relaxation at each node with an exponential number of constraints and columns.

In the presented case, at each node of the search a problem with exponential number of constraints and variables must be solved. To deal with this special problem structure, a Relax-and-Price algorithm is proposed.

First, the decomposition method for problem P will be presented. As in a pure column generation method, a pricing problem is required. A detailed description of this subproblem will be made. Next, the procedure to make updates on the Lagrangian coefficients are detailed and finally, the General purpose Relax-and-Price algorithm will be described.

5.2.1. Decomposition Method

Consider (P-LR) as a Lagrangian relaxation of (P). The modified objective function is presented by the equation (112). This function dualizes the set of constraints defined by the exponential set Ψ, while the linking constraints in the set J remain unchanged. Let μ_j be the lagrangian coeffient for each relaxed constraint $j \in \Psi$. The target of (P-LR) is to find the maximum lower bound to (P). Therefore, (P-LR) can be stated as:

$$\text{(P-LR): maximize}_{\vec{\mu}} \quad \text{minimize} \sum_{i \in I} x_i (c_i - \sum_{j \in \Psi} \mu_j a_{ij}) + \sum_{j \in \Psi} \mu_j b_j \tag{112}$$

Subject to 111 and:

$$\sum_{i \in \bar{I}} a_{ij} x_i \geq b_j \quad \forall j \in J \tag{113}$$

Let (P-CLR) be the continuous relaxation of (P-LR). It is re-defined by substituting the set of constraints (111) by the following sets of equations:

$$x_i \leq 1, \quad \forall i \in I \tag{114}$$

$$x_i \geq 0, \quad \forall i \in I \tag{115}$$

On another hand, by following the column generation principle, lets restrain the set I to a subset $\bar{I}$. That is, let the decision variables $x_i \in \bar{I}$ be explicitly considered in the problem while the set $x_i \in \bar{I} \setminus I$ be fixed to zero. By doing so, a restrained version of P-LR for fixed values of the vector $\vec{\mu}$ is stated as follows:

$$\text{(RvP-LR1): maximize}_{\vec{\mu}} \quad \text{minimize} \sum_{i \in \bar{I}} x_i(c_i - \sum_{j \in \Psi} \mu_j a_{ij}) + \sum_{j \in \Psi} \mu_j b_j \tag{116}$$

Subject to:

$$\sum_{i \in \bar{I}} a_{ij} x_i \geq b_j \quad \forall j \in J \tag{117}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \bar{I} \tag{118}$$

Therefore, the optimal solution to (RvP-LR1) for fixed values of vector $\vec{\mu}$ is an upper bound for (P-LR).

Naturally, the issue of having a large set of variables is managed by adding dynamically those variables that potentially will decrease the value of the objective function in the minimization problem described in equation 116. The procedure to identify interesting variables will be described in section 5.2.2.

Nonetheless, equation (116) still requires complete evaluation of the set of variables $\mu_j \; \forall j \in \Psi$ which is significantly large as stated before. It is proposed to restrain the set of variables $\vec{\mu}$ to the set $\bar{\Psi} \subset \Psi$ into equation (116) as follows:

$$\text{(RvP-LR2): maximize}_{\vec{\mu}} \quad \text{minimize} \sum_{i \in \bar{I}} x_i(c_i - \sum_{j \in \bar{\Psi}} \mu_j a_{ij}) + \sum_{j \in \bar{\Psi}} \mu_j b_j \tag{119}$$

Subject to equations (117) and (118).

Note once more that (RvP-LR2) is equivalent to (RvP-LR1) if the values for μ_j corresponding to constraints $j \in \Psi / \{\bar{\Psi}\}$ are fixed to zero, which corresponds to a natural initialization of the Lagrangian relaxation procedure. Thus, the set of explicitly penalized constraints $\bar{\Psi}$ can be iteratively updated by including new elements. This procedure is performed by evaluating an incumbent solution and identifying one or more violated constraints not belonging already to the set $\bar{\Psi}$. An exact separation procedure for violated constraints will be detailed for a specific problem in section 5.3.3. Also, it is proposed to select an initial set $\bar{\Psi}$ using the following rules:

1. Study the specific substructure of the problem (P) considering possible non- dominated constraints in Ψ at (RvP-LR1) given that the set of decision variables is restrained to $\bar{I}$. Then, every time a new variable x_i is added to the set $\bar{I}$, it might

become useful to add a new element into $\bar{\Psi}$ as well, by setting the corresponding μ-value larger than zero.

2. Set, by a priori understanding of the specific problem, which are the most relevant and "complicating" constraints that one would be interested in penalizing from the very beginning of the algorithm.

3. Consider that the more elements in $\bar{\Psi}$, the more updates to the corresponding μ coefficient are required. Updating a single Lagrangian coefficient is a procedure performed in complexity $O(2n)$ where $n = |\bar{I}|$. In addition, if the polyhedron formed by the initial set of constraints $\bar{\Psi}$ is far from being representative of the polyhedron formed by the set of constraints Ψ, finding a feasible solution will require more iterations.

It should be stated now that if the number of "complicating" constraints Ψ is not very large, the algorithm converges towards a classic column generation method. In that case, the method should set $\bar{\Psi} = \Psi$ very quickly. Also, If the number of variables is not significantly large, the method will rapidly converge towards a Lagrangian Relaxation procedure since the set of considered variables $\bar{I}$ will eventually be equal to the set I.

Now, in the subsequent section, the explanation on how to dynamically add a new element into the set $\bar{I}$ by solving a pricing problem is provided.

5.2.2. Pricing Problem

Based on the continuous relaxation of (RvP-LR2) for fixed values of the lagrangian multipliers $\vec{\mu}$, the corresponding dual problem can be defined. Consider $y_j \ \forall j \in J$ to be the dual variables corresponding to constraints (117). The dual program is:

$$\text{(DRP2): maximize} \sum_{j \in J} y_j \tag{120}$$

Subject to:

$$\sum_{j \in J} a_{ij} y_j \geq (c_i - \sum_{j \in \bar{\Psi}} \mu_j a_{ij}), \quad \forall i \in I \tag{121}$$

$$y_i \geq 0, \quad \forall i \in I \tag{122}$$

Assuming that the set of equations (117) defines a bounded convex polyhedron and that the primal problem has an optimal solution, (DRP2) has a feasible solution. Once (DRP2) is solved to optimality and the optimal values for the set of y_j variables are known, the pricing problem will be solved with the objective of finding a variable x_i associated with a negative reduced cost (π_i), which is computed by the following equation:

$$\pi_i = (c_i - \sum_{j \in \bar{\Psi}} \mu_j a_{ij}) - \sum_{j \in J} a_{ij} y_j \tag{123}$$

To prove optimality for the continuous relaxation of (RvP-LR2), is equivalent to show the nonexistence of any variable $i \in I$ such that $\pi_i < 0$. Otherwise, this variable could be included into the set of decision variables in (RvP-LR2) and potentially improve the objective value. In that case, the new variable is added iteratively and therefore the values of y_j are updated at each iteration.

5.2.3. Subgradient method

This section details the proposed procedure to update the vector of Lagrangian coefficients $\vec{\mu}$. It should be noted that there exists a number of different methods to perform this operation. The subgradient method, the ellipsoid algorithm, and the Bundle method are the most commonly used, but the subgradient method has the advantage of being simple and effective [25]. It consists on estimating a direction of movement for vector $\vec{\mu}$ and a step length. Each iteration, the Lagrangian coefficients are corrected using equations (124) to (126). The step length $\delta^{(p)}$ at iteration p is computed by equation (125). Each component in the correction vector $\nu^{(p)}$ at iteration p is computed by equations (126).

$$\mu_k{}^{(p)} = \max\{0, \mu_k{}^{(p-1)} + \delta^{(p)}\nu_k^{(p)}\} \ \forall k \in \Psi' \tag{124}$$

$$\delta^{(p)} = \frac{(C(S^*) - C(\hat{S}))}{\|\nu^{(p)}\|} \tag{125}$$

$$\nu_k^{(p)} = b_k - \sum_{i \in \bar{I}} a_{ij} x_i \ \forall k \in \Psi' \tag{126}$$

5.2.4. Global procedure

This section explains how to exploit the decomposition principle for an optimization problem considering a large set of constraints and a large set of variables as exposed in section 5.2.1. The challenge comes from the coordination of procedures between updating Lagrangian coefficients $\vec{\mu}$ and the inclusion of new variables into the subset of decision variables $\bar{I}$ (as explained in section 5.2.2) within the simplified model described by equations (119), (117)-(118).

The scope of the presented procedure is to compute near-optimal solutions in short times and using limited computational resources as opposed to exact methodologies where computation times grow exponentially and large computational resources are required for this type of problems. In detail, algorithm 6 presents the pseudo-code of the Relax-and-Price method using functions described in detail next.

Globally, the outer loop in steps 4 to 14 executes iterations inspired on a Lagrangian relaxation method until the stopping criteria is met. The inner loop, at steps 5 to 7, iterates by solving the incumbent relaxed model and adds columns until no further variables are interesting to be added.

The initialization procedure in step 1 is performed by selecting a subset of decision variables $\bar{I} \subset I$ and an initial set of constraints that will be explicitly penalized in the objective function in a Lagrangian fashion from the beginning of the algorithm: $\bar{\Psi} \subset \Psi$. A priori any type of variables could be included. Nonetheless, an analysis of the structure of the problem might be useful to determine a good set of initial variables.

Algorithm 6 Relax-and-Price

```
1:  Initialize Ī and Ψ̄;
2:  μ⃗ ← 0⃗;
3:  S, S* ← ∅;
4:  while (the stopping criteria is not met) do
5:      repeat
6:          Ī ← Ī∪ Solve Pricing_Problem( μ⃗, Ψ̄ );
7:      until ( π_i ≥ 0,  ∀i ∈ I )
8:      S ← Solve RvP-LR2( μ⃗, Ī, Ψ̄ );
9:      Ψ̄ ← Ψ̄ ∪ Separation_violated_constraints(S);
10:     if f(S) < f(S*) and S is feasible then
11:         S* ← S
12:     end if
13:     μ⃗ ← Update Lagrangian Multipliers;
14: end while
```

Naturally, the chosen subset $\bar{I}$ resulting from this procedure should guarantee a non-empty feasible region described by the set of inequalities (117).

Step 2 initializes to zero the values of the Lagrangian coefficients for the relaxed constraints. Step 3 initializes the solution S and the best found solution S^* to be empty.

As said before, the outer loop going from steps 4 to 14 are repeated until the stopping criteria is met. Since the algorithm is heuristic, several ways might be proposed to control the computational time. Examples of this stopping criteria might be when reaching a time limit on the execution, a maximum number of evaluated solutions, a convergence of the Lagrangian multipliers, or a fixed number of iterations without improvement, among others.

Given that the initial set of included decision variables $\bar{I}$ is a subset of I, the inner loop described from steps 5 to 7 is conceived to solve iteratively the continuous relaxation of (RvP-LR2) until no more potential improvement exist by including a decision variable $i \in I/\{\bar{I}\}$. This loop starts by solving the LP of (RvP-LR2) to optimality. Next, using the optimal values of the dual variables for (RvP-LR2), the subproblem is solved (to optimality) by the function `Pricing_Problem`($\vec{\mu}, \bar{\Psi}$). It minimizes the reduced cost of a new variable i (π_i) as defined by equation (123). If the optimal π_i is negative, the corresponding variable i is included into the set $\bar{I}$. If the optimal π_i is zero or positive, the optimal solution for the continuous relaxation of (P-LR) is found. Nonetheless, feasibility is not guaranteed.

At step 8, for fixed sets $\bar{I}, \bar{\Psi}$ and fixed values $\vec{\mu}$, the MIP solver computes the optimal solution for RvP-LR2 and stores it as S. At step 9, the algorithm evaluates for the feasibility of S by searching for any violated constraint defined by index of constraints $\Psi \setminus \bar{\Psi}$. If some violated constraints are identified, the corresponding Lagrangian coefficient is created and the constraints are included into $\bar{\Psi}$. In the event that no constraint in the set of constraints Ψ is found to be unsatisfied, S is a feasible solution for (P) and the best feasible solution found is stored at S^* (step 10-12). Step 13 updates the Lagrangian multipliers $\vec{\mu}$ using a subgradient method explained in section 5.2.3.

5.3. Inventory Routing Problem and Mathematical Model

In the vehicle routing problem (VRP), a set of geographically dispersed nodes have to be visited by a fleet of vehicles departing from a source node, such that the traveled distance is minimized. It is widely known that the VRP and its extensions are NP-Hard problems[4]. Nonetheless, based on the Dantzig-Wolfe decomposition and column generation, both exact and hybrid methodologies have being successfully proposed [5, 15, 17, 18, 28].

It is proposed to apply a Relax-and-Price approach presented on section 5.2 on a new set-covering like formulation for the multi vehicle Inventory-Routing Problem. This problem aims to coordinate routing decisions with inventory management decisions over a multi-period planning horizon (see Archetti et al.[1, 2]).

The problem considers a set of dispersed facilities V, including a 0 element representing the depot and a set V' of retailers. Decisions must be made over a discrete planning horizon H with p periods. The effect of decisions made at period p are reflected at period $p+1$, therefore the set H' is defined as $H \cup \{p+1\}$. Every period $t \in H$, a quantity r_t is made available at the depot and each facility $i \in V$ is associated to an initial inventory I_{i0}, and a holding cost per unit of product per unit of time h_i. Also, each retailer $i \in V'$ faces a non-constant demand d_{it} per period $t \in H$ and has a storage capacity of C_i. A maximum of K_{max} vehicles of capacity Q units of product is available per period.

The presented formulation is dedicated to the "Maximum-level" inventory policy IRP (ML-IRP as defined for the single-vehicle Inventory Routing problem) where retailers are replenished by a quantity that is always inferior to a limit C_i and independent to the previous inventory level. This is opposed to the "Order-up-to" level policy IRP (OU-IRP) where retailers must be replenished when visited with a quantity such that the inventory level raises to the maximum level C_i (see Solyali et al.[32]). Using these parameters, the master problem for the MIRP under the "Maximum-level" inventory policy and considering multiple vehicles is defined as follows:

5.3.1. Master Problem

To define the master problem, let the decision variables q_{it} be defined as the quantity delivered at period $t \in H$ on facility $i \in V'$. Let I_{it} be an auxiliary decision variable representing the inventory-on-hand at facility $i \in V$ at period $t \in H'$. Consider the set Ω to be defined as the set of all possible routes as sequences of retailers starting and ending at the depot, with a total traveling cost c_r. Let the parameter a_{ir} be equal to 1 iff retailer i is visited by the route r. Let the decision variable x_{rt} be equal to 1 if the route $r \in \Omega$ is used at period $t \in H$ and 0 otherwise.

Also, let Ψ be defined as the set of all possible combination of retailers, or clusters of retailers. Each cluster $k \in \Psi$ is associated to a subset of retailers $S_k \subseteq V'$. Now, the IRP is stated as follows:

$$\text{(IRP) minimize} \sum_{t \in H'} h_0 I_{0t} + \sum_{i \in V'} \sum_{t \in H'} h_i I_{it} + \sum_{r \in \Omega} \sum_{t \in H} c_r x_{rt} \tag{127}$$

Subject to:

$$I_{0t} = I_{0t-1} + r_t - \sum_{i \in V'} q_{it} \quad \forall t \in H \tag{128}$$

$$I_{0t} \geq 0 \quad \forall t \in H \tag{129}$$

$$I_{it} = I_{it-1} + q_{it} - d_{it} \quad \forall i \in V', \ \forall t \in H \tag{130}$$

$$I_{it} \geq 0 \quad \forall i \in V', \ \forall t \in H \tag{131}$$

$$I_{it} \leq C_i \quad \forall i \in V', \ \forall t \in H \tag{132}$$

$$q_{it} + I_{it-1} \leq C_i \quad \forall i \in V', \ \forall t \in H \tag{133}$$

$$\sum_{i \in S_k} q_{it} \leq Q \sum_{r \in \Omega} \beta_{rk} x_{rt} \quad k \in \Psi, \ \forall t \in H \tag{134}$$

$$\sum_{r \in \Omega} a_{ir} x_{rt} \leq 1 \quad \forall i \in V', \ \forall t \in H \tag{135}$$

$$q_{it} \geq 0 \quad \forall i \in V', \ \forall t \in H \tag{136}$$

$$\sum_{r \in \Omega} x_{rt} \leq K_{max} \quad \forall t \in H \tag{137}$$

$$x_{rt} \in \{0, 1\} \quad \forall r \in \Omega, \ \forall t \in H \tag{138}$$

Equation (127) presents the objective function including the holding costs at depots, at retailers, and the routing cost. The set of equations (128)-(133) are inventory management constraints. The set of equations (128) define the inventory levels at the depot per period. Equations (129) state that the inventory at depot is never negative. Analogously, constraints (130) define the inventory at retailers while (131) force the inventory at retailers to be non-negative. Storage capacity at retailers is guaranteed by equations (132) and (133).

Equations (134)-(137) are routing constraints. Distribution and inventory are coordinated by the set of constraints (134). They state that the total quantity delivered to cluster $k \in \Psi$ is less or equal to the vehicle capacity times the number of vehicles visiting the cluster. These constraints also guarantee that the vehicle capacities are respected. Constraints (135) force the routing construction to visit each retailer once at the most per period. Split deliveries are not allowed. The quantity delivered at each retailer is non-negative as stated by equations (136). The set of equations (137) forces to limit the number of used vehicles up to K_{max}. Finally, the constraints (138) define the x-variables to be binary.

Further, the IRP version presented by Archetti et al.[1, 2] restrained the distribution to be made by a single-vehicle. This is guaranteed by constraint (139), when $K_{max} = 1$, as:

$$\sum_{r \in \Omega} x_{rt} \leq K_{max}, \quad \forall t \in H \tag{139}$$

The later version studied by Coelho et al.[12, 9] studies the multi-vehicle version ($K_{max} > 1$) of the IRP.

Recall the decomposition method, proposed in the Relax-and-Price algorithm, proposes to consider a subset of decision variables and a subset of explicitly penalized constraints. These are denoted by the subsets $\bar{\Omega} \subseteq \Omega$ and $\bar{\Psi} \subseteq \Psi$ respectively. Thus, following the procedure described in section 5.2, the set of constraints (134) are relaxed in a Lagrangian fashion with a coefficient μ_{kt} $\forall k \in \bar{\Psi}$, $\forall t \in H$. Note that this makes the inventory problem to be independent from the routing problem since the relaxed constraints were the only ones linking $x-$variables to the other decision variables. As a matter of fact, the resulting routing problem is equivalent to solve H single-period problems since there is no constraint linking routing between periods. Therefore, the objective function for this sub-problem for a given $t \in H$ is:

$$\text{minimize} \sum_{r \in \bar{\Omega}} c_r x_{rt} - Q \sum_{k \in \bar{\Psi}} \sum_{r \in \bar{\Omega}} \mu_{kt} \beta_{rk} x_{rt} \tag{140}$$

These problems are subject to satisfy equations (135), (137) and (138). The resulting optimization problem is equivalent to (P-LR) as presented in section 5.2. Since a subset of decision variables is considered per iteration, the method evaluates for potential improvement to equation (140) by including elements into the set $\bar{\Omega}$. It is computed by solving the pricing problem, explained in detail at section 5.3.2.

5.3.2. Pricing Problem for the IRP formulation

As described in section 5.2.2, a route i in the set $\Omega/\{\bar{\Omega}\}$ potentially improves the optimal value if the reduced cost for period t, π_i^t, defined for the general case by equation (123), is negative. For the presented formulation of the IRP, consider y_{jt} to be the dual variables associated to the set of constraints (135) and dual variables y_{0t} those associated to constraints (137) in the primal formulation. The optimal solution of dual problem corresponding to (140) requires y_{jt} and y_{0t} to be non-positive. The pricing problem for a given period $t \in H$ reduces to solve:

$$\text{minimize } \pi_i^t = c_i - Q \sum_{k \in \bar{\Psi}} \mu_{kt} \beta_{rk} - \sum_{j \in V'} a_{jr} y_{jt} - y_{0t} \tag{141}$$

Let be denoted by w_{uv}, the cost associated to the traveling cost from facility u to facility v and $\hat{x}_{uv}^i$ the corresponding binary variable indicating whether the route i uses the arc (u, v) or not, $\forall\, u, v \in V'$. Now, let the cost of a route c_i be equivalent to the sum of the weights associated to the arcs composing the route ($\sum_{u \in V} \sum_{v \in V} w_{uv} \hat{x}_{uv}^i$) and the value of $a_{ui} = \sum_{v \in V} \hat{x}_{uv}^i$. Using these properties, equation (141) is equivalent to:

$$\begin{aligned}\pi_i^t &= \sum_{u\in V}\sum_{v\in V} w_{uv}\hat{x}_{uv}^i - Q\sum_{k\in\bar{\Psi}}\mu_{kt}\beta_{rk} - \sum_{u\in V} y_{ut}\sum_{v\in V}\hat{x}_{uv}^i \\ &= \sum_{u\in V}\sum_{v\in V}(w_{uv} - y_{ut})\hat{x}_{uv}^i - Q\sum_{k\in\bar{\Psi}}\mu_{kt}\beta_{rk} \qquad (142)\end{aligned}$$

This problem is introduced by Guerrero et al.[23] as the Generalized elementary shortest path problem (GESPP). The aim of the problem is to find a minimum cost path from a source node 0 (the depot) and back to it while visiting a subset of nodes in the set V'. Further, there is a predefined set of non-disjoint clusters of nodes defined by $\bar{\Psi}$. Note that there might be nodes that do not belong to any cluster. The cost of a path comprises the cost of the traversed arcs ($w_{uv} - y_{ut}$) and, for every cluster k, there is an associated profit to the cost function for visiting at least one element in the cluster k (in that case $\beta_{rk} = 1$, 0 otherwise). The magnitude of this profit is $Q\sum_{k\in\bar{\Psi}}\mu_{kt}$ for every visited cluster.

The GESPP reduces to the well-known elementary shortest path problem (ESPP) considering negative weight cycles reachable from the source node if the profits associated to visit the clusters are equal to 0. Now, the algorithms proposed by Dijkstra [16] and Bellman-Ford [7], polynomially bounded in nature, forbid negative weight cycles reachable from the source node. Therefore, they are incapable of solving the GESPP. Even more, the GESPP is NP-hard given that the ESPP with negative weight cycles is known to be NP-hard [13]. To solve this problem, the authors proposed a truncated labeling heuristic [23] inspired by the exact labeling algorithm for the ESPP with resource constraints (ESPPRC) by Feillet et al.[19]. If the best solution found by the algorithm has positive reduced cost, an integer programming solver is invoked to find the optimal path.

5.3.3. Separation Procedure

Separation procedures aim to identify violated constraints. This procedures are typically invoked in branch-and-cut or cutting plane methods for the CVRP. These routines evaluate for a given solution S if previously relaxed constraints are violated. If such a constraint exists, it is iteratively included into the pool of considered constraints.

Papers by Augerat et al.[3], Lysgaard et al.[27], Padberg et al.[30], and Ralphs[31] implement both exact and heuristic procedures to separate capacity and sub-tour elimination constraints. Heuristic procedures to identify violated constraints show competitive results within a Branch-and-Cut framework but they might not achieve their purpose. In that case, it might be required to invoke an exact procedure.

The mathematical formulation for our separation problem given a solution for delivery quantities $q_{it}^* \ \forall i \in V'$, $\forall t \in H$ and selected routes x_{rt}^*, $\forall r \in \bar{\Omega}$, $\forall t \in H$ is as follows. Naturally, the target is to find a set of retailers $S \subseteq V'$ such that $\sum_{i\in S} q_{it}^* > Q\sum_{r\in\Omega}\beta_{rk}x_{rt}^*$ given that k is the new constraint to penalize.

Then, consider parameter $b_{ik} = 1$ if retailer i is in cluster $k \in \bar{\Psi}$. Let be defined the following decision variables: $w_i = 1$ if retailer i is included into S and 0 otherwise. Also, $v_r = 1 \, \forall \, r \in \bar{\Omega}$ if the formed set S has at least one element in common with route

r. The objective function of the separation procedure, for a given period $t \in H$, is:

$$maximize \sum_{i \in V'} q^*_{it} w_i - Q \sum_{r \in \bar{\Omega}} x^*_{rt} v_r \quad (143)$$

subject to:

$$\sum_{i \in V'} b_{ik}(1 - w_i) + (1 - b_{ik}) w_i \geq 1, \ \ \forall k \in \bar{\Psi} \quad (144)$$

which is equivalent to:

$$\sum_{i \in V'} (1 - 2b_{ik}) w_i \geq 1 - |S_k|, \ \ \forall k \in \bar{\Psi} \quad (145)$$

and

$$\sum_{i \in V'} w_i a_{ir} \leq v_r, \ \ \forall r \in \bar{\Omega} \quad (146)$$

$$w_i \in \{0, 1\}, \ \forall\, i \in V' \quad (147)$$

$$v_r \in \{0, 1\}, \ \forall\, r \in \bar{\Omega} \quad (148)$$

The objective function (143) is to select the maximum amount of product delivered to a subset of retailers that exceeds the total capacity of the vehicles visiting the cluster. Constraints (144) and (145) are equivalent and both force to select at least one cluster not already included into cluster k or not to select at least one cluster included in $k \forall k \in \bar{\Psi}$. These constraints force to select a new cluster different from those in $\bar{\Psi}$. Constraints (146) force to count the number of routes visiting the new cluster through variables v_r.

5.4. Computational study

5.4.1. Benchmark Instances

Benchmark instances for the single-vehicle IRP have being proposed by Archetti et al.[1, 2, 8]. They have being used by Coelho et al. [12] to present a Branch-and-cut procedure for several classes of IRP, including a multi-vehicle version; by Coelho et al.[9] to permit transshipment between retailers; by Coelho et al.[10] to evaluate consistency in solutions; and by Coelho et al. [11] to test a branch-and-cut algorithm for a multi-product, multi-vehicle version.

These instances are randomly generated with up to three time periods and 50 customers, and up to six time periods and 30 customers. Two types of instances are analyzed in terms of holding costs (High or Low holding costs). For the multi-vehicle version of the IRP, Coelho et al.[10] proposed to divide the original vehicle capacity by the number of maximum vehicles allowed (K_{max} = 2 to 5).

Originally, these instances are solved to optimality by Bertazzi et al.[8] considering an “Order-up-to” inventory policy and a single vehicle. Nonetheless, we have relaxed these constraints and we solve the “Maximum-level” (ML) version where retailers have limited storage capacity. The optimal solutions for this version are provided by Coelho et al.[12].

5.4.2. Implementation features

The presented algorithm was coded in C using an Intel Xeon with 2.80Ghz processor and 12 GB of RAM running on Windows 7 Professional, a single thread was used for all computations and MIP models are solved with Xpress-IVE 7.0. The following features were implemented in the code:

First, the procedure to solve the GESPP presented in section 5.3.2 and called at step 7 of the pseudo-code 6 might return several (up to 6) paths with negative reduced cost. If the heuristic procedure proposed by Guerrero et al.[23] fails to find a path with negative reduced cost, the problem is solved to optimality by an IP solver.

Second, the function `Separation_violated_constraints` at step 9 in algorithm 6 solves the IP model described at section 5.3.3 and it is only invoked when the number of included clusters of retailers in $\bar{\Psi}$ is inferior to the maximum number of possible permutations Ψ_{max}. Otherwise, every possible combination of the n retailers has being included for penalization in the MIP. Ψ_{max} is precomputed as:

$$\Psi_{max} = \sum_{i=1}^{n} \left(\begin{array}{c} n \\ i \end{array} \right) = \sum_{i=1}^{n} \frac{n!}{i!(n-i)!} \tag{149}$$

Third, Lagrangian multipliers $\vec{\mu}$ are initialized to zero and updated using a subgradient method [6]. Step length λ is initialized at 0.5 and halved every $\lceil 100 * \lambda \rceil$ iterations without improvement of the lower bound or no new routes are found by solving the pricing problem. If the pricing problem finds new routes with negative reduced cost, λ is increased by adding 0.05 to λ. Setting step length is described as the "easiest thing" by Lémarechal [25], but in practice it is a very difficult since it highly affects convergence of a traditional Lagrangian relaxation procedure [6].

Fourth, if λ is not larger than $1.0e^{-3}$, the following valid inequalities are added into the model in order to accelerate convergence:

$$q_{it} \leq Q \sum_{r \in \Omega} a_{ir} x_{rt} \quad i \in V', \ \forall t \in H \tag{150}$$

Please note that these constraints are equivalent to those in equation (134) for every cluster in Ψ with a single element. These force to visit every retailer using a vehicle if the replenished quantity is positive.

Fifth, the following set of valid inequalities is included into (RvP-LR2) invoked at step 6 of the algorithm from the beginning of the procedure in order to accelerate convergence towards the optimal solution.

$$\sum_{i \in V'} q_{it} \leq Q \cdot K_{max} \quad , \ \forall t \in H \tag{151}$$

Constraints (151) force to limit the total amount delivered at period t to all retailers to be the total delivery capacity at the most. That is, the vehicle capacity times the maximum number of vehicles available.

Sixth, at every call of the `Separation_violated_constraints`(S) procedure to evaluate feasibility of the solution, one might try to repair it heuristically by making best-insertion of the unvisited retailers into the solution S. The final effect of this repairing procedure is to add the corrected routes into the pool of routes $\bar{\Omega}$. Since this correction procedure is heuristic, it might fail to find a feasible solution.

Finally, the algorithm stops when at least one of the following conditions are met:

- The current value for λ is less than $1.0e^{-4}$ and a feasible solution has been found. In this case the algorithm has reached a low convergence rate.
- More than 200 iterations of the outer loop have been executed and a feasible solution has been found. Previous tests show that more iterations require much more computational time than desired and little improvement is made.

5.4.3. Computational Results

Optimal solution values or lower bounds for the tests instances are obtained from the branch-and-cut procedure by Coelho et al.[12] coded in C++ using solver CPLEX with six threads within up to 12 hours of computation. They are available at www.leandro-coelho.com. Their computations were executed on a grid of Intel Xeon processors running at 2.66 GHz with up to 48 GB of RAM installed per node. This is clearly a superior computing capacity than ours. Our results, on the other hand, show the performance of the relax-and-price method using a single thread and single workstation. The comparison of performance is presented so the reader can make its own conclusions keeping in mind the different workstations for each method and commercial solvers used. Also, dividing our computational time by the number of computers used by Coelho et al.[12] is not proven to be an unbiased scaling factor.

Table 16 presents the comparison between both methods for instances with a single vehicle ($k = 1$), and up to five vehicles ($k = 2$ to 5), three periods, "high" and "low" inventory holding costs and up to 50 retailers. Column $\sharp$ *opt* shows the number of optimal solutions found by the branch and price method reported by Coelho et al.[12] among the 5 instances. Column *cpu* presents the average computation time in seconds.

Table 16 also presents the average computational time for the relax-and-price method, and the gap to the best known lower bound or the optimal solution when known. On average, the method computes solution that are 5.05% larger than the lower bound for instances with "high" inventory holding cost and 5.57% for instances with "low" inventory holding cost. It is to remark that the method presented by Coelho et al.[12] is highly sensitive to the number of vehicles available. Meanwhile, the computational times of the relax-and-price heuristic remain relatively stable when increasing the number of vehicles. Further, it should be noted that the quality of the lower bound is not guaranteed for large instances ($n \geq 35$) and 5 vehicles.

Results show consistent performance for instances with 6 periods. Table 17 presents the comparative results for these benchmark instances. They are made for up to 30 retailers. Once more, the problem is solved for the single vehicle case and up to five vehicles. These instances are harder to solve and the quality of the best known lower bound is not very tight for the instances with more than three vehicles. Also, the number of known optimal solutions is less than on the previous case. On average, the relax-and-price method finds solution on 341s. The average quality of the solutions found are estimated to be at 6.26% from the best lower bound for instances with "high" inventory holding cost, and 11.03% for those with "low" inventory holding cost.

Table 16: Computational results for $p = 3$ benchmark instances

Instance		High holding Cost				Low holding Cost			
		Branch-&-Cut[1]		Relax-&-Price [2]		Branch-&-Cut[1]		Relax-&-Price [2]	
		$\sharp$ opt	cpu(s)	gap(%)	cpu(s)	$\sharp$ opt	cpu(s)	gap(%)	cpu(s)
$K_{max} = 1$	n=5	5/5	0.2	0	66.0	5/5	0	0	67.2
	n=10	5/5	0.2	0	61.0	5/5	0.6	1.42	58.8
	n=15	5/5	1	0.95	62.6	5/5	1.2	1.76	56.4
	n=20	5/5	3.6	1.12	72.0	5/5	1.4	1.22	79.8
	n=25	5/5	3.8	1.37	81.0	5/5	4.6	1.64	76.2
	n=30	5/5	9	1.60	79.4	5/5	4.8	2.48	62.6
	n=35	5/5	6.6	1.51	85.0	5/5	4.8	2.54	78.2
	n=40	5/5	13.6	1.61	90.2	5/5	7.6	2.62	66.8
	n=45	5/5	16.6	1.18	112.8	5/5	10.2	3.12	92.6
	n=50	5/5	48.8	1.92	111.4	5/5	62.6	2.85	79.0
$K_{max} = 2$	n=5	5/5	3	0	70.2	5/5	4	0	74.3
	n=10	5/5	6	0.14	69.3	5/5	8	0.57	84.7
	n=15	5/5	12	1.63	74.3	5/5	12	1.85	96.1
	n=20	5/5	24	1.42	85.2	5/5	24	2.12	104.6
	n=25	5/5	31	1.34	90.1	5/5	32	1.36	127.1
	n=30	5/5	70	2.34	88.7	5/5	62	3.41	136.7
	n=35	5/5	66	1.63	94.1	5/5	56	3.21	135.4
	n=40	5/5	479	1.74	116.4	5/5	525	1.45	147.3
	n=45	5/5	1595	2.36	121.4	5/5	3868	2.74	156.2
	n=50	5/5	4432	1.62	131.8	4/5	10797	8.31	163.8
$K_{max} = 3$	n=5	5/5	3	0	83.4	5/5	5	0	76.2
	n=10	5/5	13	1.31	85.1	5/5	17	1.68	94.3
	n=15	5/5	26	0.98	88.9	5/5	31	1.64	104.7
	n=20	5/5	217	0.24	95.2	5/5	221	1.13	107.5
	n=25	5/5	1014	1.64	113.4	5/5	574	1.55	132.4
	n=30	5/5	1623	1.42	115.2	5/5	1286	1.04	124.1
	n=35	5/5	2696	1.93	121.0	5/5	1936	1.12	146.9
	n=40	5/5	6312	1.63	142.1	5/5	9092	1.75	152.4
	n=45	2/5	32821	18.14	150.7	2/5	31805	4.07	167.2
	n=50	0/5	42991	24.34	144.2	0/5	42930	14.25	157.6
$K_{max} = 4$	n=5	5/5	3	0	75.2	5/5	4	0.98	91.6
	n=10	5/5	13	0.99	79.1	5/5	41	1.04	99.3
	n=15	5/5	26	1.42	84.3	5/5	119	1.54	106.5
	n=20	5/5	217	1.95	86.1	5/5	5544	1.41	118.8
	n=25	5/5	1014	2.33	90.1	5/5	4666	3.58	119.2
	n=30	2/5	1623	2.41	87.4	2/5	29715	7.04	119.3
	n=35	2/5	2696	1.94	96.7	2/5	31756	10.15	125.1
	n=40	2/5	6312	1.44	115.4	0/5	43010	18.91	140.9
	n=45	1/5	32821	16.84	106.2	1/5	34722	14.12	149.2
	n=50	0/5	42991	51.42	120.4	0/5	42999	20.36	160.9
$K_{max} = 5$	n=5	5/5	3	0	92.3	5/5	5	0	102.4
	n=10	5/5	13	0.12	94.7	5/5	94	0.72	104.4
	n=15	5/5	26	0.85	110.4	5/5	1196	1.41	115.1
	n=20	4/5	217	2.45	145.2	3/5	14619	5.64	129.7
	n=25	2/5	1014	6.41	132.4	2/5	26720	4.91	130.1
	n=30	1/5	1623	7.10	153.6	1/5	39794	14.60	134.4
	n=35	0/5	2696	12.74	157.9	0/5	43010	13.77	134.8
	n=40	0/5	6312	10.16	163.4	0/5	34767	26.41	152.7
	n=45	0/5	32821	13.65	162.3	0/5	43046	17.46	185.0
	n=50	0/5	42991	15.25	175.4	0/5	43046	41.66	203.4

[1] Grid of Intel Xeon 2.66 GHz processors with up to 48 GB of RAM installed per node [12]
[2] Intel Xeon with 2.8Ghz processor and 12 GB of RAM

Table 17: Computational results for $p = 6$ benchmark instances

	Instance	High holding Cost				Low holding Cost			
		Branch-&-Cut[1]		Relax-&-Price [2]		Branch-&-Cut[1]		Relax-&-Price [2]	
		♯ opt	cpu(s)	gap(%)	cpu(s)	♯ opt	cpu(s)	gap(%)	cpu(s)
$K_{max} = 1$	n=5	5/5	1.6	0	86.0	5/5	3.2	0.69	88.0
	n=10	5/5	6.4	1.64	94.8	5/5	7.8	1.01	96.4
	n=15	5/5	22.4	1.31	101.2	5/5	22.4	1.72	99.5
	n=20	5/5	28.6	0.04	101.3	5/5	40.6	1.14	105.3
	n=25	5/5	43.2	1.09	104.4	5/5	54.0	1.55	110.6
	n=30	5/5	70.0	1.47	118.4	5/5	96.8	1.44	129.9
$K_{max} = 2$	n=5	5/5	9.0	1.54	172.1	5/5	56.6	0	74.3
	n=10	5/5	57.6	2.70	187.8	5/5	14578.6	0.57	84.7
	n=15	5/5	351.0	1.88	198.4	5/5	18761.4	1.85	96.1
	n=20	5/5	4035.8	1.87	212.8	5/5	42751.8	2.12	104.6
	n=25	5/5	10160.2	3.61	264.4	4/5	43047.4	1.36	127.1
	n=30	3/5	28788.8	4.80	298.9	1/5	43079.6	3.41	136.7
$K_{max} = 3$	n=5	5/5	38.0	0	205.5	5/5	56.6	1.78	205.5
	n=10	4/5	14611.2	1.73	281.2	4/5	14578.6	2.40	262.4
	n=15	4/5	12470.4	2.66	321.4	4/5	18761.4	4.35	311.9
	n=20	0/5	42985.6	9.58	361.4	1/5	42751.8	16.81	363.3
	n=25	1/5	39241.4	8.23	400.6	0/5	43047.4	14.90	396.7
	n=30	0/5	42963.8	9.47	407.7	0/5	43079.6	20.14	420.3
$K_{max} = 4$	n=5	5/5	51.2	0	391.9	5/5	77.6	0.63	356.6
	n=10	2/5	25466.6	5.99	462.2	2/5	30232.4	7.07	500.9
	n=15	0/5	39944.6	6.42	545.6	0/5	42292.8	8.86	522.2
	n=20	0/5	37441.0	11.95	547.7	0/5	43098.2	23.49	537.5
	n=25	0/5	34401.6	16.33	572.9	0/5	42537.2	26.26	542.6
	n=30	0/5	43080.0	15.41	591.3	0/5	42684.6	36.91	610.0
$K_{max} = 5$	n=5	5/5	131.0	2.00	474.9	4/4	207.8	1.74	477.5
	n=10	0/5	37589.2	8.54	505.5	0/5	36498.4	14.61	512.2
	n=15	0/5	42801.2	10.32	532.1	0/5	37435.2	18.44	535.1
	n=20	0/5	42965.0	18.41	612.8	0/5	42445.8	30.36	660.7
	n=25	0/5	41063.8	18.84	681.8	0/5	42281.8	41.03	689.8
	n=30	0/5	43042.8	20.05	703.3	0/5	30445.4	44.48	722.4

[1] Grid of Intel Xeon 2.66 GHz processors with up to 48 GB of RAM installed per node [12]
[2] Intel Xeon with 2.8Ghz processor and 12 GB of RAM

The exposed results are outperformed by the heuristic presented in chapter 3. It presented a Multi-start Iterated local search for the multi-depot version of the IRP. The exposed ideas in chapter 3 are simple and easy to implement. Further, the computational results showed the efficiency of the method and the capability of improving the benchmark.

Nevertheless, potential room for improvement of the relax-and-price method is acknowledged. Two directions are possible:

- First, to make appropriate tuning of the algorithm (the method to update Lagrangian multipliers, the dynamic update of the step size, the number of routes generated by the heuristic used to solve the GESPP, how often should the separation procedure be invoked, the initialization of the pool of routes, among others).
- Second, to use more sophisticated branching methods are used to solve simpler problems by fixing variable values. That is, to embed the algorithm within a search tree inspired on Branch-and-cut-and price algorithms.

Finally, it is important to state that long computational times were obtained by the presented method since an exact separation algorithm is used instead of trying to improve the solution by a local search operator. Nonetheless, our expectations are set on better results once the code of the algorithm is fully optimized. As a matter of fact, chapter 4 provides interesting results for the Inventory Location-Routing problem (ILRP). Faster than those computed by the cooperative heuristic exposed in chapter 4. Therefore, a similar behavior is expected for the MIRP.

5.5. Conclusions

A new relax-and-price algorithm is presented. The procedure is designed for problems with exponential number of variables and simultaneously exponential number of constraints. Even if the set of constraints might be large, it is assumed that feasibility of the solution might be checked in polynomial time. A computational study proved that the method, by alternating between a column generation procedure and a Lagrangian relaxation technique, provides a near-optimal solution for the problem.

The first contribution of the paper is to present a new mathematical formulation for the Inventory-Routing problem. It is based on a Dantzig-Wolf formulation of the routing variables. From this point, it would be easy to extend the inventory-routing problem towards considering time-windows and/or split deliveries because column generation techniques and branch-and-price frameworks have shown good results for VRPs with these constraints.

The second contribution is the implementation of the relax-and-price algorithm on this problem based on the new mathematical formulation and to provide interesting results on a large set of benchmark instances even though limited using computation resources were used. This is current research and robust design of the algorithm is still in progress.

Acknowledgements

This research is supported by the Champagne-Ardenne Regional Council (France), Centro de Estudios Interdisciplinarios Básicos y Aplicados - CEIBA (Colombia) and ICD-LOSI, Université de Technologie de Troyes.

5.6. Résumé en français

Un principe de décomposition pour des problèmes de programmation en nombres entiers (MIP) avec une structure spéciale est étudié. Considérons un problème d'optimisation combinatoire qui compte un grand nombre de variables et un grand nombre de contraintes. Deux principales techniques d'optimisation sont ici combinées: 1) La génération de colonnes, où l'objectif est de gérer un grand nombre de variables, et 2)la relaxation de Lagrange, qui cherche à resoudre un problème avec des contraintes "difficiles" par un problème plus simple. Un algorithme capable de combiner ces deux techniques pour résoudre un problème avec ces caractéristiques est présenté. L'application étudiée est une extension du problème de tournées de véhicules (vehicle routing problem, VRP) qui intègre les décisions de gestion des stocks (Inventory-Routing Problem IRP). L'IRP intègre des décisions de routage sur un horizon de planification multi-période, il est NP-difficile [4].

Considérons un MIP traditionnel sous la forme d'un problème de minimisation. La décomposition de Dantzig-Wolfe reformule le MIP en substituant les variables d'origine par des variables représentant les points extrêmes du problème [14]. Cette formulation du problème est souvent appelée le problème maître (Master Problem MP).

La génération de colonnes est basée sur la décomposition de Dantzig-Wolfe. En prenant en compte un sous-ensemble de variables dans le MP, une version restreinte est proposée (Restrained Master Problem RMP). Limiter le nombre de variables dans RMP mène naturellement à le résoudre plus rapidement que le problème original. La méthode vise de manière itérative à trouver les variables exclues qui pourraient améliorer la fonction-objectif si elles étaient incluses. La téchnique la plus courante pour le faire est de chercher les variables dans la version relaxée LP du RMP avec un coût réduit négatif. La résolution de ce sous-problème de minimisation est appelé le problème de "pricing". Quand il n'existe pas de nouvelles variables avec un coût réduit négatif, la solution LP du MP est optimale. Le lecteur est renvoyé aux papiers de [15, 26, 34] pour en savoir plus sur cette méthode.

D'autre part, la relaxation lagrangienne fonctionne comme suit: Un ensemble de contraintes (qui souvent compliquent le problème) est relaxé et pénalisé dans la fonction-objectif. Un coefficient de pénalité est introduit pour chaque contrainte violée. Comme quelques contraintes dans le problème original ont été relâchées, le problème qui en résulte fournit une borne inférieure s'il est résolu à optimalité. Cette version du problème est censé être beaucoup plus facile à résoudre que le problème initial. Itérativement, la méthode met à jour les coefficients lagrangiens afin de maximiser la borne inférieure obtenue. La méthode de sous-gradient, la méthode du "bundle", et l'algorithme de volume sont quelques-unes des procédures les plus courantes pour effectuer cette mise à jour [21, 22, 25].

Ainsi, la génération de colonnes et la relaxation lagrangienne sont des méthodologies fortement liées. Elles sont toutes deux destinées à obtenir des bonnes bornes inférieures pour un MIP dans sa forme de minimisation, meilleures que celles obtenues à partir de la relaxation linéaire du problème. En outre, on pourrait prouver qu'il est équivalent de résoudre le problème dual de la relaxation linéaire de la reformulation de Dantzig-Wolfe et de résoudre le problème dual de Lagrange du problème initial.

Inspiré par ce constat, Huisman et al.[24] ont proposé un algorithme pour résoudre un problème de lotissement avec capacités, mais aussi pour un problème de tournées intégré avec le problème d'ordonnancement d'équipage, pour un problème de localisation d'usines, et pour un problème de découpes. Ainsi, van den Akker et al.[33] étudient un problème d'ordonnencement avec date limite commune sur une seule machine en utilisant une combinaison des deux méthodes. Enfin, Nishi et al.[29] posent un problème d'ordonnancement de flowshop résolu par un algorithme basé sur l'intégration des deux méthodologies également.

L'approche présentée travaille sur des problèmes d'optimisation avec une structure spéciale. Pour expliquer en quoi la méthode présentée est différente des exemples précédents [24, 29, 33], considérons un problème MIP noté (OP). Le résultat de la décomposition de Dantzig-Wolfe correspondante est le problème d'optimisation combinatoire (P) suivant:

$$\text{(P): minimiser } \sum_{i \in I} c_i x_i \tag{152}$$

Sous les contraintes:

$$\sum_{i \in I} a_{ij} x_i \geq b_j \quad \forall j \in J \cup \Psi \tag{153}$$

$$x_i \in \{0, 1\}, \quad \forall i \in I \tag{154}$$

Soit $\{x_i,\ \forall i \in I\}$ un grand ensemble de variables de décision représentant les points extrêmes de (OP). Soit Ψ un grand ensemble de contraintes du problème reformulé et J un ensemble de petite taille des contraintes de liaison. Dans une décomposition de Dantzig-Wolfe, nous envisageons un nombre exponentiel de variables I et Ψ ainsi qu'un ensemble exponentiel de contraintes nécessitant des ressources de calcul importantes.

Des recherches antérieures par [24, 29, 33] proposent de travailler en même temps avec une dualité entre (OP) et (P) pour optimiser l'exploration car le fait d'utiliser les deux formulations pourraient donner de meilleurs résultats. Néanmoins, ces méthodes considèrent que (P) a un petit ensemble de contraintes. Cependant, le problème d'optimisation abordé dans le présent document est défini par un ensemble de contraintes exponentiel Ψ. En outre, étant données les nombreuses ressources informatiques nécessaires pour effectuer le calcul parallèle de (P) et (OP), l'approche présentée est basée exclusivement sur (P).

Dans la section 5.2, la méthode de décomposition relax-and-price pour résoudre (P) sous la forme générale à la quasi-optimalité est développée. Dans la section 5.3, une nouvelle formulation pour une version de l'IRP avec plusieurs véhicules est présentée. Les expériences de calcul correspondantes de l'application de l'algorithme présenté pour cette formulation IRP sont précisées à la section 5.4. Pour conclure, l'article présente dans la section 5.5 la discussion finale et les remarques du papier.

En conclusion, un nouvel algorithme de relax-and-price a été présenté. La procédure est conçue pour des problèmes avec un nombre exponentiel de variables et avec un nombre exponentiel de contraintes simultanément. Même si l'ensemble des contraintes pourrait être grande, il est supposé que la faisabilité de la solution peut être vérifiée en temps polynomial. Les tests informatiques ont prouvé que la méthode, par l'alternance entre une procédure de génération de colonnes et une technique de relaxation lagrangienne, fournit une solution proche de l'optimalité pour le problème.

La première contribution de ce papier est de présenter une nouvelle formulation mathématique du problème de tournées avec gestion de stocks appropiée à une décomposition de Dantzig-Wolfe sur les variables de routage. De ce point, il serait facile d'étendre la formulation afin de considérer des livraisons fractionnées et/ou des fenêtres horaires pour la livraison puisque les techniques de génération de colonnes et des cadres de branch-and-price ont montré de bons résultats pour les problèmes de tournées avec ces contraintes.

La deuxième contribution de ce papier est la mise en œuvre de l'algorithme de relax-and-price sur ce problème sur la base de la nouvelle formulation mathématique. Les expériences montrent des résultats intéressants sur un grand nombre d'instances de référence, même si les ressources informatiques utilisées sont limitées. Cette recherche est en cours et des améliorations concernant une conception robuste de l'algorithme sont envisagées.

References

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

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

[3] Augerat, P., Belenguer, J., Benavent, E., Corberán, A., Naddef, D., 1998. Separating capacity constraints in the CVRP using tabu search. European Journal of Operational Research 106, 546 – 557.

[4] Baita, F., Ukovich, W., Pesenti, R., Favaretto, D., 1998. Dynamic routing-and-inventory problems: a review. Transportation Research Part A: Policy and Practice 32 (8), 585–598.

[5] Baldacci, R., Toth, P., Vigo, D., 2007. Recent advances in vehicle routing exact algorithms. 4OR 5, 269–298.

[6] Beasley, J., 1993. Lagrangean relaxation. In: Reeves, C. (Ed.), Modern heuristic techniques for combinatorial problems. Operations Research Proceedings. Blackwell Scientific Publications, pp. 243–303.

[7] Bellman, R., 1958. On a routing problem. Quarterly of Applied Mathematics 16 (1), 89–90.

[8] Bertazzi, L., Paletta, G., Speranza, M., 2002. Deterministic order-up-to level policies in an inventory routing problem. Transportation Science 36 (1), 119–132.

[9] Coelho, L., Cordeau, J., Laporte, G., 2012. The inventory-routing problem with transshipment. Computers & Operations Research 39 (11), 2537–2548.

[10] Coelho, L., Cordeau, J.-F., Laporte, G., 2012. Consistency in multi-vehicle inventory-routing. Transportation Research Part C: Emerging Technologies 24 (0), 270 – 287.

[11] Coelho, L., Laporte, G., 0. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. International Journal of Production Research 0 (0), 1–14.

[12] Coelho, L., Laporte, G., 2013. The exact solution of several classes of inventory-routing problems. Computers & Operations Research 40 (2), 558–565.

[13] Cormen, T., Leiserson, C., Rivest, R., Stein, C., 2001. Introduction to algorithms, 2nd Edition. MIT Press, Cambridge, MA, USA.

[14] Dantzig, G., Wolfe, P., 1960. Decomposition principle for linear programs. Operations Research 8 (1), 101–111.

[15] Desaulniers, G., Desrosiers, J., Solomon, M., 2005. Column Generation. Springer US.

[16] Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. NUMERISCHE MATHEMATIK 1 (1), 269–271.

[17] Dror, M., Langevin, A., 2000. Transformations and Exact Node Routing Solutions by Column Generation. Cahiers du GÉRAD. GERAD, École des hautes études commerciales, Groupe d'études et de recherche en analyse des décisions (Montréal, Québec).

[18] Feillet, D., 2010. A tutorial on column generation and branch-and-price for vehicle routing problems. 4OR 8, 407–424.

[19] Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks 44 (3), 216–229.

[20] Feillet, D., Gendreau, M., Medaglia, A., Walteros, J., 2010. A note on branch-and-cut-and-price. Operations Research Letters 38 (5), 346–353.

[21] Fisher, M., 2004. The lagrangian relaxation method for solving integer programming problems. Management Science 50 (12 Supplement), 1861–1871.

[22] Geoffrion, A., 2010. Lagrangian relaxation for integer programming. In: Jünger, M., Liebling, T., Naddef, D., Nemhauser, G., Pulleyblank, W., Reinelt, G., Rinaldi, G., Wolsey, L. (Eds.), 50 Years of Integer Programming 1958-2008. Springer Berlin Heidelberg, pp. 243–281.

[23] Guerrero, W., Velasco, N., Prodhon, C., Amaya, C., 2013. On the generalized elementary shortest path problem: A heuristic approach. Electronic Notes in Discrete Mathematics 41 (1), 503–510.

[24] Huisman, D., Jans, R., Peeters, M., Wagelmans, A., 2005. Combining column generation and lagrangian relaxation. In: Desaulniers, G., Desrosiers, J., Solomon, M. (Eds.), Column Generation. Springer US, pp. 247–270.

[25] Lemaréchal, C., 2001. Lagrangian relaxation. In: Jünger, M., Naddef, D. (Eds.), Computational Combinatorial Optimization. Vol. 2241 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 112–156.

[26] Lübbecke, M., Desrosiers, J., 2005. Selected topics in column generation. Operations Research 53 (6), 1007–1023.

[27] Lysgaard, J., Letchford, A., Eglese, R., 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming 100 (2), 423–445.

[28] Mbaraga, P., Langevin, A., Laporte, G., 1999. Two exact algorithms for the vehicle routing problem on trees. Naval Research Logistics (NRL) 46 (1), 75–89.

[29] Nishi, T., Isoya, Y., Inuiguchi, M., 2011. An integrated column generation and lagrangian relaxation for solving flowshop problems to minimize the total weighted tardiness. International Journal of Innovative Computing, Information and Control 7 (11), 6453–6471.

[30] Padberg, M., Rinaldi, G., 1991. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Review 33 (1), 60–100.

[31] Ralphs, T., 2003. Parallel branch and cut for capacitated vehicle routing. Parallel Computing 29 (5), 607 – 629, parallel computing in logistics.

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

[33] van den Akker, M., Hoogeveen, H., de Velde, S., 2002. Combining column generation and lagrangean relaxation to solve a single-machine common due date problem. INFORMS Journal on Computing 14 (1), 37–51.

[34] Wilhelm, W., 2001. A technical review of column generation in integer programming. Optimization and Engineering 2, 159–200.

6. Branch-and-Cut and heuristic procedure for the Generalized Elementary Shortest Path Problem

W.J. Guerrero[1,2], N. Velasco[2,3], C. Prodhon[1], C.A. Amaya[2]

[1] ICD-LOSI (UMR-STMR CNRS 6279), Université de Technologie de Troyes,
12 rue Marie Curie, CS 42060 10004 Troyes Cedex, (France)
[2] PyLO, School of Engineering, Universidad de los Andes,
Cra 1 # 18A- 12, Bogotá, (Colombia)
[3] Invited Professor, Université de Technologie de Troyes,
12 rue Marie Curie, CS 42060 10004 Troyes Cedex, (France)

The Generalized Elementary Shortest Path Problem (GESPP) is studied in this paper. Consider a graph where nodes belong to predefined non-disjoint clusters. Each cluster is associated to a fixed profit. The problem is to compute the minimum cost path from a given source node to the sink node in a graph. The cost value of a path is given by the sum of the cost of the chosen arcs and subtracting the profits collected from visiting the corresponding clusters. Application contexts include school bus routing, pricing problems, public transportation or telecommunication network design. Depending on the case, clusters could be interpreted as groups of nodes with linking features as, for example, being easily reachable from each other, or some kind of coverage guarantee. The GESPP is compared to similar problems in the literature. A new mathematical formulation is proposed and two solution methods are compared. A Branch-and-cut procedure and a two-phase heuristic algorithm for graphs including negative cycles are presented and compared. Tests on benchmark random instances with up to 100 nodes are performed to evaluate the capability of the algorithms.

Keywords: Routing problems, Branch-and-Cut procedure, Elementary shortest path problem, Heuristics.

6.1. Introduction

A variant of the elementary shortest path problem (ESPP) is studied by considering arbitrary arc costs and profits associated to visit predefined clusters of nodes at least once. It is the generalized elementary shortest path problem (GESPP). As a matter of fact, it reduces to the ESPP with negative weight cycles reachable from the source node, if the profits associated to visit the clusters are equal to 0. Recall that the polynomially bounded algorithms by Dijkstra [8] and Bellman-Ford [3] forbid negative weight cycles reachable from the source node. Then, the GESPP is NP-hard given that the ESPP with negative weight cycles is known to be NP-hard [4]. This research presents a stronger formulation than previous in the literature. A Branch-and-cut method to solve the problem to optimality is proposed and a heuristic algorithm is proposed inspired by the exact labeling algorithm for the ESPP with resource constraints (ESPPRC) from Feillet et al.[10]. A recent survey on resource constrained shortest path problems with emphasis on exact methods is presented by Di Puglia Pugliese et al. [7].

An example to illustrate the GESPP is now provided. Consider a network in which a set of non-disjunctive clusters of nodes, denoted as the set Ψ, is predefined. No constraints on the size of the clusters are imposed (it might even contain 1 node or all the nodes) and there might be nodes not belonging to any cluster. The objective of the optimization problem is to find the minimum cost path from a defined source node to the sink node. The cost of the path P is computed by aggregating the costs c_{ij} of each traversed arc (i, j) in the path and the aggregated profits p_t for each visited cluster $C_t, \forall t \in \Psi$. To collect a particular profit from a cluster, at least one node within the cluster must be in the path P. Figure 12 presents the graph for an example of the GESPP considering $n = 9$. Nodes 0 and 10 are the source and sink respectively. Consider four clusters: $C_1 = \{1, 2, 4, 5\}$, $C_2 = \{2, 3\}$, $C_3 = \{7, 8\}$, and $C_4 = \{5, 6, 8, 9\}$, associated to the profits p_1, p_2, p_3 and p_4 respectively. The path $\{0 - 2 - 4 - 7 - 10\}$ would have a cost equal to $c_{0,2} + c_{2,4} + c_{4,7} + c_{7,10} - p_1 - p_2 - p_3$.

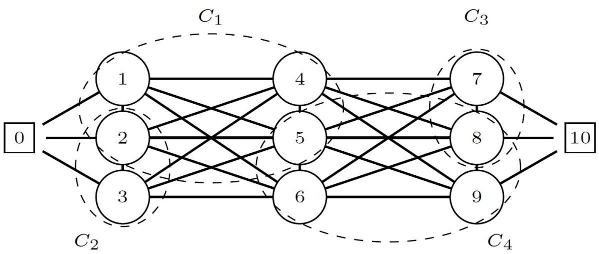

Figure 12: Graph for the GESPP - an example

Similar problems in the literature have been studied. For example, Current et al.[5] introduced the median shortest path problem (MSPP). The MSPP is a bi-objective problem trading off the distance of the path with the accessibility of the path. The distance of the path is the sum of the travel cost of each chosen arc. Accessibility is measured as the total weighted travel distance that demand must traverse to reach the nearest node on the selected path. This last objective matches with the p-median facility location problem. The solution approach proposed by the author is based on the enumeration of k-shortest paths, and it is limited to graphs with non-negative cycles. Further, Nepal and Park[17] propose to restrain the search by limiting the paths with excessive path length or excessive accessibility cost, in order to make a faster computation of the Pareto frontier.

The median cycle problem (MCP), studied by Labbé et al.[14], aims to design a tour to visit a subset of nodes in the graph while satisfying that the sum of the distance from every unvisited node to the closest node in the tour is not greater than a predefined bound. It also presents computational results for a branch-and-bound algorithm and two heuristic procedures.

The GESPP and the traveling salesman problem (TSP) with profits [9] have both in common that a subset of nodes is visited and a notion of profit is proposed. While the TSP with profits associates a benefit to each node, the GESPP associates a profit per cluster of nodes. Still, in the TSP with profits, the model objective is to minimize the distance of a tour that visits a subset of nodes while maximizing or satisfying a minimum collected profit from each visited node. More on the TSP with profits and orienteering problems can be found at Baldacci et al. [2] and Tricoire et al. [18].

Also, Ahmed et al. [1] presents a shortest possibly non-simple path problem. This particular problem considers a set of subpaths to be forbidden. Unlike the proposed algorithms by Villeneuve and Desaulniers [19], for their application on optical networks, information about the forbidden sequences of arcs is not known a priori, only when a path fails to connect the source with the destination node.

Finally, considering a set of predefined clusters of nodes, Festa et al.[11] consider the problem of computing a minimum cost path from a given origin node to a given destination node while visiting at least one node within every cluster. Further, clusters must be visited in a fixed order. It is denoted as the shortest path tour problem (SPTP). For the SPTP, clusters are forced to be visited at least once and the path does not have to be elementary. Table 18 summarizes the main features of related shortest path problems.

Table 18: Summary of main features for related problems

Problem	multi objective	Clusters of nodes	Profits	Negative cycles	Resource constraints	Solution approach
GESPP		✓	✓	✓		exact
ESPPRC[10]				✓	✓	exact
MSPP[5]	✓					exact
MCP[14]						exact
TSP with profits [9]	✓		✓	✓		survey
OP [18]			✓			heuristic
SP with forbidden subpaths [1, 6, 19]						exact
SPTP [11]		✓				exact

Application contexts for the GESPP include: 1) Urban transportation network design, to optimize the design of a new bus or metro line; 2) New rail lines design to connect two mayor cities while deserving smaller villages; and 3) Telecommunications network design with profits for increased reliability when interconnecting hubs.

In the following: section 6.2 presents two GESPP mathematical formulations. The proposed solution methods are presented in section 6.3. Subsection 6.3.1 details the branch-and-cut procedure and subsection 6.3.2 explains the heuristic algorithm, denoted as $\mathcal{H}$, and a performance evaluation is studied in section 6.4. Conclusions are exposed in section 6.5.

6.2. Mathematical Formulations

Let the GESPP be defined over a complete, weighted and undirected graph G composed by a set J of n nodes, a source node $\{0\}$ and a sink node $\{n+1\}$. In the sequel the set of nodes V is defined as $J \cup \{0\} \cup \{n+1\}$. Each arc in $A = \{(i, j), \forall i, j \in V$ in G is associated to a cost $c_{ij} \in \mathbb{R}$ (G may contain negative cycles). Also, nodes are aggregated in predefined non-disjoint clusters. Each cluster $t \in \Psi$ is associated with a profit $p_t \geq 0$ to the cost function if at least one node in t is visited. Depending on the application, clusters could be interpreted as groups of nodes with linking features, easily reachable from each other, or some kind of coverage guarantee. Let x_{ij} be a binary decision variable indicating if the arc $(i, j|i < j)$ belongs to the path. Let y_t be a binary variable equal to 1 iff cluster $t \in \Psi$ is visited at least once.

The mathematical formulation, as presented by Guerrero et al. [12], is:

$$\text{GESPP:} \quad min \sum_{i \in J \cup \{0\}} \sum_{j \in J \cup \{n+1\}} c_{ij} x_{ij} - \sum_{k \in \Psi} y_k p_k \tag{155}$$

Subject to:

$$\sum_{i \in J} x_{0,i} = 1 \tag{156}$$

$$\sum_{i \in J} x_{i,n+1} = 1 \tag{157}$$

$$\sum_{i \in J \cup \{0\}} x_{ij} - \sum_{i \in J \cup \{n+1\}} x_{ji} = 0 \,, \forall j \in J \tag{158}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \,, \forall \, S \subseteq J \tag{159}$$

$$\sum_{i \in S_k} \sum_{j \in J \setminus \{S_k\}} x_{ij} \geq y_k \;\; , \forall \, k \in \Psi \tag{160}$$

$$x_{ij} \in \{0,1\} \;\; \forall \, i \in J \cup \{0\}, \forall \, j \in J \cup \{n+1\}, y_k \in \{0,1\} \;\; \forall \, k \in \Psi \tag{161}$$

The objective of the problem is defined by equation (155) aiming to minimize the result of the path length cost after subtracting the cluster profits. Constraints (156), (157), and (158) are traditional to a shortest path problem. The path must start at the source node and end at the sink node as imposed by equations (156) and (157). Constraints (158) are flow conservation constraints. Subtour elimination constraints (159) are required given the potential negative cycles in G. A cluster profit is obtained iff the path visits any node belonging to the cluster as stated by constrains (160). Decision variables are binary as defined by equations (161).

In addition, this paper introduces a binary decision variable z_j equal to 1 iff node $j \in J$ belongs to the chosen path. Then, a new and stronger formulation for the GESPP is composed by equations (155)-(157), (161), and the following constraints:

$$\sum_{i \in V, i<j} x_{ij} + \sum_{i \in V, j<i} x_{ji} = 2 \cdot z_j \,, \forall j \in J \tag{162}$$

$$\sum_{i \in S} \sum_{j \in S, i<j} x_{ij} \leq \sum_{j \in S} z_j - z_k \,, \forall \, S \subseteq J, \text{ for some } k \in J \tag{163}$$

$$\sum_{i \in S_t} z_i \geq y_t \;\; , \forall \, t \in \Psi \tag{164}$$

$$z_i \in \{0,1\} \;\; \forall \, i \in J \tag{165}$$

Constraints (162) are flow conservation constraints while subtour are forbidden by constraints (163). Cluster profit are collected by including constraints (164). Finally, decision variables z are binary as defined by equations (165).

6.3. Solution Procedures

In the following sections, two solution approaches are proposed to solve the GESPP to optimality and near-optimality. First, a Branch-and-Cut algorithm is presented. Second, a two-phase heuristic procedure is developed.

6.3.1. Branch-and-Cut

Using the strong formulation presented in section 6.2, a Branch-and-Cut procedure was implemented. Since the set of constraints (163) is exponential, they are dynamically added at each node of the search. A heuristic separation procedure is used to identify at least one subtour at each node of the search. If more than a single cut is found, all of them are added.

The reader is referred to Lysgaard et al.[15] for more literature on exact and heuristic separation procedures for Branch-and-Cut procedures in routing problems.

Further, our branching strategy is to select the variable which has a solution value for the LP resolution closest to 0.5. This is a standard practice in Branch-and-Cut implementations. The target is to select a variable $\chi \in \{x, y, z\}$ and to impose a disjunction $(\chi = 0) \wedge (\chi = 1)$.

6.3.2. Heuristic Procedure $\mathcal{H}$

Even when many applications of this problem are considered to be strategic, shortest path problems often appear as sub-problems of more complex problems. Therefore, there exists interest on developing heuristic methods to provide high-quality solutions in short computation times.

To find the path with minimum cost value heuristically, a two-phase procedure is proposed. In the first phase, a truncated labeling algorithm is executed. Each node j is associated with a set of labels representing paths from node 0 to j. Each label L'_j keeps track of all visited clusters in the set $S_{L'_j} \subseteq \Psi$, the visited nodes in the set $V_{L'_j} \subseteq J$, and the path cost $C(L'_j)$ including both, traveled cost and collected profits. One label L'_j dominates another label L''_j ($L'_j \prec L''_j$), where L'_j and L''_j represent different paths from node 0 to the same node j if 1) $C(L'_j) < C(L''_j)$, $S_{L'_j} \subseteq S_{L''_j}$ implying that L'_j visits at least all the clusters visited by L''_j, and $V_{L'_j} \subseteq V_{L''_j}$ meaning L'_j visits at least all the nodes visited by L''_j ; or 2) if $C(L'_j) = C(L''_j)$, and $S_{L'_j} \subsetneq S_{L''_j}$ implying that L'_j visits at least all the clusters visited by L''_j or $S_{L'_j} \neq S_{L''_j}$, and $V_{L'_j} \subsetneq V_{L''_j}$. The first phase enumerates paths from 0 up to every node by keeping only non-dominated labels. Extensions for label L'_j are limited towards nodes in the set $J \cup \{n+1\} / V_{L'_j}$ to guarantee elementary paths only. It stops when all the existing labels have been extended to unvisited nodes. A limit of K non-dominated labels is imposed per node after extending labels, in order to speed up the search.

When extending labels, two different rules are analyzed. Consider the case in which the labels on a particular node i are to be extended towards another node j. Before the extension, node i and j have a list of K non-dominated labels:$\left\{L_i^1, L_i^2, \cdots, L_i^K\right\}$ and $\left\{L_j^1, L_j^2, \cdots, L_j^K\right\}$ respectively. After the extension, node j will be associated to a set of labels $\left\{L_j^1, L_j^2, \cdots, L_j^K\right\} \cup \left\{L_j^{k+1}, L_j^{k+2}, \cdots, L_j^{2K}\right\}$. The last K labels are computed by adding the arc (i, j) into the labels associated to node i. Assume the worst case, in which all of the $2K$ labels are non-dominated. The two studied rules to keep only K non-dominated labels are: 1) Keep the K non-dominated labels with the lowest cost ($C(L_j)$). 2) Keep the first computed K non-dominated labels, which is the set $\left\{L_j^1, L_j^2, \cdots, L_j^K\right\}$ in the presented example.

Further, a local search procedure is performed as post-optimization with the following traditional neighborhoods:

- EXCHANGE: Modifies the position of a node in the path.
- SWAP: Interchanges the position of two nodes in the path.
- 2-Opt: Erases two arcs in the path and reconnects it with two different arcs.
- 3-Opt: Erases 3 arcs in the path and reconnects it with three different ones in the best possible way.
- INSERT: Insert an unvisited node into the path.

Our local search is structured to be a variable neighborhood descent (VND) [13]. It evaluates each neighborhood starting from EXCHANGE and if an improving movement is found, the search starts again from the first neighborhood. If no improving movement is found in the current neighborhood, the search continues with the next one until the last neighborhood is explored without finding further improvement movements. A best improvement movement is applied in all the search.

6.4. Computational Experiments

Tests for a set of 100 random instances with up to 100 nodes are run on an Intel Xeon with 2.80Ghz processor, 12 GB of RAM and coded in C. We introduce 20 instances with $n = 20$, 30 instances with $n = 50$ and 50 instances with $n = 100$. The coordinates (x_i, y_i) of each node i are randomly generated over a grid of 100×100 together with a random value $\delta_i \sim Normal(50, 20), \forall i \in J$. Arc costs are computed as: $c_{ij} = \left\lceil 100 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - \delta_i/2 - \delta_j/2 \right\rceil$. The number of clusters is generated using a uniform distribution between the range $[n, 2 \cdot n]$. Each node i belongs to cluster t with a probability of 0.5. The complete set of instances is available online at:
`http://ftpprof.uniandes.edu.co/~pylo/inst/GESPP/instances.htm`

Table 19 presents average results for each instance set provided by the presented Branch-and-cut procedure and they are compared to the results by Guerrero et al.[12] which implements the MTZ subtour elimination constraints [16]. Both exact methods are solved under the same conditions using Xpress-IVE. Column # **opt** presents the number of optimal solutions found. Column **avg cpu** presents the average computational time in seconds and column σ presents its standard deviation, also in seconds. Finally, column **avg nodes** accounts for the average number of nodes explored by the procedure.

Only one instance could not be solved by the Branch-and-Cut procedure within a time limit of two hours (Instance 49). We were able to solve 99 instances to optimality,

Table 19: Comparative results for exact procedures

Instance set	IP solver [12]		Branch-and-Cut			
	# opt	avg cpu[s]	# opt	avg cpu[s]	σ[s]	avg nodes
n = 20	20/20	2.90	20/20	0.77	1.00	49.90
n = 50	25/30	1485	29/30	3.90	3.96	146.76
n = 100	0/50	-	50/50	94.38	208.72	7182.20
Average	**45/100**	**826.0**	**99/100**	**48.96**	**154.64**	**3680.44**

with an average computational time of 48.96 seconds. Our Branch-and-Cut explores on average a Branch-and-Bound tree with 3680.4 nodes. In the meanwhile, the Branch-and-Bound procedure presented in Guerrero et al. [12] solves only 45 instances with an average computation time of 826 seconds.

To show that the continuous relaxation of the mathematical formulation presented by Guerrero et al. [12] is weaker than the one presented by equations (155)-(157), (161), and (162)-(165) in section 6.2, the following results are provided. Table 20 presents in columns GAP$_{LP}$ the average gap between the LP relaxation at the root node (z_{LP}) and the optimal solution (z^*) computed as $100 \cdot (z_{LP} - z^*) / z^*$.

Table 20: Gap between LP relaxations and optimal solution for GESPP MIP formulations

	Formulation in Guerrero et al. [12]		Formulation in section 6.2	
Instance set	avg GAP$_{LP}$	max GAP$_{LP}$	avg GAP$_{LP}$	max GAP$_{LP}$
n = 20	8.16	15.64	3.02	12.73
n = 50	2.92	4.26	1.06	2.40
n = 100	1.37	1.98	0.41	0.98
Average	**3.20**	**15.64**	**1.13**	**12.73**

Not only an important reduction in computational time is perceived by using the new mathematical formulation presented in section 6.2, but also the linear relaxation of the problem is improved. On average, the gap between the LP problem and the optimal solution is 1.13% using the presented formulation. The maximum gap for the tested sample of instances is 12.73%. Therefore, we conclude that the formulation provided here is stronger than the one presented in Guerrero et al.[12].

For the two-phase heuristic presented in section 6.3.2, we tested for several values of K. As expected, for low values of K, $\mathcal{H}$ requires extensive intensification in the post-optimization phase while larger values makes $\mathcal{H}$ to perform slowly in the first phase.

Table 21 presents the average results for the same instances as before. Tests are performed for K values from 2 to 100 and the two truncation rules explained in section 6.3.2. The average gap between $\mathcal{H}$ and the optimal solution (gap), and the average computation time in seconds (cpu) is reported for the most relevant K values. Super index 1 is for the solution of the first phase, and 2 for the solutions after local search. Note that increasing K improves the quality of the solution before local search. But even when $K = 100$, the first phase is about 13.16% and 40.05% larger than the optimal solution when using a lowest cost rule and the first computed rule respectively, indicating that the lowest cost rule is better. Also, the impact of the local search operator is critical. In all cases, the solution quality before the local search is greater than 13%, and by applying the operator the quality is improved up to less than 0.9%. In brief, truncating labels with the first computed rule allows a faster performance of the algorithm and gives more chance to the Local Search operator to improve the solution.

Table 21: Computational results:

$\mathcal{H}$ Truncating labels by Lowest cost rule:

	$\mathcal{H}$(K = 2)			$\mathcal{H}$(K = 4)			$\mathcal{H}$(K = 6)			$\mathcal{H}$(K = 8)			$\mathcal{H}$(K = 10)		
Test Set	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu
20 nodes	23.32	1.27	1.79	20.05	0.84	1.93	16.54	1.09	1.91	16.38	0.78	1.99	15.32	0.75	1.99
50 nodes	22.18	0.40	1.82	20.67	0.35	2.05	19.62	0.36	2.24	18.85	0.46	2.62	18.31	0.41	2.87
100 nodes	16.07	0.23	2.83	15.00	0.23	4.47	14.66	0.23	6.06	14.32	0.23	8.42	14.48	0.23	9.96
Average	19.35	0.49	2.32	17.71	0.39	3.24	16.52	0.44	4.08	16.09	0.41	5.39	15.80	0.39	6.24

	$\mathcal{H}$(K = 14)			$\mathcal{H}$(K = 20)			$\mathcal{H}$(K = 35)			$\mathcal{H}$(K = 50)			$\mathcal{H}$(K = 100)		
Test Set	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu
20 nodes	14.11	1.29	2.05	13.28	1.21	2.22	11.46	1.42	2.49	10.45	1.01	2.95	8.56	1.00	4.91
50 nodes	17.89	0.33	3.58	17.00	0.35	4.76	17.21	0.35	8.65	17.08	0.38	13.98	16.30	0.45	40.18
100 nodes	14.08	0.22	15.44	14.02	0.21	25.07	13.62	0.23	62.33	13.34	0.24	116.57	13.12	0.24	401.64
Average	15.23	0.47	9.20	14.77	0.45	14.41	14.26	0.50	34.26	13.88	0.44	63.07	13.16	0.46	213.86

$\mathcal{H}$ Truncating labels by first computed rule:

	$\mathcal{H}$(K = 2)			$\mathcal{H}$(K = 4)			$\mathcal{H}$(K = 6)			$\mathcal{H}$(K = 8)			$\mathcal{H}$(K = 10)		
Test Set	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu
20 nodes	67.26	0.81	1.94	62.83	1.13	2.05	59.80	0.23	2.05	57.76	0.08	2.11	56.08	0.45	2.08
50 nodes	60.96	0.33	1.90	56.08	0.36	2.07	55.28	0.36	2.25	53.42	0.30	2.45	53.08	0.32	2.53
100 nodes	48.49	0.23	2.08	42.09	0.22	2.73	41.61	0.23	3.32	38.83	0.21	4.23	38.82	0.21	4.87
Average	55.98	0.38	2.00	50.43	0.45	2.40	49.35	0.18	2.74	46.99	0.21	3.27	46.55	0.29	3.61

	$\mathcal{H}$(K = 14)			$\mathcal{H}$(K = 20)			$\mathcal{H}$(K = 35)			$\mathcal{H}$(K = 50)			$\mathcal{H}$(K = 100)		
Test Set	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu	gap^1	gap^2	cpu
20 nodes	53.68	0.52	2.12	51.15	0.07	2.28	45.65	2.11	2.49	43.64	2.47	2.80	39.24	3.32	4.17
50 nodes	52.16	0.36	2.95	51.28	0.31	3.63	49.96	0.40	5.74	49.13	0.36	8.28	47.87	0.35	20.31
100 nodes	38.17	0.21	6.84	37.24	0.21	10.25	36.43	0.22	19.78	36.17	0.22	30.89	35.69	0.21	82.68
Average	45.47	0.32	4.73	44.23	0.22	6.67	42.33	0.65	12.11	41.55	0.71	18.49	40.05	0.87	48.27

The rule for truncating the labels has an impact on the performance of $\mathcal{H}$. The lowest cost rule provides better results before local search while the first computed rule provides paths with fewer visited nodes. Given that the local search does not evaluate the removal of nodes, the local search combined with the second rule and a K value sufficiently large could have better exploration of solution space. In fact, the best results are shown for K=6 using the first computed K non-dominated labels. A cpu of 2.74s and a gap to the optimal solution of 0.18% is the best trade-off between solution quality and computation time.

Fig. 13 and 14 depict the growth of the average computational time for different instance sizes and different K values. Fig. 13 compares the average results for $\mathcal{H}$ using the lowest cost rule versus the Branch-and-Cut procedure. The computation times for $\mathcal{H}$ and $K \geq 50$ are larger than the Branch-and-Cut procedure. We remark an important growth of the computation time with the increase in K. Fig. 14 makes a comparison of $\mathcal{H}$ configured with the first computed rule and the Branch-and-Cut. In this case, it is concluded that the growth of computational times is significantly smaller than the Branch-and-Cut.

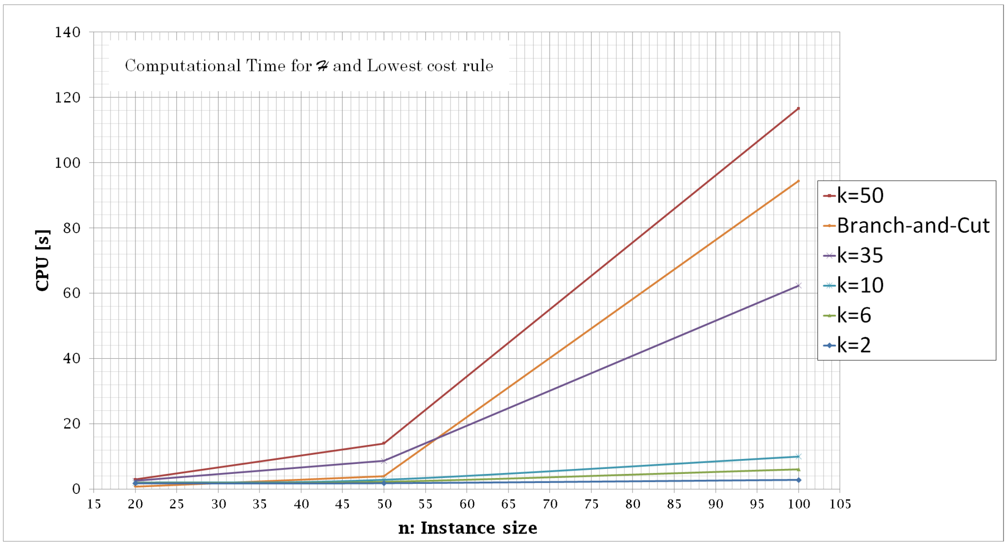

Figure 13: CPU times for $\mathcal{H}$ using Lowest cost rule and Branch-and-Cut

The impact of the first phase of $\mathcal{H}$ to compute an initial solution are measured by providing average results for 5 executions of a random start heuristic with the local search operator. These results are depicted in table 22. While building a random initial solution, improved by the local search operator, is a fast procedure, the performance of this heuristic is volatile and it is outperformed by the proposed approach. This exposes the contribution of the initial solution and the local search operator.

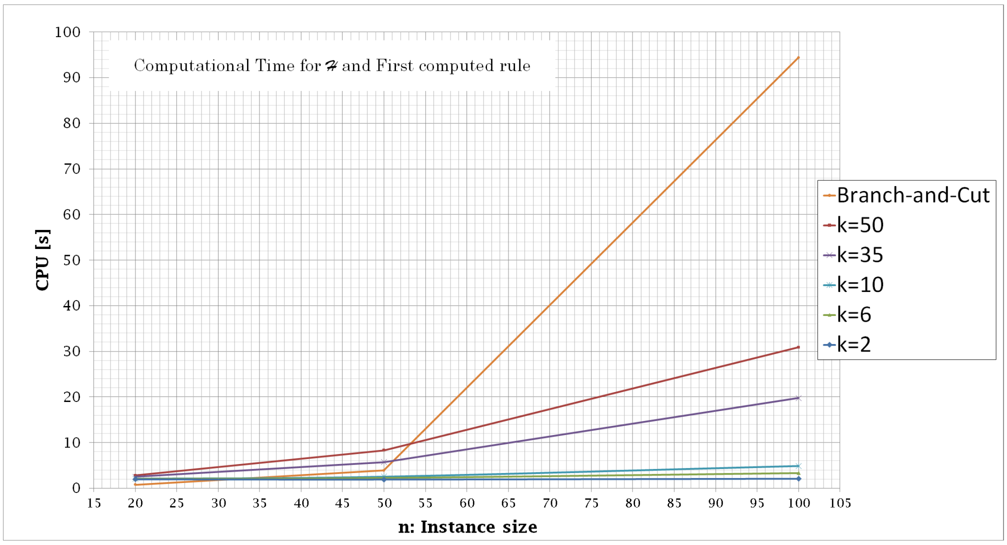

Figure 14: CPU times for $\mathcal{H}$ using First computed rule and Branch-and-Cut

Table 22: Performance of the Random Start Heuristic

	Random Start		
Instance Set	gap[1]	gap[2]	cpu
20 nodes	87.65	2.52	0
50 nodes	60.52	0.39	0.02
100 nodes	49.77	0.21	0.25
Total	**60.57**	**0.73**	**0.13**

6.5. Conclusions

The Generalized Elementary Shortest Path Problem (GESPP) is studied in this paper. It is a generalization of the elementary shortest path problem. Applications are shown in telecommunications, urban transportation network design, and rail lines design.

Two solution methods are proposed which are suitable methods to solve the problem on networks with negative cost cycles. First, a Branch-and-Cut procedure is implemented, based on a new formulation of the problem.

Second, a two-phase heuristic algorithm, denoted $\mathcal{H}$, is proposed. Solving heuristically the problem is relevant when the GESPP arises as a sub-problem in a more complex problem. In that case, it is necessary to compute high quality solutions quickly. In the first phase, a truncated labeling algorithm is performed. A limit K on the number of labels considered per node is forced to speed up the search of an initial solution. Two different truncation rules are proposed and compared. In a second phase of the heuristic, a local search operator is applied with a Variable Neighborhood Descent (VND). Naturally, if the imposed limit in the first phase is sufficiently large, the optimal solution is guaranteed but the computational burden is higher.

Experiments are carried out comparing different mathematical formulations of the problem and different configurations of the $\mathcal{H}$ heuristic. Tests on random instances with up to 100 nodes show improved results for exact methods. These instances are available online. We were able to compute 99% of the optimal solutions under a time limit of 2h and the average computation time is 48s. Heuristic results are also provided, $\mathcal{H}$ shows the best performance when keeping the first computed non-dominated labels as truncating rule and $K = 6$, with an average gap of 0.18% to the optimal obtained in 2.74s.

Future research involves considering forbidden turns, forbidden subpaths and/or multi-path problems. Also, including resource constraints could be an interesting extension (see Di Puglia Pugliese et al.[7]).

Acknowledgements

This research is partially supported by Champagne-Ardenne Regional Council (France), Centro de Estudios Interdisciplinarios Básicos y Aplicados - CEIBA (Colombia) and Université de Technologie de Troyes.

Chapter 6 was submitted as:

W.J. Guerrero, N. Velasco, C. Prodhon, C.A. Amaya (2013) *Branch-and-Cut and heuristic procedure for the Generalized Elementary Shortest Path Problem.* NETWORKS. IN SUBMISSION

A preliminary version of this chapter was published at:
W.J. Guerrero, N. Velasco, C. Prodhon, C.A. Amaya (2013) *On the Generalized Elementary Shortest Path Problem: A heuristic approach.* Electronic Notes in Discrete Mathematics 41(1): 503-510.
`www.sciencedirect.com/science/article/pii/S1571065313001340`

Preliminary results were presented at:
W.J. Guerrero, N. Velasco, C. Prodhon, C.A. Amaya (2013) Heuristic algorithm for the Generalized Elementary Shortest Path Problem. Congrès ROADEF 2013, Troyes, France.

W.J. Guerrero, N. Velasco, C. Prodhon, C.A. Amaya (2013) On the Generalized Elementary Shortest Path Problem: A heuristic approach. INOC – International Network Optimization conference, May 20-22, Tenerife, Spain.

6.6. Résumé en français

Ce chapitre presente dans la section 6.2, la formulation mathématique du problème du plus court chemin généralisé (Generalized Elementary Shortest Path Problem GESPP). Elle est plus forte que celle présentée dans Guerrero et al. [12]. Les méthodes de résolution proposées sont présentées dans la section 6.3. Dans la section 6.3.1, une procédure de branchements et coupes est proposée et la section 6.3.2 explique l'algorithme heuristique, noté $\mathcal{H}$. Une évaluation de la performance est étudiée dans la section 6.4. Les conclusions sont exposées dans la section 6.5.

Le GESPP peut être défini sur un graphe complet, pondéré et non orienté G composé d'un ensemble J de n nœuds, un nœud source $\{0\}$ et un sommet destination $\{n+1\}$. Dans la suite, l'ensemble des nœuds V est défini comme $J \cup \{0\} \cup \{n+1\}$. Chaque arc qui appartient à $A = \{(i, j), \forall i, j \in V$ dans G est associé à un coût $c_{ij} \in ($ G peut contenir des cycles négatifs). En outre, les nœuds sont regroupés en clusters non-disjoints prédéfinis. Chaque cluster $t \in \Psi$ est associé à un profit $p_t \geq 0$ dans la fonction de coût si au moins un nœud dans le cluster t est visité. Selon l'application, les clusters pourraient être interprétés comme des groupes de nœuds avec des liaisons, facilement accessibles les uns aux autres, ou une sorte de garantie de couverture. Soit x_{ij} une variable de décision binaire indiquant si l'arc $(i, j|i < j)$ appartient au chemin choisi. Soit y_t une variable binaire égale à 1 si et seulement si le cluster $t \in \Psi$ est visité au moins une fois et soit z_j une variable binaire égale à 1 si et seulement si le nœud $j \in J$ appartient au chemin choisi.

L'objectif est défini par l'équation (155) visant à minimiser le résultat du coût de la longueur du chemin après la déduction des profits des clusters. Les contraintes (156), (157) sont traditionnelles au problème du plus court chemin: Le chemin doit commencer au niveau du nœud source et finir au niveau du nœud de destination. Les contraintes (158) sont les contraintes de conservation de flux. Les contraintes d'élimination de sous-tours (159) sont nécessaires compte-tenu des éventuels cycles négatifs dans G. Un profit d'un cluster est obtenu si et seulement si le chemin visite n'importe quel nœud appartenant au cluster comme indiqué par les contraintes (160). Les variables de décision sont binaires tels que définis par les équations (161). Une deuxième formulation est proposée en remplaçant les équations (158)-(160) par les contraintes (162)-(165).

En utilisant la deuxième formulation présentée dans la section 6.2, une procédure de branchements et coupes a été mise en œuvre. Étant donné que l'ensemble des contraintes (163) est exponentiel, elles sont ajoutées dynamiquement à chaque nœud de la recherche. Une procédure de séparation heuristique permet d'identifier des sous-tours à chaque nœud de l'arbre de recherche. Si plus d'une coupe est trouvée, elles sont toutes ajoutées. Le lecteur est renvoyé à Lysgaard et al. [15] pour plus de détails sur les procédures de séparation exactes et heuristiques dans les algorithmes de branchements et coupes sur des problèmes de tournées. De plus, notre stratégie de branchement consiste à sélectionner la variable ayant une valeur de solution après la résolution de la relaxation linéaire la plus proche de 0.5. Il s'agit d'une pratique courante dans les implémentations de branchements et coupes. L'objectif est de sélectionner une variable $\chi \in \{x, y, z\}$ et d'imposer une disjonction ($\chi = 0) \wedge (\chi = 1)$ afin de continuer la recherche.

Des tests sur un ensemble de 100 instances générées aléatoirement avec un maximum

de 100 nœuds sont faits avec un processeur Intel Xeon avec 2.80GHz, 12 Go de RAM et codées en C. Nous avons généré 20 instances avec $n = 20$, 30 instances avec $n = 50$ et 50 instances avec $n = 100$. Les coordonnées (x_i, y_i) de chaque nœud i sont générées aléatoirement sur une grille de 100 $\times$ 100 avec une valeur aléatoire $\delta_i \sim$ normal (50,20), $\forall i \in J$. Les coûts pour traverser un arc sont calculés comme suit:
$c_{ij} = \left\lceil 100 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - \delta_i/2 - \delta_j/2 \right\rceil$.

Le nombre de clusters est généré en utilisant une distribution uniforme entre l'intervalle $[n, 2 \cdot n]$. Chaque nœud i appartient au cluster t avec une probabilité de 0.5. L'ensemble complet des instances est disponible en ligne à l'adresse:
`http://ftpprof.uniandes.edu.co/~pylo/inst/GESPP/instances.htm`

Une seule instance ne peut être résolue par la procédure de branchements et coupes dans un délai de deux heures (Instance 49). Nous avons réussi à résoudre 99 des instances à l'optimalité, avec un temps de calcul moyen de 48.96 secondes. Notre méthode de branchements et coupes explore en moyenne un arbre avec 3680.4 nœuds, alors que celui présenté dans Guerrero et al. [12], basé sur une autre formulation mathématique, résout seulement 45 instances avec un temps de calcul moyen de 826 secondes sur le même ordinateur.

Dans le tableau 20, il est présenté dans les colonnes GAP$_{LP}$ l'écart moyen entre la relaxation linéaire au niveau du nœud racine (z_{LP}) et la solution optimale (z^*) calculé comme $100 \cdot (z_{LP} - z^*)/z^*$.

Non seulement une réduction importante des temps de calcul est perçue en utilisant la nouvelle formulation mathématique présentée dans la section 6.2, mais aussi la relaxation linéaire du problème est améliorée. En moyenne, l'écart entre le LP du problème et la solution optimale est de 1.13% en utilisant la formulation présentée. L'écart maximal pour l'échantillon testé est de 12.73%. Par conséquent, nous concluons que la formulation présentée ici est plus forte que celle présentée par Guerrero et al.[12].

Une heuristique en deux phases est ensuite proposée. Dans la première phase, un algorithme d'étiquetage tronqué est réalisé. Une limite sur le nombre de labels considérés par nœud est imposée pour accélérer la recherche d'une solution initiale. Deux règles de troncature différentes sont proposées et comparées. Dans une deuxième phase de l'heuristique, un opérateur de recherche locale est appliqué avec une structure de voisinage variable (variable neighborhood descent VND). Naturellement, si la limite imposée dans la première phase est suffisamment grande, la solution optimale est garantie mais l'effort de calcul informatique est plus élevé.

Les résultats de l'heuristique montrent les meilleures performances lorsque la limite des labels $K = 6$ et en utilisant une première règle calculée lors de la troncature des étiquettes, avec un écart moyen de 0.18% de l'optimal calculée en 2.74s. La recherche future consiste à considérer les contraintes de ressources, les virages interdits, sous-chemins interdits et / ou des problèmes multi-chemin.

References

[1] Ahmed, M., Lubiw, A., 2013. Shortest paths avoiding forbidden subpaths. Networks 61 (4), 322–334.

[2] Baldacci, R., Bartolini, E., Laporte, G., 2010. Some applications of the generalized vehicle routing problem. Journal of the Operational Research Society 61 (7), 1072–1077.

[3] Bellman, R., 1958. On a routing problem. Quarterly of Applied Mathematics 16 (1), 89–90.

[4] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2001. Introduction to algorithms, 2nd Edition. MIT Press, Cambridge, MA, USA.

[5] Current, J., Revelle, C., Cohon, J., 1987. The median shortest path problem: A multiobjective approach to analyze cost vs. accessibility in the design of transportation networks. Transportation Science 21 (3), 188–197.

[6] Di Puglia Pugliese, L., Guerriero, F., 2012. Shortest path problem with forbidden paths: The elementary version. European Journal of Operational Research In PRESS.

[7] Di Puglia Pugliese, L., Guerriero, F., 2013. A survey of resource constrained shortest path problems: Exact solution approaches. Networks 62, 183–200.

[8] Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. NUMERISCHE MATHEMATIK 1 (1), 269–271.

[9] Feillet, D., Dejax, P., Gendreau, M., 2005. Traveling salesman problems with profits. Transportation Science 39 (2), 188–205.

[10] Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks 44 (3), 216–229.

[11] Festa, P., Guerriero, F., Laganà, D., Musmanno, R., 2013. Solving the shortest path tour problem. European Journal of Operational Research 230 (3), 464 – 474.

[12] Guerrero, W., Velasco, N., Prodhon, C., Amaya, C.-A., 2013. On the generalized elementary shortest path problem: A heuristic approach. Electronic Notes in Discrete Mathematics 41, 503– 510.

[13] Hansen, P., Mladenovic, N., 2003. Variable neighborhood search. In: Glover, F., Kochenberger, G. (Eds.), Handbook of Metaheuristics. Vol. 57 of International Series in Operations Research & Management Science. Springer New York, pp. 145–184.

[14] Labbé, M., Laporte, G., Rodríguez-Martín, I., González, J. S., 2005. Locating median cycles in networks. European Journal of Operational Research 160 (2), 457–470.

[15] Lysgaard, J., Letchford, A. N., Eglese, R. W., Jun. 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming 100 (2), 423–445.

[16] Miller, C. E., Tucker, A. W., Zemlin, R. A., 1960. Integer programming formulation of traveling salesman problems. Journal of the ACM 7 (4), 326–329.

[17] Nepal, K., Park, D., 2005. Solving the median shortest path problem in the planning and design of urban transportation networks using a vector labeling algorithm. Transportation Planning and Technology 28 (2), 113–133.

[18] Tricoire, F., Romauch, M., Doerner, K., Hartl, R., 2010. Heuristics for the multi-period orienteering problem with multiple time windows. Computers & Operations Research 37 (2), 351–367.

[19] Villeneuve, D., Desaulniers, G., 2005. The shortest path problem with forbidden paths. European Journal of Operational Research 165 (1), 97 – 107.

Part III.
General Conclusion

7. Conclusions

In conclusion, this thesis studies the integrated problem of optimizing depot location, routing and inventory levels. Two solution methodologies are exposed, showing innovative decomposition methods for a complex problem and providing competitive upper bounds for random instances. Further, the presented approach proposes to optimize globally a series of decisions in logistics that are highly interconnected when designing a supply chain.

For that matter, solving simultaneously location, inventory management, and routing decisions has an increasing interest. Considering the cost of performing distribution and the increasing need to transport goods for large distances across the globe, finding high quality solutions for the Inventory Location-Routing Problem (ILRP) within reasonable computational times, is nowadays a relevant issue that industries need to do to achieve competitive performance.

The problem considered in this thesis aims to optimize a supply chain design assuming rather realistic inventory and routing constraints. After identifying a set of feasible candidate locations for building or renting depots and their storage capacity, the challenge is to select the optimal corresponding subset to perform the logistic operations.

The target of the supply chain studied for the ILRP is to manage to transport and store product at depots and retailers. The objective function is to minimize the sum of the costs of opening a depot, holding inventory at retailers and depots, and performing routes between the former facilities. It is realistic to assume that retailers and depots are both interested in optimizing this objective function since they will both get benefits from it. The simple case is when retailers and depots are owned by the same stakeholder, in which case he probably has no preference in the cost balance between retailers and depots. On the other hand, if depots and retailers cooperate to reduce costs, this might have an impact on prices to benefit fairly both agents in the supply chain.

Three groups of constraints are presented: One group is dedicated to make location decisions and a feasible allocation of retailers, a second group aims to coordinate the management of stock through the logistic chain, and a third group is considered to guarantee the feasibility of the routing decisions. On the first group of constraints, the equations that define the selected structure for the supply chain are given. Multiple depots might be chosen and every retailer must be allocated to a single depot. This last constraint is studied since product traceability is highly improved with it.

Among inventory control constraints, limited storage capacity is considered at both echelons of the supply chain. Finally, distribution must be made by a fleet of capacitated and identical vehicles. To ease logistic operations, each period a retailer is forced to be visited at most once and each vehicle performs at most a single tour.

This definition of the ILRP opposes to most of the precedent approaches where these decisions are divided into strategical, tactical and operational levels. The advantage of decomposing the problem is to make it simpler to solve. Nevertheless, global optimality is not guaranteed. When routing and inventory management costs over several periods are in the same order of magnitude as location costs, it is relevant to optimize simultaneously the three decisions.

Further, in this thesis the inventory-routing problem (IRP) has also received special attention. If the location of depots is considered to be fixed, the quantities to deliver and the sequence of retailers to visit is optimized simultaneously by the IRP. This is a supply chain management problem and it has been studied mainly recently and application contexts include gas and oil distribution.

In fact, routing and inventory management decisions are closely related. The optimal set routes to perform distribution is built as a function of the quantities to deliver. Also, the optimal quantities and frequency of replenishment consider the cost of transportation resulting from the choice of the sequence of visited retailers. This problem reduces to the classical vehicle routing problem (VRP) if a single period is taken as planning horizon. Therefore, it represents an important challenge due to its computational complexity, the potential savings that may arise from its optimization and the large application contexts.

From the beginning of the thesis, a literature review is presented and the clear conclusion is that the ILRP has not being extensively studied before. Few journal papers have been published and some few more conference papers are available. Nonetheless, we have not found benchmark instances or deep analysis on the integrated problem.

Then, the difficulty of solving the problem is exposed. Additionally, two decomposition methods are presented based on the premise that the resulting sub-problems are solvable using exact methodologies and cooperation with heuristics without losing a global optimization perspective. These are defined as hybrid methodologies or alternativelly as “Matheuristics”. Chapter 2 presents a cooperative approach between exact and heuristic components while chapter 4 exposes a heuristic based on a Dantzig-Wolfe decomposition for the routing variables that has been called “relax-and-price”.

Tests are made on a new set of randomly generated instances. On average, our cooperative approach computes solutions faster than the “relax-and-price” method, but the average quality of the solutions computed by the former are higher. Both of these methods outperform the studied heuristics based on sequential optimization and on Branch-and-Bound. Hence, the exposed results prove that hybrid methods are suitable alternatives to solve such a complex problem and as the performance of commercial solvers improves, it is expected the presented methodologies to get faster as well.

The conceptual ideas of the presented methods are also adapted and tested on related sub-problems. The algorithm presented in chapter 2 is tested on benchmark instances for a Location-Routing problem providing competitive performance. Inspired by this cooperative algorithm, chapter 3 presents a Multi-Start Iterated Local Search to solve a new variant of the Inventory-Routing problem considering multiple depots. This method is able to compute a significant number of new best known solutions for benchmark instances for the single-depot case, in shorter computational times than state-of-the-art results. New instances for the multi-depot configuration are proposed and the corresponding computational study is also presented.

Combining Lagrangian Relaxation together with Column generation techniques as presented in chapter 4 is also tested on the Inventory-Routing Problem. Chapter 5 presents a new Dantzig-Wolfe formulation for the IRP and the current research. So far, results provide high quality solutions but computational times are larger than benchmark methods. In general, the procedure seems promising but room for improvement is acknowledged.

Finally, chapters 4 and 5 imply the resolution of a pricing sub-problem to compute new columns for the master problem. Based on the new formulations, the proposed pricing problem is denoted as the Generalized Elementary Shortest Path problem. Applications for this new problem include the design of new lines for public transportation systems and telecommunications. Two different mathematical formulations were proposed, together with exact methodologies based on Branch-and-Bound and a two phase heuristic in chapter 6.

The exposed research has resulted on the publication of two journal papers available on the *International Journal of Production Economics* and on *Electronic Notes on Discrete Mathematics*. Further, preliminary results were presented on 14 national and international conferences.

It is important to state as final conclusion that the computational experiments made on the ILRP that were presented in this thesis compare upper bounds obtained by several heuristics. Though it is a valid comparison to evaluate the relative performance of each method, it is still unknown how far from the optimal solutions these results are. Obtaining high quality lower bounds will be the scope of future research.

Other research perspectives, additional to the possible extensions exposed in chapter 1.1, include the assumption of having stochastic demand. The proposed model is under the assumption that demand is deterministic. It is a good first approach to start studying this difficult problem. Nevertheless, it could be interesting to consider how operational decisions are affected by the randomness of the demand by evaluating the objective function using a discrete event simulation model. It could be considered to apply more sophisticated modeling techniques and optimization methods such as simulation-based optimization approaches.

8. Conclusion générale en Français

Un intérêt croissant sur la résolution simultanée de problèmes de localisation de dépôts, de gestion des stocks, et de tournées de véhicules est porté. Trouver des solutions de qualité pour ce problème intégré de localisation et tournées avec gestion de stocks (Inventory Location-Routing Problem ILRP) est de nos jours pertinent. C'est un challenge considérant le coût de distribution résultant du besoin croissant de transporter des marchandises pour des grandes distances à travers le monde, et que les industries doivent resoudre ce problème pour obtenir un rendement compétitif.

La version considérée dans cette thèse vise à optimiser une conception de la chaîne d'approvisionnement soumis à des contraintes de gestion de stocks et de routage plutôt réalistes. Après avoir identifié un ensemble d'emplacements possibles pour la construction ou la location des dépôts et leur capacité de stockage, le défi consiste à sélectionner le sous-ensemble optimal pour effectuer les opérations logistiques.

Dans ce contexte, l'attention est portée sur les chaînes d'approvisionnement à deux échelons. La fonction-objectif cherche à minimiser la somme des coûts d'ouverture des dépôt, le coût de possession des stocks chez les détaillants et les dépôts, et le coût d'effectuer la distribution entre les installations. Il est réaliste de supposer que les détaillants et les dépôts sont à la fois intéressés par l'optimisation de cette fonction-objectif, car ils en seront bénéficiaires. Le cas le plus simple apparaît quand les détaillants et les dépôts sont détenus par la même partie prenante, dans ce cas, elle n'a probablement pas de préférence dans l'équilibre des coûts entre les détaillants et les dépôts. D'autre part, si les dépôts et les détaillants coopèrent pour réduire leurs coûts, cela pourrait avoir un impact sur les prix, faisant ainsi bénéficier équitablement les deux agents dans la chaîne d'approvisionnement.

Dans le modèle, 3 groupes de contraintes sont présentés: un groupe est dédié aux décisions de localisation et d'affectation des détaillants, un deuxième groupe cherche à coordonner le flux de produit dans la chaîne logistique, et un troisième groupe garantit la faisabilité de décisions de routage. Dans le premier groupe de contraintes, les équations qui définissent la structure choisie pour la chaîne d'approvisionnement sont donnés. Plusieurs dépôts peuvent être sélectionnés et tous les détaillants doivent être affectés à un seul dépôt. Ce dernier ensemble de contraintes a été ajouté car il améliore fortement la traçabilité des produits.

Parmi les contraintes de gestion des stocks, nous considérons la capacité limitée de stockage aux deux échelons de la chaîne d'approvisionnement. Enfin, la distribution doit être faite par une flotte de véhicules de capacité limitée et identique. Pour faciliter les opérations logistiques, à chaque période un détaillant doit être visité au plus une fois et chaque véhicule doit effectuer un seul tour tout au plus.

Cette définition impose une approche de résolution différente à celle de la plupart des précédentes où les décisions sont divisés en stratégiques, tactiques, et opérationnelles. L'avantage de décomposer le problème est de le rendre plus simple à résoudre. Néanmoins, optimalité globale n'est pas garantie quand les coûts relatif à la distribution et à la gestion des stocks sur plusieurs périodes sont du même ordre de grandeur que les coûts de localisation, il est alors pertinent d'optimiser simultanément les trois décisions.

En outre, dans cette thèse le problème de tournées avec gestion des stocks (Inventory-Routing Problem IRP) a également reçu une attention particulière. Si l'emplacement de dépôts est considéré comme fixe, les quantités à livrer et la séquence des détaillants à visiter sont optimisés simultanément par l'IRP. Il s'agit d'un problème de gestion de la chaîne d'approvisionnement et il a été particulièrement étudié récemment et ses contextes d'application comprennent la distribution de gaz et du pétrole.

Au delà de la conception de réseau, les problèmes intégrant les décisions de routage et de stockage simultanément sont intéressants car ces décisions sont étroitement liées. L'ensemble optimale de tournées pour effectuer la distribution est construit en fonction des quantités à livrer. En outre, les quantités et la fréquence optimales d'approvisionnement se basent sur le coût de transport résultant du choix de la séquence des détaillants visités. Ce problème se réduit au problème de tournées de véhicules classique (Vehicle Routing Problem VRP) si une seule période est considérée comme horizon de planification. Par conséquent, il représente un défi important en raison de sa complexité de calcul, les économies potentielles qui pourraient résulter de son optimisation, et la diversité des contextes d'application.

Alors, dès le début de la thèse, une revue de la littérature est présentée et la conclusion claire est que l'ILRP n'a pas reçu d'études approfondies jusqu'à maintenant. Quelques articles de journaux ont été publiés et des quelques documents de conférence sont disponibles. Néanmoins, nous n'avons pas trouvé des test de référence ou des analyses en profondeur sur le problème intégré.

La difficulté du problème est exposée. Ensuite, les deux méthodes de décomposition sont présentées sur la base de la prémisse que les sous-problèmes résultant peuvent être résolus en utilisant des méthodes exactes et de la coopération avec des heuristiques sans perdre une perspective d'optimisation globale. Celles-ci sont définies comme des méthodes hybrides ou souvent appelées “ Matheuristics ”. Chapitre 2 présente une approche coopérative entre les composants exacts et heuristiques tandis que le chapitre 4 expose une heuristique basée sur une décomposition de Dantzig- Wolfe pour les variables de routage qui a été appelée “relax-and-price”.

Les expériences sont réalisées sur un nouvel ensemble d'instances générées aléatoirement. En moyenne, notre approche coopérative calcule des solutions plus rapidement que la méthode de “relax-and-price”, mais la qualité moyenne des solutions calculées par la deuxième méthode est plus élevée. Ces deux méthodes sont plus performantes que les heuristiques étudiées basées sur l'optimisation séquentielle et sur la méthode de “Branch-and-Bound”. Par conséquent, les résultats exposés prouvent que les méthodes hybrides sont des bonnes alternatives pour résoudre un problème aussi complexe. Ajoutons que quand les performances des solveurs commerciaux s'amélioreront, il est prévu que les méthodologies présentées soient plus rapides aussi.

Les idées conceptuelles sont également testés sur des sous-problèmes liées à l'ILRP. L'algorithme présenté dans le chapitre 2 est testé sur des instances de référence pour un problème de Localisation-Routage montrant une performance compétitive. Inspiré par cet algorithme coopératif, le chapitre 3 présente une recherche locale itérée à plusieurs points de départ ou “multi-start iterated local search” (MS-ILS) pour résoudre une nouvelle variante du problème de tournées avec gestion de stocks qui considère plusieurs dépôts. Cette méthode permet de calculer un nombre important de nouvelles meilleures solutions pour des instances de référence pour le problème à un seul dépôt, en temps de calcul plus courts que les résultats des dernières méthodes de la littérature. Des nouvelles instances pour le problème de tournées avec gestion de stocks à plusieurs dépôts sont proposées et l'étude de performance correspondante est également présentée.

La combinaison des méthodes de relaxation de Lagrange ainsi que des techniques de génération de colonnes telle que présentées dans le chapitre 4 est également testée sur le problème de tournées avec gestion de stocks. Le chapitre 5 présente une nouvelle formulation de Dantzig-Wolfe pour l'IRP qui nécessite encore quelques ajustements. C'est une recherche qui est en cours. Jusqu'à présent, les résultats fournissent des solutions de haute qualité, mais les temps de calcul sont plus grands que les méthodes de la littérature. En général, la procédure semble prometteuse mais il reste incontestablement une place pour de l'amélioration.

Enfin, les chapitres 4 et 5 font appel à la résolution d'un sous-problème de "pricing" pour calculer de nouvelles colonnes pour le problème maître. Basé sur les nouvelles formulations de l'ILRP, le problème de "pricing" proposé est appelé le problème généralisé du plus court chemin élémentaire.

Les applications pour ce nouveau problème incluent la conception de nouvelles lignes pour les systèmes de transport en commun et les télécommunications. Deux formulations mathématiques différentes ont été proposées, avec des méthodologies exactes basées sur la méthode de branchement et coupes (branch-and-cut) et une méthode heuristique à deux phases dans le chapitre 6.

La recherche dévéloppée dans cette thèse a permis la publication de deux articles dans des journaux. Ils sont disponibles dans *International Journal of Production Economics* et dans *Electronic Notes on Discrete Mathematics*. En outre, les résultats préliminaires ont été présentés dans 14 conférences nationales et internationales.

Finalement, il est important de conclure que cette recherche présente des expériences de calcul sur le ILRP qui comparent des bornes supérieures obtenues par plusieurs méthodes heuristiques. Bien que cela soit une comparaison valable pour évaluer la performance relative de chaque méthode, il est encore inconnu dans quelle mesure les solutions sont proches des optimales. Le calcul des bornes inférieures de qualité sera le but de la recherche future.

Comme autres perspectives de recherche, en plus des extensions possibles exposées dans le chapitre 1.1, nous pourrions considérer une demande stochastique. Le modèle proposé est sous l'hypothèse que la demande est déterministe. Cela est une première approche pour étudier le problème. Néanmoins, il pourrait être intéressant de déterminer l'impact des aléas de la demande sur les décisions opérationnelles par l'évaluation de la fonction objectif en utilisant un modèle de simulation à événements discrets. On pourrait appliquer des techniques de modélisation plus sophistiquées et des méthodes d'optimisation telles que les approches d'optimisation basées sur la simulation.

William Javier GUERRERO RUEDA

Doctorat : Optimisation et Sûreté des Systèmes

Année 2014

Modèles et méthodes d'optimisation pour le problème de localisation-routage avec contraintes de stockage

Cette thèse considère le problème consistant à intégrer les décisions de routage et stockage lors de la conception de la chaîne logistique. Le but est de sélectionner des dépôts parmi un ensemble de candidats pour desservir un ensemble de détaillants à l'aide d'une flotte de véhicules de capacité permettant visiter plus d'un détaillant par route. On cherche à déterminer la localisation de ces dépôts et les tournées des véhicules afin de maintenir leurs niveaux optimaux de stocks. La demande chez les détaillants est connue à l'avance. Des applications dans les domaines de la logistique humanitaire et militaire sont envisageables. Pour résoudre le problème, deux matheuristiques sont proposées. Dans la première partie, une méthode coopérative qui combine des méthodes exactes pour le problème de conception de la chaîne logistique et des méthodes heuristiques de routage est présentée. Dans la deuxième partie, une méthode de décomposition utilisant une réformulation de Dantzig-Wolf sur les variables de routage est proposée. L'algorithme intègre les concepts de génération de colonnes, relaxation lagrangienne et recherche locale. Les résultats montrent la capacité des algorithmes à trouver des solutions de bonne qualité et nous estimons de façon empirique l'impact de considérer un modèle intégré au lieu d'utiliser une méthode d'optimisation séquentielle. De plus, les résultats des méthodes présentées sur des sous-problèmes sont aussi étudiés. Ces sont: le problème de localisation-routage, le problème de tournées avec gestion de stocks, et le problème de plus court chemin généralisé.

Mots clés : optimisation combinatoire - métaheuristiques - transport - gestion des stocks – logistique (organisation).

Models and Optimization Methods for the Inventory-Location-Routing Problem

The problem of designing a supply chain including simultaneously routing and inventory management decisions is studied in this thesis. The objective is to select a subset of depots to open, the inventory policies for a 2-echelon system, and the set of routes to perform distribution from the upper echelon to the next using a homogeneous fleet of vehicles over a finite planning horizon. Demand is considered to be known. Applications are found in humanitarian logistics and military logistics. To solve the problem, two matheuristic procedures are developed. On the first part a cooperative algorithm combining exact methods for the supply chain design problem and routing heuristics is presented. On the second part, a partition is proposed using a Dantzig-Wolf reformulation on the routing variables. An hybridization between column generation, Lagrangian relaxation and local search is proposed in this part, put together as a heuristic method. Furthermore, results demonstrate the capability of the algorithms to compute high quality solutions and empirically estimate the improvement in the cost function of the proposed model when compared to a sequential optimization approach. Furthermore, results of the proposed methodologies on benchmark instances for subproblems are studied as well. Those are the capacitated location-routing problem, the inventory-routing problem, and the generalized elementary shortest path problem.

Keywords: combinatorial optimization – metaheuristics - transportation - inventory control – business logistics.

Thèse réalisée en partenariat entre :

Ecole Doctorale "Sciences et Technologies"